Optimal Operation of Small LNG Refrigeration Cycles

Alexander Leguizamon Robayo

June 26, 2016

ABSTRACT

With the recent increases in the market of natural gas, new production alternatives have become available. One of these is the production in remote places. To increase the profit of this process, it is necessary to count with highly efficient transportation.

The Mini-LNG alternative(Neksået al., 2010) is an on-board refrigeration system that aims to minimize the gas loses. This work aimed to model and optimize the operation of it.

Modelling and optimization were carried out using Matlab®. To model the fluid properties, Soave's modification of Redlich-Kwong equation of state. The solution of the model and properties was done using an equations-oriented approach. The approach followed the guidelines set up by Kamath et al. (2010).

The model was optimized for a range of disturbances. The optimization problem was set up according to the plantwide control steps proposed by Skogestad (2000). The chosen method was Matlab®interior point.

The solution of the nonlinear optimization problem provided sufficiently good results to account for the internal interactions of the system.

ACKNOWLEDGEMENTS

First, I want to thank my supervisor Sigurd Skogestad for allowing me to work on such a challenging project and be part of the Process System Engineering group at NTNU. His support and insights were very important during the development of this work since the specialization project. I would like to honestly remark that his career has been a model for me to look up to.

I wish to express my most sincere thanks to my co-supervisor Adriana Reyes. Her support from the beginning of my Master has been invaluable. Her constant backing and feedback throughout the project helped me to have a clearer understanding of the project and the tasks at hand.

I want to acknowledge the whole research group in Process Systems Engineering, with a special mention to Ian Sulc for his helpful advice on the formulation and solution of the optimization problem.

Last, I want to thank all my friends and family for their support and patience throughout this project. I want to make an special mention to Emma Adams for being an invaluable source of emotional support and for her help editing the main chapters of this document.

NOMENCLATURE

Acronyms

CV	Controlled variable
DOF	Degrees of freedom
MV	Manipulated variable
PI	(Controller) Proportional Integral
PID	(Controller) Proportional Integral Derivative
SRK	Soave's modification of Redlich-Kwong equation of state
BOG	Boil-off Gas
CG	Conjugate Gradient
EO	Equations-oriented
LNG	Liquefied Natural Gas
NG	Natural Gas
NLP	Non-linear programming
VLE	vapour-liquid equilibrium

Greek Letters

α	Alpha parameter SRK
α_i	Alpha parameter SRK for an i component
ω	Acentric factor
ϕ_i	Fugacity for an i component

Roman Symbols

A	A parameter SRK
$A_{i,j}$	Interaction parameter for an i and j component pair
A_i	Partial A parameter SRK for an i component
B	B parameter SRK
B_i	Partial B parameter SRK for an i component
C_i	Correction for the fugacity for an i component
f	Fugacity
J	Cost function
J_{opt}	Optimized cost function
m	Slope as a function of the acentric factor
N_0	Number of degrees of freedom with no steady-state effect
N_{MV}	Number of dynamic manipulated variables
N_{ss}	Steady-state degrees of freedom
P	Pressure
P_c	Critical pressure
P_r	Reduced pressure
T	Temperature
T_c	Critical temperature
T_r	Reduced temperature
Z	Compressibility factor

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 Project Background

Natural gas (NG) is a hydrocarbon mixture that is mostly composed by methane. It can include varying amounts of light alkanes and some small portion of carbon dioxide, nitrogen and/or hydrogen sulfide. This gas is used mostly as fuel and as raw material for manufacturing. As a fuel, it is used in steam boilers, brick and cement making; glass making and as a heat source for sterilizing instruments for food processing. The NG is used as raw material for petrochemical manufacturing, this can covers a broad spectrum of products, from hydrogen production to ammonia production, among others (Mokhatab, Saeid. et al., 2006).

Natural gas is considered as an environmentally friendly fuel, which offers a strong environmental advantage over other fossil fuels (Mokhatab, Saeid. et al., 2006). In order to supply the gas demand, it is important to stablish an efficient mean of transportation for the gas. As an answer to this challenge, the liquefaction process for gas was developed in the 70s. In this process the gas is cooled to $-160\,^{\circ}\mathrm{C}$ and liquefies. The liquefied natural gas (LNG) has a volume more than a thousand times lower than that of gas at room temperature (Mokhatab, Saeid. et al., 2006).

Nowadays NG constitutes around a quarter of the global energy demand (IGU, 2015b) and it is projected to increase by 1.9% per year, until 2030 (B.P., 2015). The participation of LNG corresponds to 10% of the total NG global demand. This has been a considerable increase given that its share was only 4% in 1990. LNG supply is the fastest growing source of gas (growing at 7% per year since 2000) and it is predicted to keep on expanding its contribution in the gas market up to 2020 (IGU, 2015b).

As a consequence of this growth of the demand for NG, new alternatives have been developed to improve the efficiency and viability of new and remote NG sources (IGU, 2015b; Neksået al., 2010). One of the most important developments is the introduction of floating LNG. This creates a new opportunity to sell gas that would otherwise be lost through transportation, as well as to avoid flaring gas from remote

oil offshore locations. However, this technology is fairly new and there is still some degree of uncertainty attached to the operation of these on-board facilities (IGU, 2015a).

One of the many alternatives for a floating LNG facility is the one developed by Neksået al. (2010). The study, modelling and optimization of this plant will be the main focus for this thesis.

1.2 Scope of this work

The main objective of this work can be defined as finding the active constraints regions for the Mini-LNG plant proposed by Neksået al. (2010) within the framework of *Self-optimizing plantwide control* proposed by Skogestad (2000).

Specifically, this objective can be achieved by carrying out two main tasks:

- **Modelling.** Proposing an appropriate model of the plant, well suited for optimization given different perturbations. This means that the model should be robust enough to converge throughout the different optimization runs, as well as reliable enough to provide results that are actually representative of the real plant. In practical terms, to satisfy this to criteria a trade of between detail and robustness is made.
- **Optmization.** Once a model is defined the next step is to optimize it. The optimization is carried in two main parts.
  - The first corresponds to a qualitative part in which the problem is defined, this includes carrying out a degrees of freedom analysis to define the optimization variables and the main disturbances. Additionally operational and quality constraints should be carefully defined.
  - The second step is the solution of this optimization problem. This includes the analysis of the results from which the active constraints regions can be defined. For this, the model is both solved an optimized in an equations-oriented (EO) approach.

1.3 Previous work

This thesis is a continuation of the work carried out by Leguizamon (2015) as a specialization project.

Leguizamon (2015) proposed a first modelling approach using Matlab in a sequential-modular approach for the Mini-LNG plant. His approach set up the basis for the main model of this work.

However, there are some shortcomings regarding the thermodynamics and the degrees of freedom analysis on the previous work.

- *Thermodynamics:* the previous work gives an initial formulation in which the nonlinearities of the thermodynamics do not allow a straightforward convergence. On Chapter 4, a solution to this issue is proposed and implemented.
- *Degrees of freedom:* the active charge is not considered as a degree of freedom in the previous work. This is taken into consideration and improved in the analysis carried out in Section 5.2.

In this thesis, these problems are solved, improving the previous sequential-modular formulation in order to have a more reliable and robust base case for the optimization problem.

1.4 Thesis structure

This thesis is divided in 9 chapters as follows:

Chapter 1 contains the introductory remarks such as the scope and context of this work.

Chapter 2 gives a theoretical summary of optimization is given, focusing on interior point methods used on this work.

Chapter 3 summarizes the plantwide control (Skogestad, 2000) procedure and sets up the context for this work inside that framework.

Chapter 4 studies the thermodynamic foundation for the model. It includes details on the implementation of Soave-Redlich-Kwong equation of state and the validation of the implementation.

Chapter 5 introduces the Mini-LNG plant (Neksået al., 2010) and describes the assumptions to model this plant.

Chapter 6 formulates the optimization problem including all the operational constraints.

Chapter 7 includes the results from the optimization. That is active constraints regions and the behaviour of each degree of freedom throughout the optimization.

Chapter 8 discusses the validity and underlying behaviour of the optimization results. It critically evaluates the model used and algorithm used with their numerical implications.

Chapter 9 includes the conclusions and recommendations for further work.

CHAPTER

TWO

OPTIMIZATION THEORY

This chapter provides an overview of the theory behind the optimization method and possible numerical challenges that might arise throughout the formulation and solution of the optimization problem.

2.1 Overview of non-linear programming

Let us consider the following optimization problem (Nocedal and Wright, 2006):

$$\min_{x} \; f(x) \tag{2.1a}$$
$$c_i(x) = 0, \; \forall \; i \in E \tag{2.1b}$$
$$c_i(x) \leq 0, \; \forall \; i \in I \tag{2.1c}$$
$$\tag{2.1d}$$

where:

x is the vector of optimization variables.

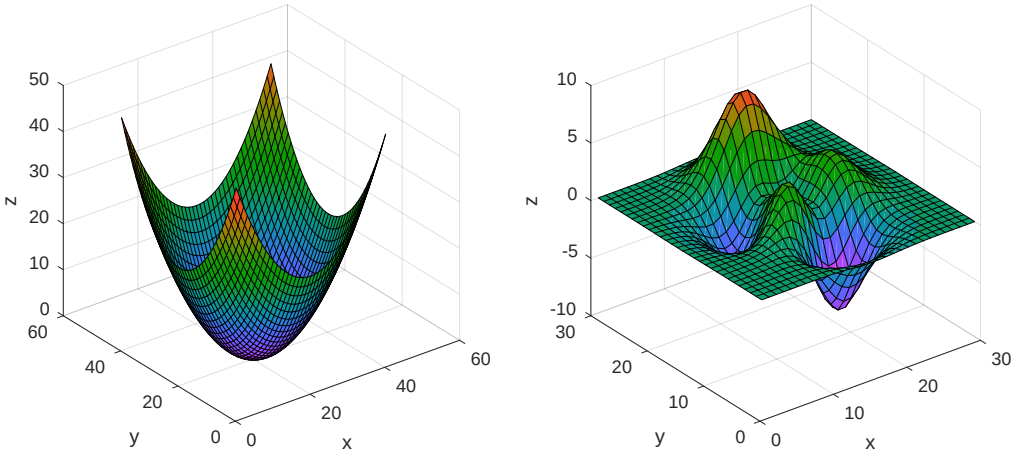
$f(x)$ is the scalar objective function to be minimized.

$c_i(x)$ is the vector of constraints. The functions that set up the restrictions for the possible values of x are as follows:

Equality constrains. These have the form $c_i(x) = 0, \; \forall \; i \in \mathcal{E}$ where $\mathcal{E}$ is the set of equality constraints.

Inequality constrains. These have the form $c_i(x) \leq 0, \; \forall \; i \in \mathcal{I}$ where $\mathcal{I}$ is the set of inequality constraints.

The difficulty of this problem depends on the properties of the objective function and of the constraints. First it is necessary to check if the f is convex (Figure 2.1a). f is convex if its domain is a convex set (one where any two points can be



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

Figure 2.1: Example of different functions.

connected by a straight line) and if for any two points a and b, f lies under the line connecting the points $(x, f(x))$ and $(y, f(y))$. If the constraints and f are convex, and the constraints, $c_i(x) = 0,\ i \in \mathcal{E}$, are linear and the inequality constraints, $c_i(x) \leq 0,\ i \in \mathcal{I}$, are concave, then the problem is convex (Nocedal and Wright, 2006). Convex problems have been widely studied and count with many algorithms for their solution.

For problems, where neither f nor the constraints are linear, $c_i(x) = 0,\ i \in \mathcal{E} \cup \mathcal{I}$, then the problem becomes a non-linear programming (NLP) problem (Figure 2.1b). In these problems convergence is not as straightforward as in the convex case, as there could be several local optimal points for f.

2.1.1 KKT-Conditions

Before explaining the algorithm used in this work, it is necessary to remember the conditions for a local solution. The requirements for a point x^* to be a local solution are known as the *Karush-Kuhn-Tucker conditions*.

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \tag{2.2a}$$
$$c_i(x^*) = 0,\ \forall\ i\ \in \mathcal{E}, \tag{2.2b}$$
$$c_i(x^*) \geq 0,\ \forall\ i\ \in \mathcal{I}, \tag{2.2c}$$
$$\lambda_i^* \geq 0,\ \forall\ i\ \in \mathcal{I}, \tag{2.2d}$$
$$\lambda_i^* c_i = 0,\ \forall\ i\ \in \mathcal{E} \cup \mathcal{I} \tag{2.2e}$$

where $\nabla\mathcal{L}$ is the gradient of the Lagrange function defined as:

$$\nabla_x \mathcal{L} = \nabla f(x^*) - \sum_{i\ \in\ \mathcal{A}(x^*)} \lambda_i^* \nabla c_i(x^*) \tag{2.3}$$

λ^* the vector of Lagrange multipliers and $\mathcal{A}(x^*)$ the active set of constraints at the point x^*. The active set is defined on equation (2.4)

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i(x) = 0\} \tag{2.4}$$

2.2 Types of NLP Algorithms

The conditions shown (2.2) are very important as they are the foundation of several optimization algorithms. The algorithms for constrained non-linear programming can be classified based on the type of solution they get (Nocedal and Wright, 2006).

- **Global:** the minimization is carried out along the whole feasible region.
- **Local:** the algorithm finds a minimum in the feasible region close to the initial estimate.

The difference between global and local optimal points can be seen on Figure 2.2. It should be noted that for local optimization methods it is very important to choose the appropriate initial point.

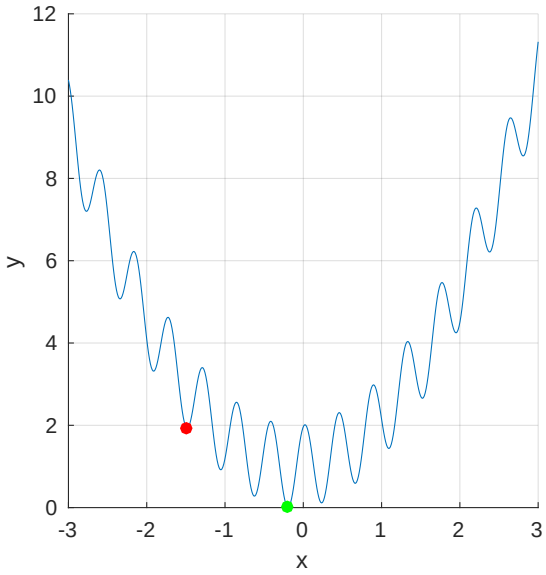


Figure 2.2: Global (green) and local (red) optimums in a function.

This work focuses on the use of local methods. These methods can be classified depending on the mathematical principle used. However, there is not a clear classification of these algorithms. Nocedal and Wright (2006) proposes the following classification for a a general optimization problem (equation (2.1)).

- **Penalty methods:** they modify the objective function by including a penalty term, which accounts for the constraints, as shown on equation (2.5). This allows the problem to be solved as a series of unconstrained optimization problems.

$$f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(x) \tag{2.5}$$

where the parameter $\mu > 0$ is the *penalty parameter*. The problem is solved sequentially for larger values of μ until a desired tolerance is achieved.

- **Barrier methods:** these methods modify the objective function by adding a term that is very small when x is inside the feasible region but tends to zero once x approaches a boundary.

$$f(x) - \mu \sum_{i \in \mathcal{I}} \log c_i(x) \tag{2.6}$$

where $\mu > 0$ is the barrier parameter. In a similar fashion as for the penalty methods, the algorithm gets to a solution by solving successive problems while decreasing the value of μ.

- **Augmented Lagrangian:** in this case the Lagrangian function (equation (2.3)) is modifying by adding a penalty term as in equation (2.5). An example of the augmented Lagrangian for the case where only equality constraints are present on problem (2.1) is shown below.

$$\mathcal{L}_a(x, \lambda, \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i \nabla c_i(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(x) \tag{2.7}$$

In this method values for λ and $\mu > 0$ are fixed. Then a value for x is found by minimizing $\mathcal{L}_a$. This new value of x is then used to update both λ and $\mu > 0$, until reaching convergence.

- **Sequential Quadratic Programming:** this method corresponds to the solution of (2.1) as a quadratic programming problem. The solution of this new problem is the search direction.

$$\min_p \frac{1}{2} p^T W_k p + \nabla f_k^T p \tag{2.8a}$$

$$\text{s.t. } A_k p + c_k = 0 \tag{2.8b}$$

where W_k is the Hessian of the Lagrangian and A_k the constraint Jacobian, for an iteration k.

The selected method for this work is the *interior-point method*. This a **barrier method** with the following advantages:

- This algorithm can handle large and small problems. Due to the way that the interior-point algorithm is implemented in Matlab, it handles large sparse problems efficiently. The problem to be solved in this work is a sparse problem.
- Matlab's implementation can recover from failed steps. This is a very valuable asset for this algorithm. The thermodynamics set up the foundation for modelling this plant.They make this optimization problem very nonlinear. Therefore it is possible to end up in unfeasible steps for throughout the optimization of the problem. Details about the thermodynamics and the model of the plant are shown in chapter 4 and 5, respectively.

2.3 Interior Point Method

This section presents a summary of the interior-point method implementation in Matlab. Further details about this method can be found in the work of Forsgren et al. (2002).

In order to illustrate the interior-point approach, let us define the following problem:

$$\min_x f(x) \tag{2.9a}$$

$$\text{s.t.}$$

$$h(x) = 0 \tag{2.9b}$$

$$g(x) \leq 0 \tag{2.9c}$$

Problem (2.9) can be approximated by using slack variables and a barrier term.

$$\min_{x,s} f_(x) - \mu \sum_i \log(s_i) \tag{2.10a}$$

$$\text{s.t.}$$

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

$$g(x) + s = 0 \tag{2.10c}$$

where $s > 0$ is a vector of slack variables. These variables transform the inequality constraints $g(x) \leq 0$ into equality constraints $g(x) + s = 0$. Additionally a barrier term is added. As mentioned earlier, mu is a barrier parameter, and as it approaches to zero, the approximate solution tends to the solution of the original problem.

The sequence of equality constrained problems (2.10) is easier to solve than the original problem (2.9).

The implementation of the interior point method in Matlab (The Mathworks, 2016a) counts with two alternatives to solve the approximated problem.

- *Direct step:* this is a step that tries to directly solve the KKT conditions (2.2) for the approximate problem (2.10) using a linear approximation. The direct step is defined by equation (2.11)

$$\begin{bmatrix} H & 0 & J_h^T & J_g^t \\ 0 & S\ diag(\lambda) & 0 & -S \\ J_h & 0 & I & 0 \\ J_g & -S & 0 & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ -\Delta y \\ -\Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f - J^T - hy - J_g \lambda \\ S\lambda - \mu e \\ h \\ g + s \end{bmatrix} \tag{2.11}$$

  where

  - H is the Hessian of the Lagrangian of f
  - J_g is the Jacobian of the constraints g
  - J_h is the Jacobian of the constraints h

- $S = diag(s)$
- λ Lagrange multipliers for g
- y Lagrange multipliers for h
- e vector of ones with the same size as g

This is the most computationally expensive step from the interior-point method.

- *Conjugate gradient:* in this step a conjugate gradient(CG) approximation is used within a trust region. The CG steps solve a quadratic approximation (2.12) of the reduced problem (2.10).

$$\min_{\Delta x, \Delta s} \nabla f^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2_{xx} L \Delta x + \mu e^T S^{-1} \Delta s + \frac{1}{2}\Delta s^T S^{-1}\ diag(\lambda)\Delta s \quad (2.12a)$$

s.t.

$$h(x) + J_h \Delta x = 0 \quad (2.12b)$$

$$g(x) + J_g \Delta x + \Delta s = 0 \quad (2.12c)$$

Matlab's algorithm tries a direct step first. If it is not possible, then it uses a CG step.

For each iteration Matlab decreases the value of a merit function (2.13).

$$merit = f_\mu(x, s) + v \left\| (h(x), g(x) + s) \right\| \quad (2.13)$$

where the parameter v can increase in order to push the solution towards feasibility. If for a given step the value of the merit function (2.13) does not decrease, the algorithm rejects that step and attempts a new one.

Matlab's interior-point algorithm can be summarized on Algorithm 2.1

2.3.1 Drawbacks of the interior-point method

This section corresponds to a summary of the advantages and limitations of interior-point methodology.

- *Accuracy:* the solution obtained using the interior-point method can be less accurate than using other methods. This results from the barrier term, as it sometimes does not converge to zero. However, this inaccuracy is trivial in many cases, as it is in the same order of magnitude as the tolerances (The Mathworks, 2016b).
- *Initialization:* as with any other NLP solver, it is necessary to have a careful initialization. The usual way to initialize the algorithm is to use a point that lies in the middle between the upper and lower bounds. Additionally for perturbed problems (problems in which a base case is successively re optimized for a series of perturbations) there are two possible ways of initialization:

Algorithm 2.1: Matlab's interior-point method

```
1  begin
2      Initialize for x_0
3      if x_0 is optimal then
4          Stop
5      else if Stopping criterion are true then
6          Stop      /* Additional stopping criteria include number of
                        iterations or step size */
7      else
8          Try Newton (direct) step (2.11)
9          if Newton step fails then
10             Try CG step (2.12)
11         end
12         Evaluate merit function (2.12)
13         if merit function does not decrease then
14             Try new step (back to line 7)
15         else
16             Update solution
17         end
18     end
19 end
```

  - *Hot start:* an optimal point of the original problem is used as a starting point for the perturbed problem. However, this point will be very close to the boundaries and requires a very small value for μ. Some modifications can be made on this point to make it more suitable and they can make the problem converge in around half the time as with "cold" initialization. Nonetheless, the Matlab algorithm does not allow for these modifications to occur in a straightforward manner (Glavic and Wehenkel, 2004).
  - *Warm start:* a previous optimal solution is not used to initialize the problem given that it is not suitable without modifications. A previous intermediate solution is used instead. In this case the problem is to define which point should be chosen (Glavic and Wehenkel, 2004).

- *Adjustment of the barrier parameter:* adjusting the barrier parameter is a very important part of the interior-point algorithms. There have been extensive discussions about this issue. In the case of the Matlab implementation, there is no information on how this parameter is adapted (Glavic and Wehenkel, 2004).
- *Local method:* it should be kept in mind that this is a local method. Therefore the results from the optimization are expected to be local minima and thus to be found around the initial estimate.

CHAPTER

THREE

PLANTWIDE CONTROL

This chapter is an overview of the plantwide control procedure proposed by Skogestad (2000) and further developed in Skogestad (2004) and Skogestad (2012). The steps within this procedure are explained throughout this chapter. There is a strong focus on the top-down analysis because this will give further insight into the context of this work, as well as the importance of the optimization results. This work will provide a framework for further implementing a complete control structure for the LNG unit analysed on this work.

3.1 Background

The plantwide control procedure is proposed based on the way in which real plants are controlled. In a real plant, control is carried out based on different time scales. By separating each time scale into a layer, it is possible to establish a control hierarchy, as depicted in Figure 3.1.

In Figure 3.1, it is possible to see that the bottom steps correspond to the control layer. This layer covers two time scales, they are: the regulatory (for seconds) and the supervisory control layers (for minutes). The controlled variables provide a connection between these two bottom levels. In this case, the supervisory layer calculates the set points to be controlled by the regulatory layer.

The regulatory layer takes care of the stable operation of the process. In this layer, there are not available degrees of freedom (DOF), due to the fact that the supervisory layer computes the set points for this layer (CV_2 on Figure 3.1). In order to carry out the task of feeding the set points to the regulatory layer, the supervisory layer uses a wide variety of tools, from simple proportional-integral-derivative controllers (PIDs) to advanced control operations, such as cascade, split range or even model predictive control (MPC). Finally, in a similar fashion, the optimization layer sets up the set points for the supervisory layer (CV_1 on Figure 3.1).

To propose a control structure it is necessary to make the following decisions:

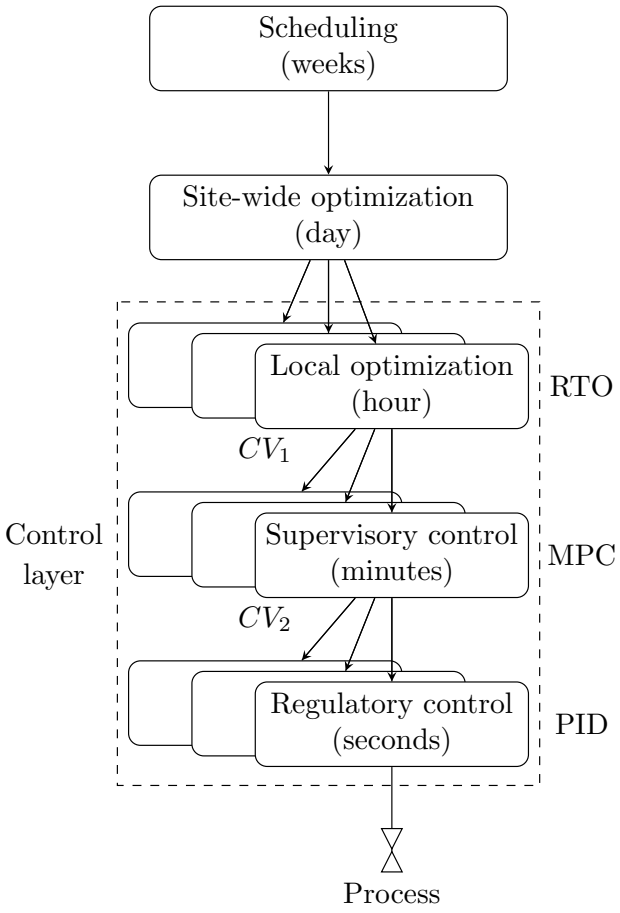


Figure 3.1: Typical control hierarchy in a chemical plant (Skogestad, 2012).

- Selection of manipulated variables(MV) or inputs.
- Selection of controlled variables(CV) or outputs (primary (CV_1) and secondary (CV_2)).
- Selection of measurements or combinations of them.
- Selection of control configuration (pairing between measurements and inputs).
- Selection of controller type (PI, PID, decoupler, etc).

In **?**Skogestad (2004, 2012), a systematic procedure for the design of a plant-wide control scheme has been proposed. This procedure ensures that the plant will have not only a stable operation but also an operation close to optimal in spite of the possible disturbances. In pursuance of this goal, it is necessary to determine which control variables should be kept constant. This is done by quantifying the economic "Loss" (Figure 3.2). The Loss for a given controlled variable is the difference between the optimal value of the cost function when keeping the controlled variable at a constant set point equal to the optimal set point at nominal point and the value of the cost function when re-optimizing and updating the set point to the optimal set point when disturbances occur.

Figure 3.2 shows the loss when two different variables are chosen as controlled variables. The bottom line corresponds to the value of the cost function for the optimized plant given a disturbance. The other two lines $C_{1,s}$ and $C_{2,s}$, correspond to the values

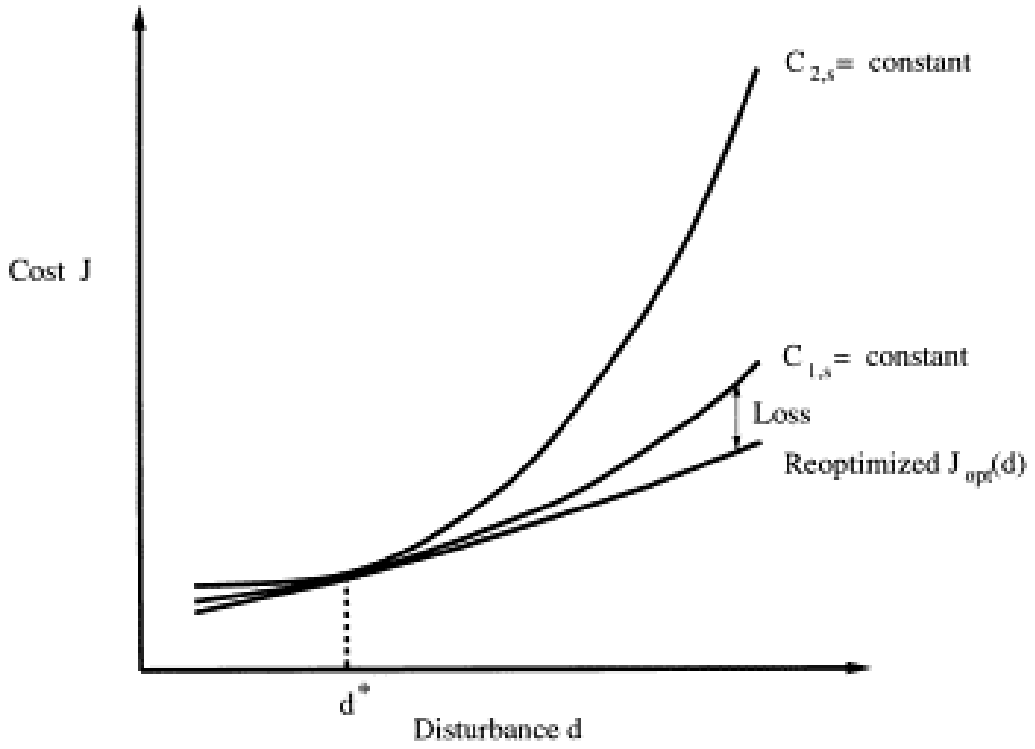


Figure 3.2: Loss imposed by keeping constant set point for the controlled variable. (Skogestad, 2000).

of the cost function for the case, in which the set point for a given variable (in this case 1 and 2, respectively) is constant. It can be seen that the Loss for variable 1 is less than the one for variable 2. An acceptable loss can be achieved by a constant set point value for the variable 1. This means that variable 1 is a *"self-optimizing variable"*.

This approach to plantwide control uses self-optimizing variables throughout the control structure for the whole plant. In this case, the structure will ensure that the minimum loss is achieved for the whole plant when disturbances occur without the need to re-optimize. This concept is known as *"self-optimizing control"*.

3.2 Plantwide control procedure

Based on the concept of Loss, Skogestad (2004, 2012) proposed a systematic approach to determine a control structure which minimizes the plant loss by keeping its operation close to optimal. This procedure is carried out in two main parts: top-down analysis and bottom-up design. A detailed description of the procedure is found on section 3.2.1 and 3.2.2, for the top-down and bottom-up parts respectively. A summary of the steps goes as follows:

- **Top-down analysis**
  This part of the procedure begins by defining the objectives of the operation and making the decisions regarding the manipulated variables or inputs, controlled variables or outputs (primary and secondary) and measurements or combinations of them. For this purpose, it is important to count with a robust and reliable steady state simulation. This simulation will allow to calculate the losses and ensure that the selected variables are the appropriate ones. There are four steps in this part of the procedure.
  - *Step 1. Definition of operational objectives.* This includes the formulation of the cost function and the constraints, which can be operational or quality constraints.
  - *Step 2. Steady state optimal operation and degrees of freedom.* Identification of the steady state degrees of freedom.
  - *Step 3. Identify primary control variables (CV_1).* The primary controlled variables are those which have the largest impact on the economic operation of the plant. These are active constraints and self-optimizing variables. For this step, it is necessary to carry out an steady state optimization and an evaluation of loss with constant set points.
  - *Step 4. Location of the throughput manipulator.* Where the production rate should be set. Its optimal location follows from the previous step but can be changed depending on operating conditions.
- **Bottom-up analysis**
  Once the control objectives and the primary variables have been defined, it is possible to begin to define pairings of variables and potential control strategies. This is carried out by setting up the regulatory layer first and then using it as a base to control the primary variables and thus achieving self-optimizing control. The final steps correspond to the possibility of further improving and validating the overall control strategy.
  - *Step 5. Stabilization and local disturbance rejection. Regulatory layer structure.* This step uses low complexity PID loops to stabilize the operation of the plant. In this stage the secondary controlled variables (those with a small or no economic impact on the plant (CV_2)) are paired with manipulated variables.
  - *Step 6. Structure of the supervisory control layer.* The objective of this layer is to ensure that the primary variables are kept at their optimal value using the set points of the secondary controlled variables or any

unused manipulated variable. In this case, MPC or decentralized control can be used.

- *Step 7. Select the structure for the optimization layer (if needed).* This step intends to identify active constraints and recompute the set points for the controlled variables. The main decision at this point is to find out if it is needed to implement real-time optimization (RTO). RTO is challenging and requires a time consuming implementation.
- *Step 8. Validation.* Once a plantwide control structure has been defined, it may be needed to validate the structure. This can be done by using nonlinear dynamic simulation of the most important parts of the process.

This work will focus on step 2: Steady state optimal operation and degrees of freedom. However, it is important to describe all the steps, so the context for this study is set clear for further work.

3.2.1 Top-down analysis

Step 1. Definition of operational objectives

This step is developed by formulating the cost function, the model constraints and the operational constraints.

The cost function is typically a scalar function J with units [*currency*/s]. The general definition of the cost function is shown in equation (3.1).

$$J = Cost\ of\ feed + Cost\ of\ utilities - Value\ of\ products \tag{3.1}$$

The model constraints can be defined as a system of equations which represent how the plant works. In this step, only a steady state model is required. The model is set up as equality constraints, as shown in equation (3.2b). This can include additional equality constraints such as given flows.

The operational constraints correspond to the ones that must be satisfied for an already built plant to operate. These can include constraints such as minimum or maximum allowed flows, temperatures or pressures. Additionally, any other restriction regarding quality, safety or the environment should be included there. The final form for these constraints is show on equation (3.2c).

The result from this three equations is an optimization problem illustrated in equation (3.2).

$$\min_{u} \; J(u, x, d) \tag{3.2a}$$

$$s.t.$$

$$f(u, x, d) = 0 \tag{3.2b}$$

$$g(u, x, d) \leq 0 \tag{3.2c}$$

Where,

J is the objective function, previously defined in equation (3.1).

u corresponds to the operational degrees of freedom. These are called operational because they only affect the way in which a given plant is operating but does not change the type or number of units.

x represents all the internal states for the model. These are the values that are used to calculate the behaviour of the plant. An example is the temperature or vapour fraction of a given stream.

d is the possible disturbances to the plant. These are the variables that affect the system but cannot be controlled. They can happen as changes in the feed rate, feed composition, or in external conditions such as price specifications.

Step 2. Steady state optimal operation and degrees of freedom

Now that the cost function has been defined, the steady state operation, as well as u,d and x should be established. In order to do so, it is necessary to define the operation modes (Skogestad, 2012).

- *Mode 1. Given throughput*
  In this mode the production is given. The objective of the optimization is then to maximize the efficiency of the process. That is to minimize the costs for a given product outflow.
- *Mode 2. Maximum throughput*
  This mode usually happens when the product prices are high. In this case, it is desired to maximize the production, despite decreasing the efficiency. This mode usually meets a barrier known as "Bottleneck" after which it is infeasible to further increase the production.

Once a mode has been chosen, it is possible to find the degrees of freedom and the main disturbances for the plant. After identifying these two sets of parameters, it is then possible to optimize the operation. The optimization is carried out not only for the nominal operation (without disturbances), but also for the case where disturbances are present. As a result from the optimization it is possible to know, for a given operation point, which constraints are active. The same constraints can be active for different operation points. These sets of points, where the same constraints are active, are known as active constraint regions.

This step can be summarized as follows:

- *Identify steady-state(operational) degrees of freedom*
  It should be noted first that the physical degrees of freedom are different from the steady-state ones. The latter are the ones that have direct influence on the value of the cost function. A general way to identify the potential degrees of freedom is by valve counting. This can also be done using Skogestad (2012) method of potential degrees of freedom. On section 3.3, the identification of the degrees of freedom for refrigeration cycles is explained in depth.

- *Identify important disturbances and their expected range*
  The importance of a disturbance is measured by its impact on the cost function. The main disturbances are related to feed rate and composition, but they can also include external variables such as temperature and pressure of surrounding processes or the environment. Potential changes in the product specifications and in process parameters (such as efficiencies or equilibrium constants) should also be included. Lastly, variations in both product and utilities prices should be considered.

- *Optimize the operation for the expected disturbances*
  At this point, the disturbances are specified and the process is optimized by varying the degrees of freedom, while satisfying the constraints. For a given set of disturbances there will be a set of active constraints. Finding these sets is the main objective for this step.

It should be noted that this procedure can be required several times, as the plant can operate on different modes depending on the market conditions (Skogestad, 2012).

Step 3. Identify primary control variables

The main issue in this step is to decide which variables should be controlled. In order to answer this question, Skogestad (2012) proposes the following rules:

- **Rule 1:** control active constraints.
- **Rule 2:** control self-optimizing variables with the remaining degrees of freedom.

Rule 1 is an intuitive conclusion from the optimization (step 2), as for a given region the active constraints will keep a constant value in order to reach the optimum. Some of these constraints can also be found through physical insights about the process. This kind of constraints can be for either input or output constraints.

The implementation of active input constraints is trivial, as in physical terms, it only requires to fully open or close a valve. On the other hand, active output constraints requires a controller. This, however, is not as straightforward as defining the set point of the controller right at the constraint value. In this case, as a safety measure, it is necessary to include a margin. It will account for any errors (either steady-state or dynamic), so that the output does not violate the constraint (Narrawaf et al., 1991).

This safety margin is known as *back-off*. It can be quantified using equation (3.3). The back-off cannot be too large as it takes the operation farther from the optimum,

thus it is very important to minimize this value. Skogestad (2012) shows specific guidelines in order to select an appropriate value for the back-off.

$$Back - off = |Constraint - Set\ point| \tag{3.3}$$

Once the active constraints have been paired, additional self optimizing variables should be found and paired with the remaining degrees of freedom. This is done in two steps:

- Identification of candidate measurements.
- Selection of primary control variables.

There are several methods to select the primary CVs. This is achieved by determining a measurement or a combination of those, that meets the criteria proposed by Skogestad (2000):

- The optimal value of the CV should not be sensitive to disturbances.
- The CV should be easily controlled and measured.
- The CV should be sensitive to MV variations.
- If there are two or more CVs, they should not be closely related to each other.

These measurements or combinations of them can be found through different methods, which can be either quantitative (Alstad et al., 2009) or qualitative. On Skogestad (2012), lies a summary and explanation of the most important methods.

Once the control structure has been set, the value for the loss (Figure 3.2) can be estimated. Equation (3.4) shows the quantitative calculation for this penalty parameter.

$$Loss = J(u; d) - J^{opt}(u; d) \tag{3.4}$$

Step 4. Throughput manipulator

As stated earlier, defining the production rate is a very important decision. This decision corresponds to the localization of the throughput manipulator (TPM). By locating the TPM, the stream that sets the amount of mass circulating through the plant can be known. The TPM constitutes the core of the regulatory layer structure. Specific studies about the importance and functioning of the TPM are exposed in Skogestad (2012) and Aske and Skogestad (2009).

3.2.2 Bottom-up analysis

Step 5. Regulatory layer structure

This step sets up the regulatory control layer. This layer is the framework over which the self-optimizing variables are controlled. For this step, the secondary control variables (CV_2) are paired with inputs. Here, no degrees of freedom are used, as the set points for the CV_2s are given by the supervisory control layer. It is very important to carefully select the CV_2s, so that the effect of the disturbances

on the primary control variables (CV_1) is as small as possible. By doing this, the back-off is reduced and thus, the plant operates closer to the optimal point. This is further explained in Skogestad (2004) and expanded in Skogestad (2012).

Step 6. Supervisory control layer

This layer will operate on the framework set up by the regulatory layer. The supervisory layer will feed the set points for the CV_2s, so that the plant operates closer to the optimum. There are two alternatives to carry this out:

- *Decentralized control.* It is the simplest alternative and uses mainly single-loop controllers. It is recommended for processes where there is not much interaction and where the active constraints remain constant (Skogestad, 2004).
- *Multivariable control.* It is the more complex alternative and involves multivariable controllers, such as MPC. It is recommended for highly interactive processes.

Step 7. Optimization layer

The core of this step is the recalculation of the active constraints and setting up the set points for the CV_1s. In practical terms, this means to know if manual optimization will suffice or if RTO is needed. However, in many cases, RTO gives little benefit compared to the self optimizing approach and it is recommended not to use it (Skogestad, 2004).

Step 8. Validation

This step might be needed and it involves testing the plant-wide control structure using non-linear dynamic simulation (Skogestad, 2004). The aim of this part is to ensure that the plant wide control structure meets the performance requirements before implementing it in the real plant. The plant-wide procedure can be repeated from any step, and so any previous proposition made may be revised in order to improve the performance.

3.3 Degrees of freedom analysis

This section will explore in detail how the degrees of freedom (DOF) are found for different processes, focusing on the challenges that arise for closed cycles and specifically refrigeration cycles.

The goal of this analysis is to find the DOF that can be used in the economic optimization, which is key when solving the optimization problem. Additionally, it settles the number of steady-state controlled variables (N_{ss}) that need to be chosen. As explained in Section (3.2.1).

The calculation of N_{ss} is based on the following rule (Jensen and Skogestad, 2009): The total steady-state degrees of freedom is obtained by calculating the difference between the number of dynamic manipulated variables (N_{MV}) and the number of degrees of freedom (N_0) with no steady-state effect. Where N_0 includes:

- Manipulated variables with no steady-state effect on the operation, such as a fully open or closed bypass on a heat exchanger.
- Variables without steady-state effect but that need to be controlled, such as a liquid hold up without steady-state effect.

This rule can be summarized in equation (3.5).

$$N_{ss} = N_{MV} - N_0 \tag{3.5}$$

3.3.1 Potential steady-state degrees of freedom

The potential degrees of freedom correspond to the maximum possible amount of DOF for a process. This does not correspond to the DOF that will be used in the actual operation of the plant. However, knowing the potential gives further insight into the process.

Skogestad (2004) proposes a series of guidelines to find the potential steady-state degrees of freedom, in actual plants. These guidelines have been summarized by De Araújo et al. (2007) and further expanded for cyclic processes by Jensen and Skogestad (2009). Table 3.1 summarizes the mentioned guidelines for a selection of common units in a chemical process.

Table 3.1: Potential Steady-State Operational DOF for Typical Process Units (Jensen and Skogestad, 2009)

Process Unit	Potential DOF
feed	1 (feed rate)
splitter	number of exit streams - 1
mixer	0
compressor, turbine, pump	1 (work)
adiabatic flash tank	0 (1)
liquid phase reactor	1
gas phase reactor	0 (1)
heat exchanger	1 (bypass or flow)
column (without heat exchangers)	0 (1+) + number of side streams
valve	0 (1)
choke valve	1
For each closed cycle	
active charge (fluid hold up)	1
composition of fluid	number components -1

Some points have to be further explained from Table 3.1.

- For adiabatic flash tanks, gas phase reactors, columns and valves, it should be noted that the value for the pressure is usually set by the surrounding process, thus being zero DOF. However, one additional DOF should be added per each extra pressure that is independently set and has a steady-state effect.
- For cyclical processes, the active charge or the amount of fluid circulating through it can be modified and this becomes, then a DOF. It should be noted that the possibility to manipulate this during operation may not be available for all cyclic designs.
- In some cyclical processes, the fluid composition can be modified. This is very important for refrigeration cycles. In the same manner as for the active charge, it should be noted that there might be designs in which the composition cannot be changed during operation.

3.3.2 Actual steady-state degrees of freedom

As stated earlier, there is a difference between the maximum DOF and the actual operation DOF. For example a heat exchanger can have no DOF's given that its temperatures are set by other parts of the process and there are no bypasses. Understanding which units are completely defined and which ones have some available DOF is very important, since these actual DOF will be the ones that will be used to keep the operation as close to the optimum point as possible.

The first source of DOF comes from possible MV's. These DOF are the same for all kinds of processes (cyclic or not). For cyclic processes, there may be two additional sources of DOF: active charge and the composition of the circulating fluid. As stated in the previous section, the composition of the fluid can be modified in some processes but this might not be always possible. As for the active charge, a further explanation is shown below in this section. Table 3.2 summarizes both of these considerations.

Table 3.2: Actual Steady-State Operational DOF for Typical Process Units (Jensen and Skogestad, 2009)

Unit	Actual DOF
MV (valve, heat exchanger, compressor, turbine etc)	1
Variables with no steady-state effect:	
Fluid hold ups after active charge (for cyclic processes)	-1

It should be noted that Table 3.2 has a very general approach and that each plant should be considered with the insight from the potential DOF. For example in a linear process, the level in a tank is a variable with no steady-state effect. However, if that tank is the only one present in a refrigeration cycle, this level will have an steady state effect by affecting the amount of refrigerant flowing at a given moment (active charge).

An additional consideration comes from the possible MV's. This should also be carefully examined. A heat exchanger could have all its parameters set and not have any bypasses. In this case, it will not affect the actual DOF count.

Active Charge

The active charge in a closed cycle is analogous to the feed DOF in a sequential process. Jensen and Skogestad (2009) define it as follows *"The active charge is defined as the total mass accumulated in the process equipment in the cycle, mainly in the condenser and evaporator, but excluding any adjustable mass in liquid receivers (tanks)."*

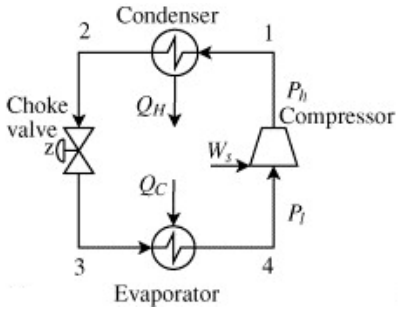


Figure 3.3: Simple refrigeration cycle (Jensen and Skogestad, 2007).

Let us consider a simple cycle such as the one shown in Figure 3.3. Equation 3.6 shows the mass balance for the circulating fluid. In this general refrigeration cycle, there is no tank. However, in practice it is common to have a tank with a variable liquid mass. In order to not affect the pressure of the system, this tank should be working at equilibrium pressure.

$$m_{tot} = m_{evap} + m_{con} + m_{tanks} \tag{3.6}$$

Based on the definition by Jensen and Skogestad (2009), the active charge is defined by equation (3.7).

$$m_{active} = m_{evap} + m_{con} \tag{3.7}$$

Equation (3.8) is obtained, by replacing (3.7) into (3.6). In this expression, the total mass is constant. Therefore, any modification on the hold up of the tank will have an effect on the active charge. This means that the mass in the tank will have an indirect steady-state effect and it can be used for control purposes given the means to change it. It should be noted that adding additional tanks will not increase the number of DOF related to the active charge.

$$m_{tot} = m_{active} + m_{tanks} \tag{3.8}$$

Jensen and Skogestad (2007) summarize the following rules for the active charge.

1. For each closed cycle there is only one DOF related to the active charge, which can be indirectly modified by introducing a variable hold up in the cycle.
2. For each closed cycle there is one hold up that is not required to be explicitly controlled. The biggest capacity is usually the one not directly controlled. In order to avoid overfilling or emptying of the capacities in the system, the other tanks must be controlled in a way consistent with inventory control.

It could be possible to control all the levels in the system, this will lead to explicit control of the active charge. However, as stated in *Rule 2* it is not required to control all of them. Additional tanks can be introduced for the operation and depending on the system one of the following scenarios can hold:

- For pure liquids, introducing additional capacities does not create any extra DOF, because mass can be transferred between the tanks without having any steady-state effect. It should be noted that *Rule 2* still holds and the level for any additional capacity should be controlled.
- For mixed refrigerant cycles, two or more liquid capacities can be used to control the composition of the circulating fluid. In order to do so, at least two tanks with different compositions are needed. This configuration might not be present in every multicomponent design. Moreover, for more complex configurations the maximum possible number of DOF related to tank hold ups is the number of components in the mixture.

CHAPTER

FOUR

THERMODYNAMIC MODEL

4.1 Thermodynamic model

4.1.1 Soave-Redlich-Kwong equation of state

The thermodynamic model used for this work is based on Soave's modification of Redlich-Kwong equation of state (Soave, 1972) (SRK). The model is explained in two parts: one for a single component and the other for a multicomponent mixture.

Single component

SRK model for a pure component:

$$T_r = T/T_c \tag{4.1a}$$

$$P_r = P/P_c \tag{4.1b}$$

$$m = 0.480 + 1.574 * \omega - 0.176 * \omega^2 \tag{4.1c}$$

$$\alpha = (1 + m * (1 - T_r^{0.5}))^2 \tag{4.1d}$$

$$A = 0.42747 * \alpha * P_r/T_r^2 \tag{4.1e}$$

$$B = 0.08664 * P_r/T_r \tag{4.1f}$$

$$Z^3 - Z^2 + Z(A - B - B^2) - AB = 0 \tag{4.1g}$$

$$ln\left(\frac{\phi}{P}\right) = Z - 1 - ln(Z - B) - \frac{A}{B} ln\left(\frac{Z + B}{Z}\right) \tag{4.1h}$$

Where: T is the temperature, T_r is the reduced temperature, T_c is the critical temperature, P is the pressure, P_r is the reduced pressure, P_c is the critical pressure, ω is the acentric factor, Z is the compressibility factor and f is the fugacity. While m,α, A and B are parameters of the model.

The solution of equation (4.1g) provides information of the phase of the stream. If the Z value is close to 0, the phase is vapour. As for values of Z close to 1,

the stream phase is liquid. It should be noted that for one phase the value of the fugacity is equal for both phases. However, when there is phase equilibria this value changes for each phase.

Multicomponent

Multicomponent mixtures are more challenging. For a mixture, the calculations happen in two steps, as follows:

- The first step involves calculating the mixture parameters. For this model, the mixing rules chosen are the van der Waals rules (Prausnitz, John M. et al., 2001). These are based on a geometric average for each parameter (equation (4.2a)) and a molar weighting of these average parameters in order to find the mixture ones (equation (4.2c)).
- The second step uses the mixture parameters calculated from the previous one as if it was a single component mixture to calculate the Z. However, the calculations for the fugacity are carried out for each component in the mixture.

Equation (4.2) summarizes the mixture calculations for the SRK model using van der Waals mixing rules. For further information about multicomponent mixture calculations, Prausnitz, John M. et al. (2001) summarizes extensive research about the thermodynamics of mixtures.

For a component i. Binary A's:

$$A_{i,j} = (A_i * A_j)^{0.5} \quad (4.2a)$$

Mixture A and B:

$$A = \sum_{i}^{NC} \sum_{j}^{NC} x_i * x_j A_{i,j} * (1 - k_{i,j}) \quad (4.2b)$$

$$B = \sum_{i}^{NC} x_i * B_i \quad (4.2c)$$

The compressibility factor is calculated using (4.1g) with the mixture parameters

The fugacity of an i component is calculated as follows:

$$ln\left(\frac{\phi_i}{P}\right) = (Z-1)\left(\frac{B_i}{B}\right) - ln(Z-B) - \frac{A}{B}\left(2C_i\frac{A_i^{0.5}}{A} - \frac{B_i}{B}\right) ln\left(\frac{Z+B}{Z}\right) \quad (4.2d)$$

$$\text{Correction for the fugacity: } C_i = \sum_{i}^{NC} \sum_{j}^{NC} x_j * A_j^{0.5} * (1 - K_{i,j}) \quad (4.2e)$$

Derivative factor

In order to calculate more properties of a mixture using the SRK equation of state, it is necessary to calculate the derivative factor. This factor results from applying thermodynamic relations with the SRK to find additional properties such as enthalpy or entropy (Prausnitz, John M. et al., 2001).

The derivative factor corresponds to the derivative of factor A (equation(4.1f)) for a mixture. Given the mixing rules, this calculation is not as straight forward for a multicomponent system as a single component.

$$\frac{dA}{dT} = -\frac{R}{2}\left(\frac{0.42747}{T}\right)^{\frac{1}{2}} sum \quad (4.3a)$$

where the factor *sum* is:

$$sum = \sum_{i}^{NC}\sum_{j}^{NC} x_i x_j \left(m_j\sqrt{|A_i\left(\frac{T_{c,j}}{P}\frac{T^2}{P_{c,j}R^2}\right)|} + m_i\sqrt{|A_j\left(\frac{T_{c,i}}{P}\frac{T^2}{P_{c,i}R^2}\right)|}\right) \quad (4.3b)$$

Where: A_i and m_i are the values for the parameters A and m from equation(4.1) for an i component.

4.1.2 Parameters

This section contains the values for different parameters used in this work for thermodynamic calculations. Table 4.1 contains critical point information and the acentric factor values.

Table 4.1: Critical properties and acentric factors for the components used in this work (Prausnitz, John M. et al., 2001).

Component	Nitrogen	Methane	Ethane	Propane	n-Butane
Critical pressure ($\times 10^5$) $[N/m^2]$	33.98	45.99	48.72	42.48	37.96
Critical temperature $[K]$	126.20	190.56	305.32	369.83	425.12
Acentric factor	0.037	0.011	0.099	0.152	0.200

Equation (4.4) is used to compute the specific heat. The parameters for equation (4.4) are registered on table 4.2

$$\frac{C_p}{R} = A + BT + CT^2 + DT^3 + ET^4 \quad (4.4)$$

Where: $R = 8.314\frac{kJ}{kmolK}$

Table 4.2: Coefficients for the $\frac{C_p}{R}$ calculation for the components (Prausnitz, John M. et al., 2001).

Coefficient	A	B $\times 10^3$	C$\times 10^5$	D $\times 10^8$	E$\times 10^{11}$
Nitrogen	3.539	-0.261	0.007	0.157	-0.099
Methane	4.568	-8.975	3.361	-3.407	1.091
Ethane	4.178	-4.427	5.660	-6.651	2.487
Propane	3.837	5.131	6.011	-7.893	3.079
n-Butane	5.547	5.536	8.057	-10.571	4.134

4.1.3 Molar volume calculations

In order to calculate more properties (such as enthalpy or entropy), it is necessary to determine the molar volume of the mixture. The calculation shown on equation (4.5) is used for the gas phase. This equation uses the appropriate value of the compressibility, as it is vapour, this value corresponds to the root (equation (4.1g)) closest to 1.

$$V = Z\frac{RT}{P} \tag{4.5}$$

Where: Z is the gas phase compressibility factor for the mixture, calculated using SRK (equation(4.2)).

For the calculation of additional properties (such as enthalpy or entropy), the volume found from equation (4.5) is used. However, this value does not accurately predict the actual molar volume for the liquid phase. In order to circumvent this difficulty, this work uses the correction proposed by Péneloux et al. (1982). The correction corresponds to a modification on the calculation of the molar volume for the liquid phase. Equation (4.6) shows how the molar volume should be modified for the liquid phase according to Péneloux et al. (1982). This correction is only valid for liquid phase, therefore the appropriate value for Z should be used.

For a component i:

$$z_{RA,i} = 0.29056 - 0.08775\omega_i \tag{4.6a}$$

$$c = \sum_i^{NC} x_i \left(0.40768\left(0.29441 - z_{RA,i}\right)\right)\left(\frac{RT_{c,i}}{P_{c,i}}\right) \tag{4.6b}$$

The corrected molar volume of the mixture is then:

$$V = \left(Z\frac{RT}{P} - c\right) \tag{4.6c}$$

Where: $T_{c,i}$ and $P_{c,i}$ are the critical temperature and pressure for a component i, respectively.

4.1.4 Enthalpy calculations

The enthalpy calculation is done in two steps. The first step calculates enthalpy assuming ideal gas and the second step corrects it by calculating the departure value, which takes into account the difference between the real system and the ideal version of it. Equation (4.7) shows the total enthalpy calculation.

$$h = h_{ideal} - h^d_{SRK} \tag{4.7}$$

Ideal gas calculation

The ideal gas enthalpy of the mixture is the sum of the enthalpies of each component calculated by the integral of the heat capacity as shown on equation (4.8).

$$\frac{h_{ideal}}{R} = \int_{T_{ref}}^{T_f} C_p dT \tag{4.8}$$

By combining equations (4.4) and (4.8) for the mixture, the following expression is obtained:

$$\frac{h_{ideal}}{R} = \sum_i^{NC} x_i h_{Ideal,i}$$

$$\frac{h_{ideal,i}}{R} = A_i\left(T_f - T_{ref}\right) + \frac{1}{2}B_i\left(T_f^2 - T_{ref}^2\right) + \frac{1}{3}C_i\left(T_f^3 - T_{ref}^3\right) + \frac{1}{4}D_i\left(T_f^4 - T_{ref}^4\right) + \frac{1}{5}E_i\left(T_f^5 - T_{ref}^5\right) \tag{4.9}$$

The expression on equation (4.9) uses the parameters registered on Table 4.2.

Departure calculation

The departure calculation is very important because it will account for all the non idealities in the system. Its calculation requires the computation of the derivative factor (4.3a) and the actual value for the molar volumen V from equation (4.5) for both phases (gas and liquid phase). Equation (4.10) shows how the departure enthalpy is calculated (Prausnitz, John M. et al., 2001).

$$h^d_{srk} = -\left(\frac{\left(A' - T\frac{dA}{dT}\right)}{B'}\right) log\left(\frac{V}{V+B'}\right) + RT(1-Z) \tag{4.10}$$

Where: $A' = \frac{(RT)^2}{P}A$, $B' = \frac{RT}{P}B$ and A and B are the mixture values from equation (4.2).

4.1.5 Entropy calculation

Similar to the enthalpy calculations, entropy is calculated from both ideal and non-ideal terms. However, for the ideal gas calculation covers the constant pressure

effect, constant temperature effect, and mixing effect. The non-ideal calculation, only requires including the departure term using SRK.

$$s = s_{ideal} - s^{d}_{SRK} \tag{4.11}$$

Ideal gas calculation

The ideal calculations are made in two parts:

Constant pressure effect. This effect is calculated as follows:

$$\frac{s_P}{R} = \int_{T_{ref}}^{T_f} \frac{C_p}{T} dT \tag{4.12}$$

The result shown on equation (4.13) is obtained by replacing equation (4.4) into (4.12).

$$\frac{s_P}{R} = \sum_{i}^{NC} x_i s_{P,i}$$

$$\frac{s_{P,i}}{R} = A_i log\left(\frac{T_f}{T_{ref}}\right) + B_i\left(T_f - T_{ref}\right) + \frac{1}{2}C_i\left(T_f^2 - T_{ref}^2\right) + \frac{1}{3}D_i\left(T_f^3 - T_{ref}^3\right) + \frac{1}{4}E_i\left(T_f^4 - T_{ref}^4\right) \tag{4.13}$$

Constant temperature effect. This effect is calculated as follows:

$$s_T = R\ log(\frac{P}{P_{ref}}) \tag{4.14}$$

Mixture effect. This is accounts for the difference between the sum of entropies of the individual components and the mixture.

$$s_m = \sum_{i}^{NC} x_i log(x_i) \tag{4.15}$$

Total ideal enthalpy. The combined effect is calculated as follows:

$$s_{ideal} = s_P - s_T - s_m \tag{4.16}$$

Departure calculation

Similarly to the departure calculation for enthalpy, it accounts for all the non idealities in the system. Its calculation also requires the computation of the derivative factor (4.3a) and the actual value for the molar volume V from equation (4.5), for both phases (gas and liquid phase). Equation (4.17) shows how the departure enthalpy is calculated (Prausnitz, John M. et al., 2001).

$$s^{d}_{srk} = -\left(\frac{\left(T\frac{da}{dT}\right)}{B'}\right) log\left(\frac{V}{V+B'}\right) - Rlog\left(Z\left(1 - \frac{B'}{V}\right)\right) \tag{4.17}$$

Where: $A' = \frac{(RT)^2}{P}A$, $B' = \frac{RT}{P}B$ and A and B are the mixture values from equation (4.2).

4.2 Flash Calculations

This section covers the vapour-liquid equilibrium (VLE) and the flash calculations for the thermodynamic model. These are very important steps in the model of the process, as there are many streams present which are in equilibrium. In addition, these calculations will allow us to model equipment such as valves and heat exchangers.

4.2.1 Vapour-Liquid Equilibrium

VLE is a largely studied field in chemical engineering. First, it should be remembered that at equilibrium the value of Gibbs free energy (G) is minimized for any given temperature and pressure. For a system to be in equilibrium G must remain constant for any small perturbation, this is shown in equation (4.18).

$$(dG)_{T,P} = 0 \tag{4.18}$$

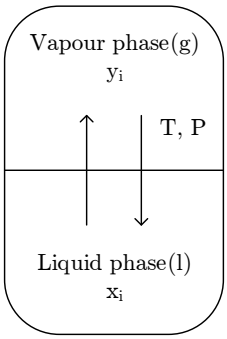


Figure 4.1: Vapour-liquid equilibrium.

Now let us assume a system where both vapour and liquid phases are present (Figure 4.1). In this system we can consider a perturbation in which some small fraction of the liquid phase (l) evaporates to the vapour phase (g). Using equation (4.18) for this situation, the following expression is obtained

$$dG = (G_{g,i} - G_{l,i})_{T,P} = 0 \tag{4.19}$$

where $G_{g,i}$ and $G_{l,i}$ are the partial G for an i component in the vapour and liquid phase respectively. The partial G for an i component is defined as chemical potential (μ_i). From equation (4.19), it can be deduced that

$$\mu_{g,i} = \mu_{l,i} \tag{4.20}$$

There are three fundamental approaches to calculate VLE for real mixtures (Skogestad, 2009). They are as follows:

1. **Based on K-values.** This uses the following expression:

$$K_i = \frac{y_i}{x_i} \tag{4.21}$$

where x_i is the i molar fraction for an i component in the liquid phase and y_i is the molar fraction for an i component in the vapour phase. K is usually a function of the temperature and pressure. It is independent of the composition for ideal mixtures. However for dilute real mixtures it can be calculated using Henry's law.

2. **Based on activity coefficients.** This is a generalization of Raoult's law to non-ideal mixtures. This approach uses an equation of state to calculate the fugacity in the gas phase. For the liquid phase the fugacity is computed as a function of the temperature pressure and the activity coefficient γ_i, which is determined a different model such as *Wilson*, NRTL, UNIQUAC and UNIFAC. This is commonly used for non-ideal mixtures at moderate pressures.

3. **Based on the same equation of state for both phases.** This method uses equation (4.20) and the same equation of state for both phases. This gives

$$\phi_i^V y_i = \phi_i^L x_i \tag{4.22}$$

where ϕ_i is calculated using the equation of state (SRK in this case) for each phase. This gives the following expression

$$K_i = \frac{\phi_i^V}{\phi_i^L} \tag{4.23}$$

The selected approach for this work is the third one. Therefore, the vapour liquid equilibrium will be calculated using equation (4.23).

4.2.2 Flash calculations

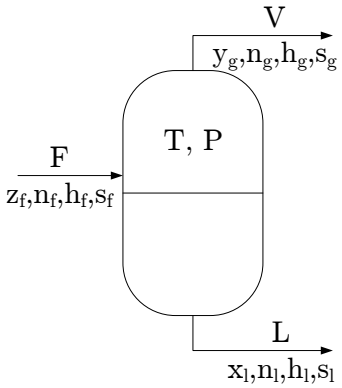


Figure 4.2: Flash tank

Let us assume a flash tank as the one shown on Figure 4.2. Where z_i is the feed composition and n corresponds to the molar flow, h to the enthalpy and s to the

entropy of the respective stream. For this system, the following steady state balances without accumulation can be calculated.

Mass:

$$z_f n_f = x_l n_l + y_g n_g \tag{4.24a}$$

Energy:

$$h_f n_f = h_l n_l + h_g n_g \tag{4.24b}$$

Entropy:

$$s_f n_f = s_l n_l + s_g n_g \tag{4.24c}$$

For the mass balance, the vapour and liquid compositions will be given by the following equation:

$$K_i = \frac{y_i}{x_i} \tag{4.25}$$

where K_i is calculated using equation (4.23) and the fugacities are computed using SRK (equation (4.2d)) for each phase.

Implementation approach

In this work, the VLE calculations are carried out in a EO way (Leguizamon, 2015). Thus, all balances are solved at the same time, as well as the solutions of the equation of state (equation(4.1g)). The algorithm for the solution of flash calculations, given temperature and pressure (TP) or enthalpy and pressure (PH), is shown on Algorithm 4.1.

Algorithm 4.1: Flash calculations: TP-flash and PH-flash (within brackets).

```
1 begin
2 |  Input: T, P, z, (h)
3 |  Solve simultaneously: Mass balance, Cubic equation of
  |    State,Vapour-liquid equilibrium (Energy Balance)
4 |  Output: x, y, v_f, (T)
5 end
```

Further information about the flash calculations algorithm can be found in Leguizamon (2015). However, it is important to make some remarks.

- These calculations require careful initialization. They require initial estimates to be as close as possible to a solution, otherwise the calculations might not converge or provide unreliable results.
- The model in this work is entirely equations oriented. This means that this set of calculations are solved simultaneously for all units at the same time. The explanation here is an specific example.

- It is not advised to use this algorithm when the conditions are very close to the boundary between one and two phases. Using this specific approach without modifications in this area leads to numerical instabilities and is unreliable (Leguizamon, 2015).

Smoothing

As mentioned in the last remark, the previous approach does not lead to reliable results for points close to the boundary between one and two phases. The reason for these instabilities is the discontinuity while calculating the K value. To circumvent this problem, the smoothing proposed by Kamath et al. (2010) is used. The function (equation (4.25)) is modified so the discontinuity becomes continuous and thus, easier to solve. This uses a parameter β which relaxes the function in a similar way as illustrated on Figure 4.3.

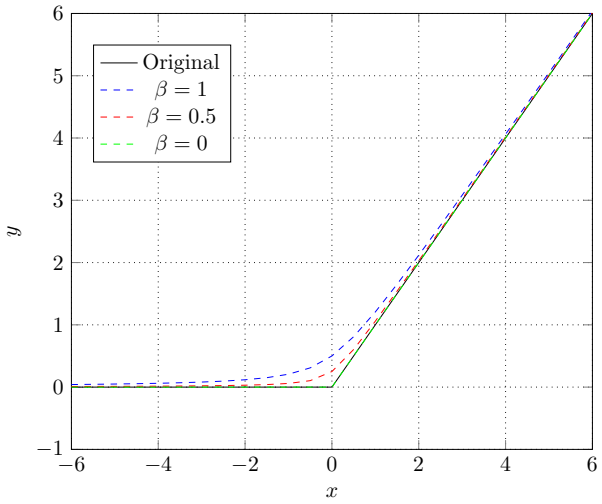


Figure 4.3: Smoothing of a non continuous function for different values for β (Gopal and Biegler, 1999)

The mathematical details of this relaxation are found in Gopal and Biegler (1999). It should be noted that the problem on Figure 4.3 is not the same as equation (4.25). For this specific case of VLE, Kamath et al. (2010) propose an adaptation of this curve smoothing for the discontinuity in the *K-value* calculation.

$$y_i = \beta K_i(P, T, x)x_i \tag{4.26a}$$
$$-s_L \leq \beta - 1 \leq s_v \tag{4.26b}$$
$$n_V \leq 0 \ \perp \ s_V \geq 0 \tag{4.26c}$$
$$n_L \leq 0 \ \perp \ s_L \geq 0 \tag{4.26d}$$

where n is the molar flow, s is the phase slack, β is the relaxation parameter and K_i the *K-value* for an i component. This formulation is used for situations when one phase can disappear, that is when the desired point is in the proximity of the discontinuity.

- The parameters s_L and s_V, are slack variables that count for the phase that is disappearing. For a vapour only case ($n_L \leq 0$), a slack variable s_L is used.
- The constraints $-s_L \leq \beta - 1 \leq s_v$, $s_V \geq 0$ and $s_L \geq 0$ work together in such a way, they ensure the value for β is 1 when converged. For $\beta = 1$, equation (4.26b) becomes $y_i = \beta K_i(P, T, x)x_i$, which is the same as the not relaxed constraint.

For this work the previous formulation (equation(4.26)) can be simplified. Based on the knowledge of the process operation, there are no refrigerant streams operating close to the bubble point. Therefore, it is not required to implement the phase change from liquid only to two phase. This simplified formulation is summarized on equation (4.27).

$$y_i = \beta K_i(P, T, x)x_i \tag{4.27a}$$
$$\beta + s_L \leq 1 \tag{4.27b}$$
$$n_L \leq 0 \ \perp \ s_L \geq 0 \tag{4.27c}$$

Equation (4.27) has two different convergence conditions:

- When $n_L \geq 0$, two phases are present. Thus convergence is achieved once $s_L = 0$ and $\beta = 1$. In other words, when there are two phases present the phase equilibrium should be met, that is:

$$y_i = K_i(P, T, x)x_i \tag{4.28}$$

- When $n_L = 0$, only vapour is present. In this case convergence can be achieved without strictly meeting the phase equilibrium constraint:

$$y_i = \beta K_i(P, T, x)x_i \mid \beta \leq 1 \tag{4.29}$$

Complementarity implementation

Complementarity constraints, such as equation (4.27c), can be implemented in different ways depending on the algorithm and program used to solve the optimization problem (Benson et al., 2002). In this work Matlab is used with the interior point method. Equation (4.31) shows the implementation of the complementary constraints for a given problem (Equation(4.30)).

Let us define the following problem:

$$min \ \phi(w) \tag{4.30a}$$
$$s.t. \ g(w) \leq 0 \tag{4.30b}$$
$$0 \leq x \perp y \geq 0 \tag{4.30c}$$
$$\tag{4.30d}$$

where $\phi(w)$ is a dummy optimization function, $g(w)$ is the set of constraints corresponding to the model, $w = [x^T, y^T, z^T]^T$ is the vector of optimization variables in which x and y are the variables involved in the complementary constraints and z is the rest of the optimization variables.

$$min\ \phi(w) + px^T y \tag{4.31a}$$
$$s.t.\ g(w) \leq 0 \tag{4.31b}$$
$$x, y \geq 0 \tag{4.31c}$$

In this case the complementarity constraints are included in the objective function as a penalty term, where p is the penalty term. This penalty formulation of the constraints ensures the complementarity conditions as both variables are maintained positive (equation (4.31c)) and at least one of them must approach to zero (equation (4.31)). This approach was selected because of its robustness and reliability in conjunction with the available NLP solvers (Kamath et al., 2010).

Additional remarks

The previous sections explained details of implementing the flash calculations solving all equations simultaneously. This algorithm is based on the one proposed by Leguizamon (2015).

It should be noted that not all the flash calculations in this process require relaxation. For this reason, the algorithm has been expanded in such a way that it is possible to toggle the use of relaxation depending on the requirements of the system. As the conditions of the streams are well known before modelling, it is possible to apply the constraint relaxation only to those cases where smoothing is certainly needed. This allows relaxation of the overall model by reducing the total amount of constraints that should be met, while still obtaining a reliable solution. The scripts corresponding to this calculation are registered on section A.1.2.

4.3 Validation

The validation of the implementation of thermodynamic model in Matlab is carried out by comparing the results from a compressor with those obtained in Aspen Hysys®. On page 6.10, Prausnitz, John M. et al. (2001) mentions the impact of selecting an equation of state when calculating compressors. Additionally, a compressor is a unit that requires both enthalpy and entropy calculations. Therefore, a compressor was selected as the unit operation to validate the model.

4.3.1 Vapour compression

Algorithm

The algorithm for vapour compression calculations has the following structure:

Algorithm 4.2: Vapour compression

begin
  Define input parameters: $z, P_{in}, P_{out}, T_{in}$
  Calculate input h_{in}, s_{in}
  Calculate T_{out}, h_{out} isoentropic output $s_{out} = s_{in}$
  Calculate $W_c = \dfrac{h_{out} - h_{in}}{\eta}$
  Calculate $h_{out,real} = h_{in} + W_c$
  Calculate $T_{out,real}$ from $h_{out,real}$
end

Parameters

These three scenarios have been chosen based on the criteria of usefulness and re usability.

- The first one, uses similar conditions to the ones in Neksået al. (2010). Describing the behaviour for that specific set of conditions is the main goal of this implementation.
- The second scenario uses the same composition but increases both temperature and pressure to verify if the model can predict a less ideal scenario.
- The last case corresponds to a heavier composition, higher pressures and temperatures. The logic for this case is similar to the second one. But it also verifies if the model can predict the behaviour for a heavier mixture; which behaves further from ideality.

Table 4.3 shows the values for the different conditions for each case.

Results

Each case was evaluated in Aspen Hysys®and the algorithm implemented in MATLAB. Table (4.4) contains the comparison for the outlet temperature and the compressor work.

4.3.2 Binary mixtures

This section covers a comparison between the Aspen Hysys®and Matlab enthalpy calculations. An energy balance was carried out for a binary mixture with varying

Table 4.3: Conditions for the different compressors for validation

Parameter	Case 1	Case 2	Case 3
P_{in} [bar]	3	20	30
P_{out}[bar]	20	30	40
T_{in} [K]	231.15	300	400
η[%]	100	100	100
z_{N_2}	0.1074	0.1074	0.1000
z_{C_1}	0.3292	0.3292	0.2000
z_{C_2}	0.4096	0.4096	0.1000
z_{C_3}	0.1345	0.1345	0.3000
z_{C_4}	0.0193	0.0193	0.3000

Table 4.4: Results from the evaluation of different compressor cases

Case	Case 1			Case 2			Case 3		
Calculation	MATLAB	Hysys	Error[%]	Matlab	Hysys	Error[%]	Matlab	Hysys	Error[%]
T_{out}[K]	331.6	330.9	0.222	324.5	324.3	0.076	415.16	416.15	0.238
W_c[kJ]	4176	4171	0.111	935.2	934.7	0.0532	840.5	904	7.020

temperature and pressure, in an isobaric and isothermal case for each phase (vapour only and liquid only). The ranges for temperatures and pressures are registered in Table 4.5. In the case of the isothermal calculations for a given phase T_{phase}, the following points are used to carry out the energy balance:

- Constant input stream with T_{phase} and P_{min} or P_{phase} and T_{min} for the isobaric case.
- Output stream with T_{phase} and P, where P varies from P_{min} to P_{max}. Or P_{phase} and T, where T varies from T_{min} to T_{max}.
- Both of these comparisons where carried out for liquid only and vapour only.

To illustrate the kind of analysis carried out, let us use the case shown on Figure 4.4a. In this case, the initial state corresponds to $P = 1\ bar$ and $T_{min} = 220.15\ K$; these two values have been set, so that the initial phase is vapour only. Then the output stream is calculated by modifying the temperature from $T_{min} = 220.15\ K$ to $T_{max} = 423.15\ K$ (this range is also selected for the fluid phase to remain constant throughout the interval) and keeping the pressure constant at 1 bar. Then the energy balance between the two streams is carried out in both Matlab and Aspen Hysys®. The error is calculated by comparing the difference between the two implementations to the value obtained using Aspen Hysys®.

Table 4.5: Composition for a binary mixture enthalpy calculation

	Liquid	Vapour
x_{C_2}	0.5	0.5
x_{C_3}	0.5	0.5

Table 4.6: Ranges for the isobaric enthalpy balance calculation for vapour and liquid phase

	Liquid phase	Vapour phase
$T_{min}[K]$	123.15	220.15
$T_{max}[K]$	223.15	423.15
$P_{constant}[bar]$	20	1

Table 4.7: Ranges for the isothermal enthalpy balance calculation for vapour and liquid phase

	Liquid phase	Vapour phase
$P_{min}[bar]$	1	1
$P_{max}[bar]$	16	16
$T_{constant}[K]$	293.15	133.15

Results

The results from these comparisons are registered in Figure 4.4

As it can be seen from Figure 4.4, the results from Matlab and Aspen Hysys®do not differ more than 6%. This validates the implementation of the thermodynamic model for a single phase for a broad range of temperatures and pressures in a binary mixture.

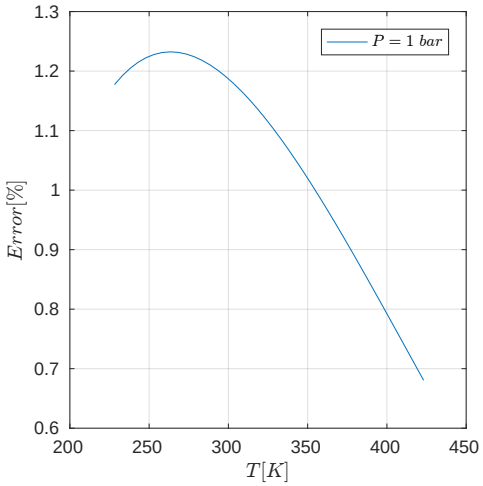
The error becomes considerably larger for low liquid phase temperatures (Figure 4.4c). This results from the high pressure and low temperature conditions that prevent an ideal liquid mixture. Nonetheless, the error of the calculation between Matlab and Aspen Hysys®is not further than 2%. As the conditions of the process studied will rarely fall below 130 K, it is safe to use this implementation.

It should be kept in mind that the following scenarios lead to more non idealities and thus a higher error on the results for any implementation of the SRK equation of state.

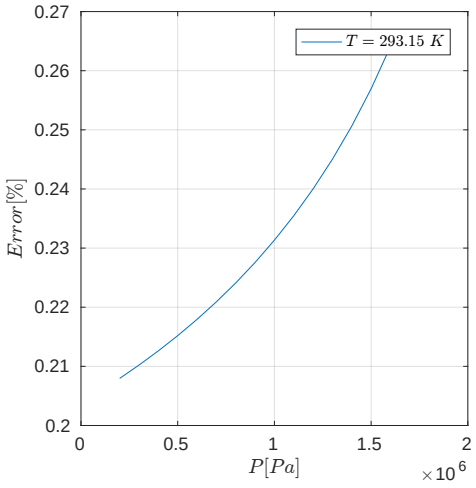
- *Heavier components.* When a mixture has higher proportions of heavy components, its behaviour is less ideal.
- *High pressures and temperatures.* At these conditions the particles of gas take considerably more volume than the ideal gas predictions. Thus making its properties harder to predict.
- *Low temperatures.* At low temperatures the effect of the intermolecular interactions of the fluid become more representative as the fluid particles count with considerably less kinetic energy.

4.3.3 Flash calculations

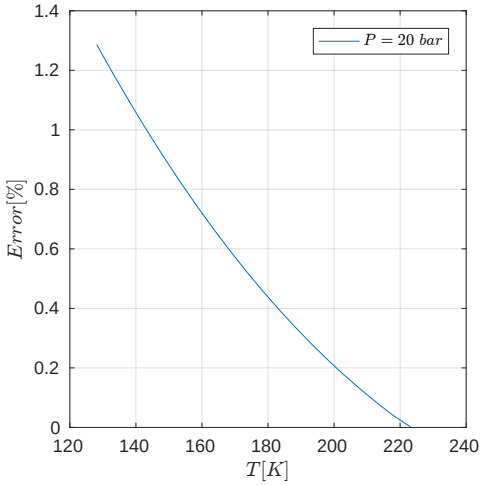
The validation of the flash calculations involves again a comparison between the implementation results and those from Aspen Hysys®. In this case, the calculations are carried out in two stages. The first one is a TP flash of a stream that is vapour



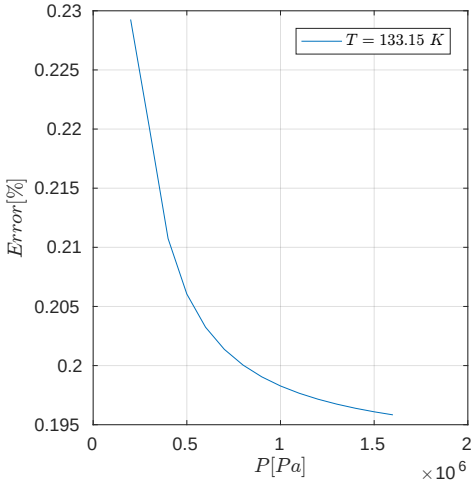
(a) Vapour phase, constant pressure



(b) Vapour phase, constant temperature



(c) Liquid phase, constant pressure



(d) Liquid phase, constant temperature

Figure 4.4: Error in enthalpy calculations compared to Aspen Hysys®

only. The second stage corresponds to several PH flashes after removing different amounts of energy from the previously calculated stream. This approach allows us to see the behaviour of the PH calculations in different points, such as vapour only, the dew point and the two phase area. The initial conditions for the TP flash are registered on Table 4.8.

Table 4.9 shows the results from the MATLAB implementation compared to those obtained using Aspen Hysys®.

It should be noted that on Table 4.9, there are results for liquid composition even when the mixture is vapour only. These results are an important part of the Matlab implementation, as they are used as initial estimate for the following point. In other words, that liquid composition lacks physical meaning but is an initial point for the next calculations.

It can be seen from these results (Table 4.9), that there is not a large difference between the flash calculations using Aspen Hysys®and the Matlab implementation.

Table 4.8: Initial point for flash calculations

$P[bar]$	3
$T[K]$	270
Mole fraction	
z_{N_2}	0.1074
z_{C_1}	0.3292
z_{C_2}	0.4096
z_{C_3}	0.1345
z_{C_4}	0.0193

Table 4.9: Results for flash calculations validation

	Matlab	Aspen Hysys®	Matlab	Aspen Hysys®	Matlab	Aspen Hysys®
$Q_R[kj]$	1000	1000	2000	2000	3000	3000
$T[K]$	248.14	248.3	228.15	228.2	222.16	222.2
v_f	1	1	0.9941	0.9947	0.9547	0.9556
x_{N_2}	0.0058	N/A	0.0009	0.0007	0.0010	0.0008
x_{C_1}	0.0560	N/A	0.0116	0.0111	0.0132	0.0128
x_{C_2}	0.3455	N/A	0.1959	0.1922	0.2424	0.2387
x_{C_3}	0.3548	N/A	0.4246	0.4257	0.4928	0.4951
x_{C_4}	0.2379	N/A	0.3670	0.3702	0.2506	0.2526
y_{N_2}	0.1074	0.1074	0.1080	0.1080	0.1124	0.1124
y_{C_1}	0.3292	0.3292	0.3311	0.3309	0.3442	0.3439
y_{C_2}	0.4096	0.4096	0.4109	0.4108	0.4175	0.4175
y_{C_3}	0.1345	0.1345	0.1328	0.1329	0.1175	0.1177
y_{C_4}	0.0193	0.0193	0.0172	0.0174	0.0083	0.0085

The differences are less than 5%. Therefore we can confidently use the Matlab implementation for modelling of flash calculations.

4.4 Closing remarks

This chapter examined in detail the different challenges when implementing a thermodynamic model. To validate our implementation the results from Matlab algorithm were compared with those obtained using Aspen Hysys®. The difference between both results was very small for both single component, mixtures and flash calculations. It should not come as a surprise, that the results are not exactly the same. This has been pointed out before by several authors (Whiting, 1996; Sadeq et al., 1997; Dohrn and Pfohl, 2002).

As not all the parameters and equations used in Aspen Hysys®are open, it is very difficult to point out the exact sources of the discrepancies between implementations. The common sources for these differences, when using the same model, as summarized by Whiting (1996), go as follows:

- *Model formulation.* Thermodynamic models can be formulated in different ways. For instance, the same SRK model can be parametrized in different ways. This can be appreciated by comparing the formulations presented by

Bruce, E Poling, John M. Prausnitz (1987) and Prausnitz, John M. et al. (2001).

- *Model solution.* In a similar fashion, there exists a plethora of numerical methods which can be used for the solution of the thermodynamics. Each process simulator uses their own implementations (Whiting, 1996) and each model counts with different numerical properties which give insight of how the model can be solved (Gani et al., 2006).
- *Model parameters.* The different simulators may have different values for critical properties.

The addition of these effects can lead to considerable differences between modelling results. Thus, it is expected to find small differences between the Matlab and Aspen Hysys®implementations. Nevertheless, the comparisons shown in this chapter are good evidence that the Matlab results are reliable enough for this work.

It is worth mentioning that the results from this work are intended for plant-wide control optimization. The main result from this optimization is to find the trend that follows the different active constraints when the system is perturbed. Therefore the results do not require 100% accuracy, but instead precise enough for the trend to be noticed.

The relaxation flash calculation is the highest difference from all the comparisons carried out. As seen earlier, the biggest problem for these calculations lies in the handling of numerical problems that come with the discontinuity in the fugacity coefficient when transitioning from one to two phases and the implementation of complementarity constraints. Overcoming these numerical problems is a very extensive research area, and different solutions have been proposed along more than 10 years (Gopal and Biegler, 1999; Kamath et al., 2010). A new solution for this numerical problem is outside the scope of this work. The actual implementation will be used as it provides sufficient robustness for the optimization. In order to increase the reliability of the results, some additional operational constraints might be required.

In short, it is possible to conclude the thermodynamic model functions are reliable and robust enough to give a strong base to trust the optimization results.

The implementation of thermodynamics and flash calculations is registered on Appendix A.

CHAPTER

FIVE

PROCESS DESCRIPTION AND MODELLING

5.1 Mini-LNG

The "mini-LNG" on board plant is depicted on Figure 5.1. This plant is a solution for the problem of boil-off gas (BOG) on LNG carriers. The goal of this system is to minimize the LNG losses during transportation.

The scheme illustrated on Figure 5.1 corresponds to a refrigeration cycle with an internal refrigeration loop. This process only uses standard plate heat exchangers, which only allow two parallel flows. The refrigerant is split in two streams. The high pressure gas stream is cooled in unit HA-02 and then condensed in HA-04. The condensed stream is split in two streams, which work as cold streams for units HA-04 and HA-07. Unit HA-07 is the one that condenses the BOG while HA-04 allows internal refrigerant cooling. The cold output streams from HA-04 and HA-07 are then mixed with the decompressed liquid stream coming from the separator. This final mixture is then warmed until evaporation in unit HA-02. The output from unit HA-02 is then sent to the mixed refrigerant compressor.

Before arriving the compressor, the output from HA-02 is slightly heated by the environment. The mixed refrigerant compressor mixes the gas stream with lubricant. After the compression is done the lubricant is then removed. Superheated compressed gas stream is sent to the desuperheating stage composed by a sea water cooler (MR precooler) which cools the refrigerant to 35°C. The refrigerant is further cooled using propylene to -35°C before entering the phase separator. The separation temperature can change to shift the equilibrium and thus modify the output compositions and flows from the separator unit.

Before condensing the BOG there are two stages of cooling. In a similar way as for the refrigerant, the BOG is initially cooled using sea water and then further cooled using propylene. Once it has been precooled, the BOG is then fed to unit HA-07 where it is condensed and sent back to the LNG storage tanks.

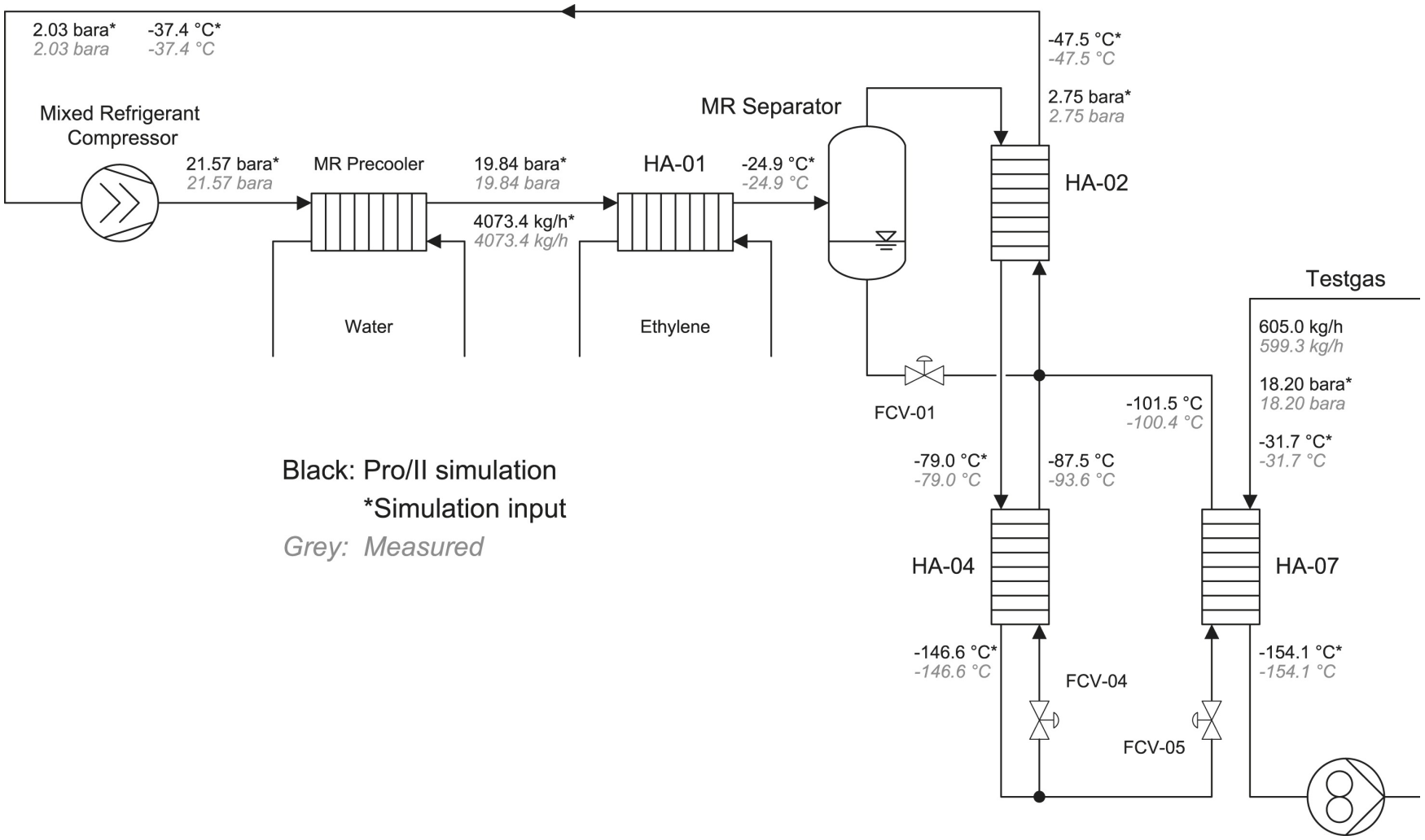


Figure 5.1: Flow diagram of Mini-LNG plant, with measured and simulated results. (Neksået al., 2010).

Operational considerations

Neksået al. (2010) introduces some preliminary operation considerations, which are relevant for this stage of modelling.

- In practice the compressor allows contact of the lubricant with the gas. This cools the refrigerant by a small margin. Additionally, it is required to separate the lubricant from the gas and this step. However, sometimes it is possible that traces of lubricant mix with the refrigerant. For this project, these effects of the lubricant on the refrigerant will not be considered, as their effect is insignificant.
- Between HA-02 and the compressor there is a long connection in which the refrigerant gets warmed by the environment. This heating is small but non-trivial. This incoming heat from the environment is another disturbance on the system. In order to account for it, a heat exchanging step is modelled between HA-02 and the compressor.
- The temperature difference in the cold pocket can be achieved by modifying the refrigerant composition. For this work, the refrigerant composition is assumed to be constant.

5.1.1 Base case

Most of the information about the operating conditions of the plant are given by Neksået al. (2010) and can be seen on Figure 5.1. Nonetheless, they are insufficient to model the plant.

The following list summarizes the conditions for the base case.

- The refrigerant composition used is registered on Table 5.1.

Table 5.1: Refrigerant composition

Component	Mole fraction
N_2	0.1074
C_1	0.3292
C_2	0.4096
C_3	0.1345
C_4	0.0193

- The initial refrigerant flow is 4073 $\frac{kg}{h}$ (Neksået al., 2010) or (with the given composition (Table 5.1)) 0.0401 $\frac{kmol}{s}$.
- The initial split flow for the cold pocket is 50%.
- The heat exchangers are modelled without a pressure loss. The outputs from the valves are all at 3 *bara* for the base case. The discharge pressure for the compressor is 20 *bara* for the base case. These two assumptions mean that there are only two pressures to define in the system: the low (output from the valves) and the high (compressor discharge).

- The base case uses the temperatures registered on Table 5.2.

Table 5.2: Stream temperatures

Stream	Temperature[$°C$]
2	35
3, 4, 5	-25
6	-75
8, 9, 10	-146.6
17	-33

5.1.2 Additional assumptions

The following assumptions are made so that the model can be simplified while keeping its reliability.

- The formulation of HA-07 is built on the basis that the output is always on specification. In this way the BOG side of HA-07 is totally defined for all optimization runs. The inlet BOG side is set up by the perturbations and the LNG output by the specification. On the refrigerant side, only the input is known (stream 11) but having the duty calculated from the BOG side the output can be easily calculated (stream 14).
- The pre-cooling stages for the BOG are not considered in this model. For optimization purposes, the perturbations on the system will be applied directly on the conditions of the BOG entering the unit HA-07.
- Between HA-02 and the compressor there is a long connection in which the refrigerant gets warmed by the environment. This heating is small but non-trivial. This incoming heat from the environment is another disturbance on the system. In order to account for it, a heat exchanging step is modelled between HA-02 and the compressor.
- In order to avoid the non-linearity associated with the solution of complementarity constraints from the thermodynamics, some phases are assumed to remain constant such as stream 2 and 8. This removes the need to solve VLE calculations.

Figure 5.2 shows the simplified scheme used to build the model, including the previously mentioned assumptions and the phases for each stream.

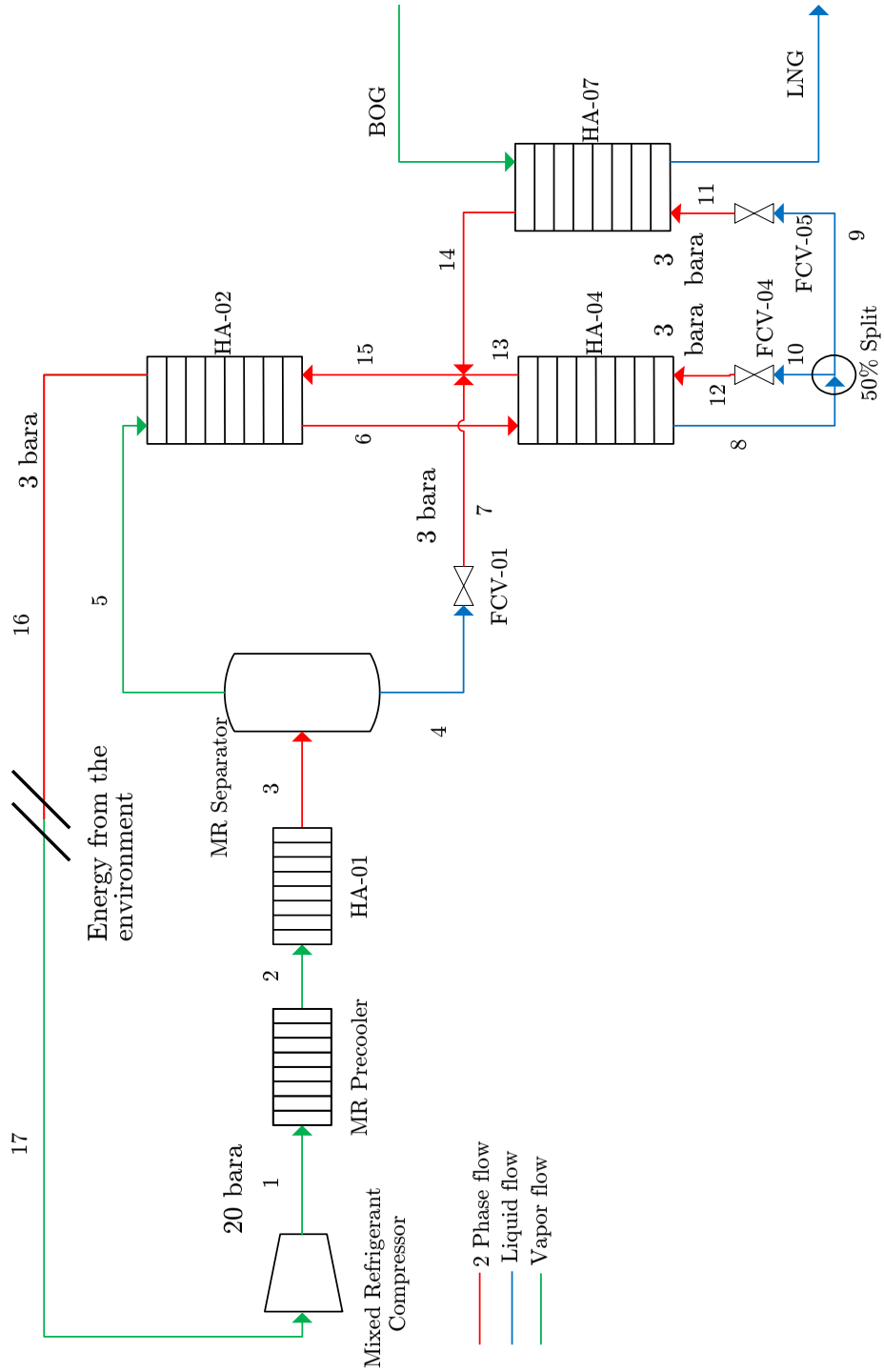


Figure 5.2: Flow diagram of the model of the Mini-LNG plant.

5.2 Degrees of Freedom Analysis

To define the variables for optimization, it is necessary to determine the steady state degrees of freedom of the plant. These can be calculated by the difference between manipulated variables and the operation variables without effect on the steady state operation. This is summarized on equation (3.5).

$$N_{ss} = N_{MV} - N_0$$

The number of manipulated variables can be found by the difference between the maximum number of DOF and the lost DOF, this is shown on equation (5.1).

$$N_{MV} = N_{ss}^{max} - N_{Lost} \tag{5.1}$$

The lost degrees of freedom are given by operational constraints and model assumptions, such as fixed temperatures or compositions.

Using Table 3.1 and the PDF of the plant (Figure 5.1), it is possible to find the maximum number of DOF in the plant. Table 5.3 sums up these results.

Table 5.3: Maximum DOF.

1	Compressor
5	Heat exchangers
3	Valves
4	Compositions
1	Active charge
14	N_{ss}^{max}

The lost DOF for this plant are summarized as follows.

- *Fixed refrigerant composition.* This means the composition will not be manipulated and thus the DOF related to it are lost.
- *MR Precooler.* The output of this unit is assumed to always be at $35°C$. This results from the units use of seawater for cooling.
- *HA-02 and HA-04.* These exchangers are completely defined from both sides, as the temperature for stream 6 and 8 are fixed. Additionally, these units have no bypasses. Therefore the degrees of freedom for these units are lost.

Temperature 8 is set so that the output is always liquid and temperature 6 is set so that the system can be solved.

Table (5.4) summarizes the lost DOF and the amount of manipulated variables.

In this plant there is one capacity that needs to be controlled, the level of the MR separator. Thus, it is possible to calculate the actual number of steady state DOF (Table 5.5).

The actual DOF and their variables are summarized on Table 5.6.

Table 5.4: Lost DOF and manipulated variables.

-2	HA-02 and HA-04
-1	MR Precooler
-4	Fixed compositions
-7	N_{Lost}
14	N_{ss}^{max}
7	N_{MV}

Table 5.5: Actual DOF.

7	N_{MV}
-1	N_0
6	N_{ss}

Model DOF

For modelling, the system not all the steady state DOF can be used in the way they are stated on Table 5.6.

First the duty for heat exchanger HA-07 is used as a perturbation for the system, given that some of the inlet conditions of the BOG can change.

The remaining DOF have the following considerations.

- *Compressor work.* This is not directly manipulated in the model, the output pressure is used instead.
- *HA-01.* For this unit the value of the output temperature T_3 is used to set up the constraints but the heat duty is the input to the model.
- *Valves FCV-04 and FCV-05.* These valves will be used at the same output pressure. One of this valves will set up the split ratio for the flow of stream 8 and the other will set up the output pressure. In the model, the values of the output pressure and split flow ratio are specified directly.
- *Active charge.* This corresponds to the fluid hold up in all the units and it is related to the actual refrigerant flow. In this problem there is not enough information to make a detailed model of each unit that calculates the hold up. For this reason, a direct specification of the refrigerant flow is used instead of the active charge.

Perturbations

There are two main sources of perturbations.

- *Inlet BOG.* The incoming BOG can have changes in the inlet conditions. These can be variations in the composition, temperature or flow. All of these variations are reflected in the duty and dimensioning of HA-07. The chosen variations for this study are only temperature and flow. The composition changes were not considered as their change is not large enough compared to the effect

Table 5.6: Steady state DOF.

1	Compressor work
2	Valves (FCV-04, FCV-05)
2	Heat duties (HA-01, HA-07)
1	Active charge
6	N_{ss}

of the other two variables. By including perturbations in the composition, the dimensions of the analysis increase by a factor of $NC - 1$, where NC is the number of components.

- *Heat from the environment.* This unit is going to be on the sea and it is subject to changing environmental conditions. This effect can be considered as an additional heat source to the system. It is included in the energy balance step between streams 16 and 17.

5.3 Screw Compressors

Screw compressors are an specific kind of rotary positive displacement compressors (Bahadori and Bahadori, 2014).

Compression takes place by the simultaneous use of two joined rotors. Power is then applied to the orbiting rotor. This rotor starts to move out of mesh from the fixed rotor. A void is created by this movement and gas is then taken in. The male rotor keeps on turning in such a way that the space between rotors is filled with gas. Once it is filled, the turning motion reduces the space between rotors. The space reduction increases the gas pressure and pushes the gas towards the outlet. Figure 5.3 shows an example of the compressor screws.

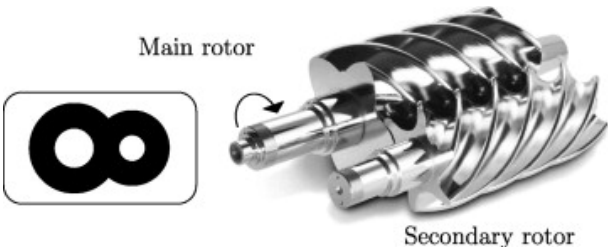


Figure 5.3: Twin screws from a screw compressor (Krichel and Sawodny, 2011).

These compressors can use oil injected to the screws. In this case the oil serves as lubricant and cooling agent. Oil-injected screw compressors have higher efficiencies and cooling allows higher compression ratios in one stage. Usual values for the adiabatic efficiency of these units are between 70 % and 80 % (Bahadori and Bahadori, 2014).

5.3.1 Model

The model of the compressor uses an expanded version of Algorithm 4.2. However, this case does not assume constant efficiency. The efficiency is calculated by using a curve based on the pressure ratio. Steps 2, 3, 7 and 8 use the equation of state

Algorithm 5.1: Compressor calculation

1 **begin**
2 Define input parameters: $z, P_{in}, P_{out}, T_{in}$
3 Calculate input h_{in}, s_{in}
4 Calculate T_{out}, h_{out} isoentropic output $s_{out} = s_{in}$
5 Calculate $eta = f(P_{out}/P_{in})$
6 Calculate $W_{c,i} = \dfrac{h_{out} - h_{in}}{\eta}$
7 Calculate $h_{out,real} = h_{in} + W_c$
8 Calculate $T_{out,real}$ from $h_{out,real}$
9 **end**

(SRK) to find the values of the desired state (h, T or s). Step 5 uses an interpolated curve to calculate the value of the efficiency. The regression is shown on equation (5.2).

Curves

The curves for screw compressors are different from the usual ones used for centrifugal compressors. Figure 5.4 shows the compressor curve used for this model.

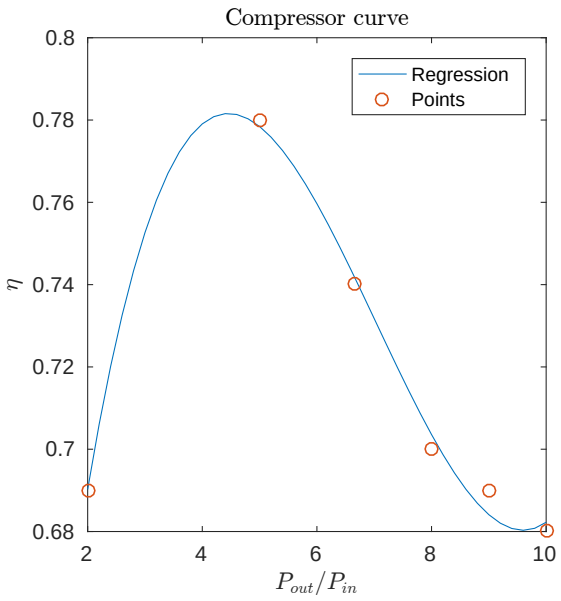


Figure 5.4: Compressor curve.

There is no information available about the actual compressor curve for the unit used in this process. Points for the curve were assumed based on the curves shown

by Eikevik (2015), Brendeng (1979) and Sauls and Sauls (1982).

This curve is only for a fixed built in volume ratio. This ratio is not specified.

$$0.0015\pi^3 - 0.0314\pi^2 + 0.1911\pi + 0.4217 \tag{5.2}$$

where $\pi = P_{out}/P_{in}$

5.4 Plate Heat Exchangers

All the heat exchangers in the Mini-LNG plant are plate heat exchangers. These units use metal plates to transfer heat between two fluids. The configuration allows the fluids to exchanger heat in a larger area than using shell and tube exchangers. The plates can be either held together by using gaskets or brazing the plates jointly. The alternative which uses gaskets enables easier maintenance, as some individual plates can be replaced. Additionally, the area of some exchangers can be modified by manipulating the amount of plates in the stack. The operation of a plate heat exchanger unit can be seen on Figure 5.5.

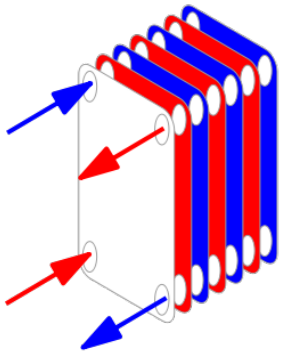


Figure 5.5: Plate heat exchanger schematic. [1]

These units can be used for small size processes, as well as for condensation or evaporation of fluids.

Plate heat exchangers can operate with smaller temperature approaches (down to $1°C$) compared to shell and tube exchangers (down to $5°C$).

The biggest drawbacks to these units are the price and the high pressure drops. However, due to their reduced size they are well suited for plants where space is a constraint.

[1] https://commons.wikimedia.org/w/index.php?curid=15033819

5.4.1 Model

The model of the heat exchangers is performed in two parts.

- *Energy balance.* The first step is to calculate the charge of a fully defined side. From this side the heat is used to calculate the remaining side. Once there is a fully defined heat exchanger (both sides meet the energy balance), the second step can be carried out.

- *Area calculations.* This is an optimization that is for operation purposes, therefore the areas of the heat exchangers have to remain constant throughout the successive optimization runs. In order to do so, the area is calculated for the base case using equation (5.3).

$$A = \frac{Q}{U\ LMTD} \tag{5.3}$$

  where

  Q is the heat duty

  U is the overall heat transfer coefficient

  A is the area

  $LMTD$ is the logarithm average of temperature differences

  The use of equation (5.3) is a very practical approach. This equation is built on the assumption of ideal counter current flow and constant heat capacities for the fluids (Skogestad, 2009). In reality, these assumptions do not hold for this refrigeration system. For a plate heat exchanger the flows are not ideally counter current and the fluids do not have constant heat capacity as they are changing phase. However, a detailed model of the heat exchangers is outside of the scope of this work. The use of equation (5.3), allows us to make a preliminary model of the process and set up the operational constraints for the heat exchangers.

  In order to calculate the area, an additional assumption on equation (5.3) has been made. As U cannot be known, it is assumed to be constant. In reality, the value of U changes as the flow of the streams change as well as their phase distribution inside the heat exchanger.

  Given that A is constant and U is assumed to be constant, then equation (5.3) becomes equation (5.4).

$$U\ A = \frac{Q}{LMTD} \tag{5.4}$$

  This allows us to easily set up the operational constraints for a unit i by using the product $U_i\ A_i$ as shown on Section 6.2.

5.5 Plant Model

The plant model is set up in an equations oriented way. That means that all model equations are solved simultaneously. The system is formulated according to Table 5.7.

Table 5.7: Summary of calculations in the plant model.

Stream	Calculation type
1	Compressor model
3	PH flash
6	TP flash
7	PH flash
11	PH flash
13	PH flash
14	PH flash
15	PH flash
16	PH flash (Slack)
17	PH flash (Slack)

On Table 5.7 the compressor model corresponds to the equations from Algorithm 5.1 and the flash calculations are the equations from Algorithm 4.1. It should be noted that the only flash calculations that use slacking are for streams 16 and 17 as they are both close to the boundary between one and two phases.

The initial estimate comes from the modular sequential model proposed by Leguizamon (2015).

The dimensioning of the heat exchangers is modelled as an operational constraint. This is further explained on Section 6.2.

The implemented code for the model is shown on Appendix B.

CHAPTER

SIX

OPTIMIZATION PROBLEM

This chapter focuses on the definition of the optimization problem. In order to solve it, it is necessary to carry out the following steps:

- Define the objective function.
- Define the variables for optimization. In this case, this definition is done by carrying out the degrees of freedom analysis (Section 5.2).
- Define the constraints.
  - *Equality constraints:* These correspond to the model and any other set parameter. The detailed explanation for the definition is shown on chapter 5.
  - *Inequality constraints:* These are operational constraints. It is very important for these constraints to be well defined as they will be used to select the main controlled variables.

This chapter will cover all of the mentioned decisions except for the ones related to the model as they are explained in detail on chapter 5.

6.1 Objective Function

Natural gas processes can be optimized in very different ways depending whether it is design or operation, and within operation there are plenty of alternatives to define the objective function. Some examples for design optimization include minimize capital, operating or annualized cost, or to maximize the profit. For operation, some usual objectives are to minimize the utility consumption or to maximize the exergy efficiency. Additionally, the optimization can be formulated for different goals at the same time (Austbø et al., 2014).

As the LNG plant is already designed, the optimization on this work will be for the operation.

For optimal operation of a process, the objective function is defined as an scalar function, which quantifies the operational objectives in terms of currency per unit time ($/s). Skogestad (2012) defines the general objective function to minimize as:

$$J = cost\ feed + cost\ utilities(energy) - value\ products \tag{6.1}$$

where all quantities are given in $/s.

Equation (6.1) shows a very general operation objective function. Different definitions for the objective function can be used (Austbø et al., 2014). However, several studies have shown that minimizing the compressor work (equation(6.2)) is the most consistent to provide minimum operation values compared to other possible objective functions (Hatcher et al., 2012; Wang et al., 2013).

$$J = W_c \tag{6.2}$$

where W_c corresponds to the compressor work.

The objective function in this problem is defined by equation 6.2.

6.2 Operational Constraints

The operational constraints defined for this problem are the following:

- *LNG specification.* This is the most important constraint of all, as it sets the output to be actual LNG. According to Neksået al. (2010), the output temperature should be $-154°C$.

$$T_{out,BOG} = -154°C \tag{6.3}$$

- *Pressure ratio.* Screw compressors operate in a range given by the pressure ratio. In this case, the pressure ratio is defined between 2 and 10.

$$2 \ \leq \ \frac{P_{out,c}}{P_{in,c}} \ \leq 10 \tag{6.4}$$

- *High pressure bound.* The high pressure (output from the compressor) is defined by an operational constraint from the heat exchangers. The maximum pressure the heat exchangers can handle is 25 *bar* (Neksået al., 2010). Therefore, the output pressure from the compressor cannot be higher than 25 *bar*.

$$P_{out,c} \ \leq 25\ bar \tag{6.5}$$

- *Low pressure bound.* For the low pressure in the system (valves output pressure and compressor input pressure), the value has been defined as at least higher than 1.5 *bar*. This has been set to be slightly higher than the atmospheric pressure.

$$P_{low} \ \geq \ 1.5\ bar \tag{6.6}$$

- *Split flow boundaries.* Stream 8 is split into streams 9 and 10. The split ration should be set in such a way that ensures a flow larger than zero for both streams 9 and 10.

$$0 \leq \frac{n_9}{n_9 + n_{10}} \leq 1 \tag{6.7}$$

- *Stream 3 temperature.* This temperature is controlled by the charge of unit HA-01. According to Neksået al. (2010), the output from this unit can be as cold as $-40°C$. The input to this unit is at $35°C$. For stream 3 then the temperature can vary only between $-40°C$ and $35°C$.

$$-40°C \leq T_3 \leq 35°C \tag{6.8}$$

- *Heat exchanger sizes.* The areas for the heat exchangers must remain constant.

$$A_{i,new} = A_{i,base} \tag{6.9}$$

  where $A_{i,new}$ is the recalculated area for a unit i and $A_{i,base}$ is the calculated area for a unit i using the conditions from the base case.
  Using the base case, a value for UA (heat transfer coefficient times area) can be calculated for the units HA-02, HA-04 and HA-07. Assuming that the overall heat transfer coefficient remains constant, equation (6.9) gives the following constraints:

$$UA_{HA-02,new} = UA_{HA-02,base} \tag{6.10a}$$
$$UA_{HA-04,new} = UA_{HA-04,base} \tag{6.10b}$$
$$UA_{HA-07,new} = UA_{HA-07,base} \tag{6.10c}$$

  The precooler and unit HA-01 are not included as there is not enough information from the base case to calculate these units.

- *Heat exchanger crosses.* To ensure feasibility and to increase the efficiency of the heat exchangers, several constraints are defined.

  HA-02:

$$T_5 \geq T_{16} \tag{6.11a}$$
$$T_6 \geq T_{15} \tag{6.11b}$$

  HA-04:

$$T_6 \geq T_{13} \tag{6.11c}$$
$$T_8 \geq T_{12} \tag{6.11d}$$

  HA-07:

$$T_{BOG,in} \geq T_{14} \tag{6.11e}$$
$$T_{BOG,out} \geq T_{11} \tag{6.11f}$$

  These constraints are included in the problem as a penalty term. More details are included on Section 6.3.

- *Refrigerant flow.* This is a closed system, therefore the degree of freedom that exists is the active charge. The amount of refrigerant which remains inside all the units is very difficult to measure. There are several reasons for this. The first is that the dimensions of the units are unknown, additionally there is a phase change occurring which increases the complexity of the estimation.

  The refrigerant flow is then used without direct boundaries. However, this flow is indirectly bounded by the other conditions. The amount of refrigerant should be the minimum that meets the energy needs of the system (the heat to condense the BOG) while being sufficient for feasible operation.

6.3 Implementation

This section covers the most relevant remarks about the implementation of the optimization problem. Further details can be checked on Appendix C

6.3.1 Objective function

To solve the problem, the objective function needs to include additional terms.

- *Flash calculations.* The flash calculations for streams 16 and 17 require the use of penalty terms to account for the complementarity constraints.

$$M_1 s_{16} v_{f,16}^T + M_1 s_{17} v_{f,17}^T \tag{6.12}$$

  where $M_1 = 10^6$.

- *Temperature crosses.* As stated earlier, in order to increase the efficiency, the temperature crosses are included as a penalty term.

  HA-02:

$$Cross_1 = T_5 - T_{16} \tag{6.13a}$$
$$Cross_2 = T_6 - T_{15} \tag{6.13b}$$

  HA-04:

$$Cross_3 = T_6 - T_{13} \tag{6.13c}$$
$$Cross_4 = T_8 - T_{12} \tag{6.13d}$$

  HA-07:

$$Cross_5 = T_{BOG,in} - T_{14} \tag{6.13e}$$
$$Cross_6 = T_{BOG,out} - T_{11} \tag{6.13f}$$

  The penalty term for the crosses is then

$$M_2 * P_{Cross} \tag{6.14}$$

where $M_2 = 10^3$ and

$$P_{Cross} = \sum_{i=1}^{6} |Cross_i| \tag{6.15}$$

Finally the objective function uses a scaled term for the work using a scaling factor $W_0 = 250$ $[kJ/s]$. The implemented objective function is shown on equation (6.16).

$$J = \frac{W_c}{W_0} + M_1 s_{16} v_{f,16}^T + M_1 s_{17} v_{f,17}^T + M_2 P_{Cross} \tag{6.16}$$

6.3.2 Perturbations

The amount of perturbation points is set based on a trade-off between time consumption of the overall run and closeness among points. Keeping this in mind the ranges shown on Table 6.1 were defined.

Table 6.1: Perturbation points

	Lower bound	Upper bound	Step Size	Number of points
$+T_{BOG,in}[K]$	0	10	5	3
$m_{BOG}(\%)$	90	110	10	3
$Q_{Env}(\%)$	90	110	5	5

The perturbation points are set up in terms of the base case.

- $+T_{BOG,in}[K]$ is the amount of additional degrees of the incoming BOG. Where $+T_{BOG,in}[K] = 0$ is the base case.
- $m_{BOG}(\%)$ is the percentage of the incoming BOG flow, where $m_{BOG}(\%) = 100$ means that the flow is the same as in the base case.
- $Q_{Env}(\%)$ is the percentage of the heat from the environment, where $Q_{Env}(\%) = 100$ means that the heat from the environment is the same as in the base case.

This distribution generates a total of 45 points. The first two perturbations (Inlet temperature and flow of BOG) are sorted based on the required charge to condense the BOG given the perturbations. Then for each point of that sorted space, the perturbations of the environment are included to solve the problem.

6.3.3 Initialization

The initial point for the whole optimization comes from the modular sequential simulation made by Leguizamon (2015). The base case is optimized first and its 15th iteration is used as 'warm' initialization for the sequential runs. Each run uses the 15th iteration from the previous run as initial point.

CHAPTER

SEVEN

RESULTS

7.1 Objective function

The surface obtained for the objective function given the perturbations is shown in Figure 7.1.

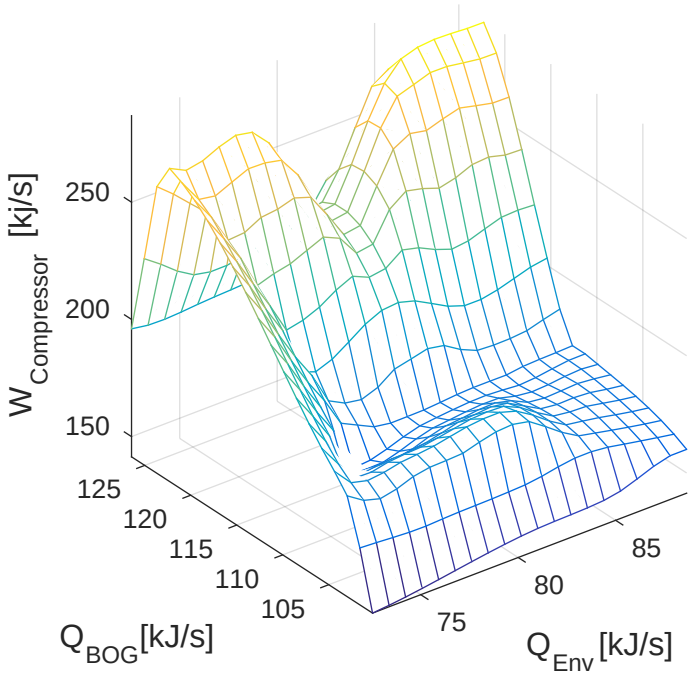


Figure 7.1: Optimal compressor charge for different perturbations.

Figure 7.1 is shown in terms of the heat of the environment Q_{Env} and the energy needed to condense all the BOG Q_{BOG}. Q_{BOG} is the duty to the unit HA-07.

Q_{BOG} is defined in terms of the incoming flow and temperature. This correlation is shown on Figure 7.2

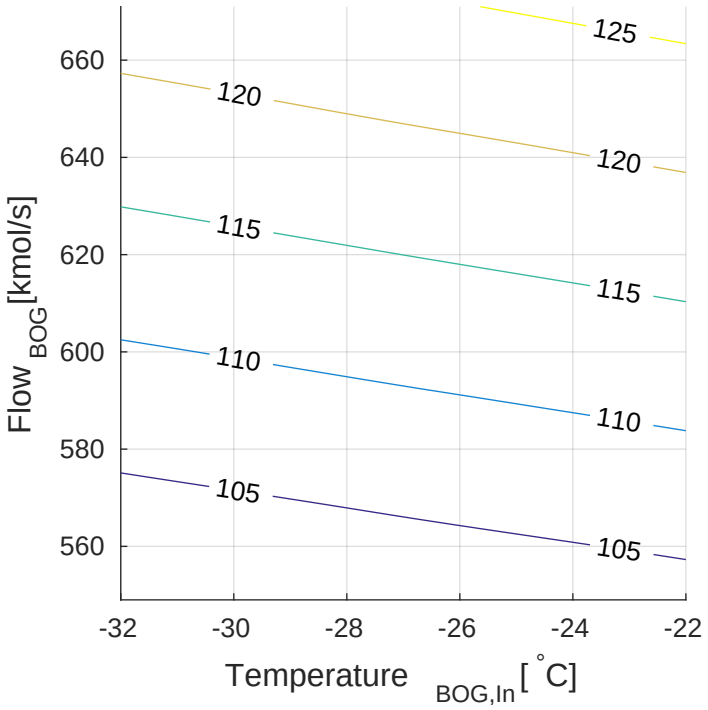


Figure 7.2: Q_{BOG} in terms of the mass and inlet temperature of BOG.

7.2 Active constraints regions

The active constraints regions obtained for this problem are shown in Figure 7.3.

There were no active constraints for all the perturbations in the system. Failed states are those in which the solver did not converge to a feasible point.

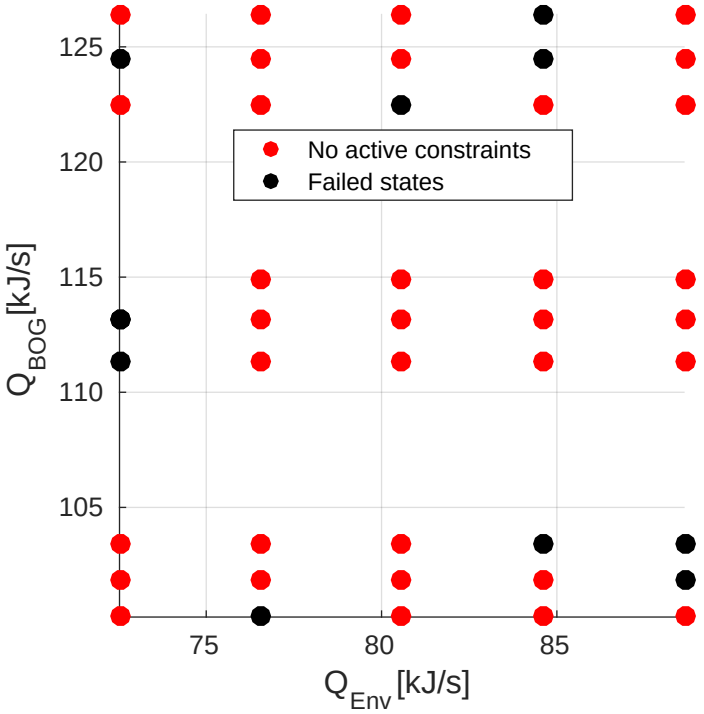


Figure 7.3: Determined active constraints regions.

7.3 Manipulated variables

As there is only one active constraint region, it is important to consider the behaviour of the manipulated variables.

7.3.1 Pressures and efficiency

The pressures are very important parameters when calculating the compressor charge. The relationship between the input and output pressures sets up the compressor efficiency. They are also very important parameters in order to calculate the enthalpies at the inlet and outlet of the compressor.

The behaviours are displayed in the following order: Low pressure or compressor inlet (Figure 7.4), high pressure or compressor output (Figure 7.5) and compressor efficiency (Figure 7.6).

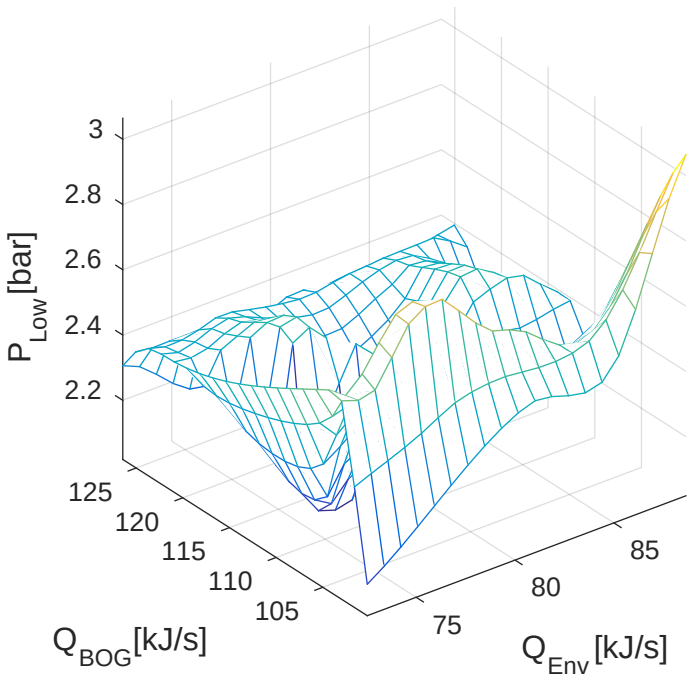


Figure 7.4: Low pressure for different perturbations.

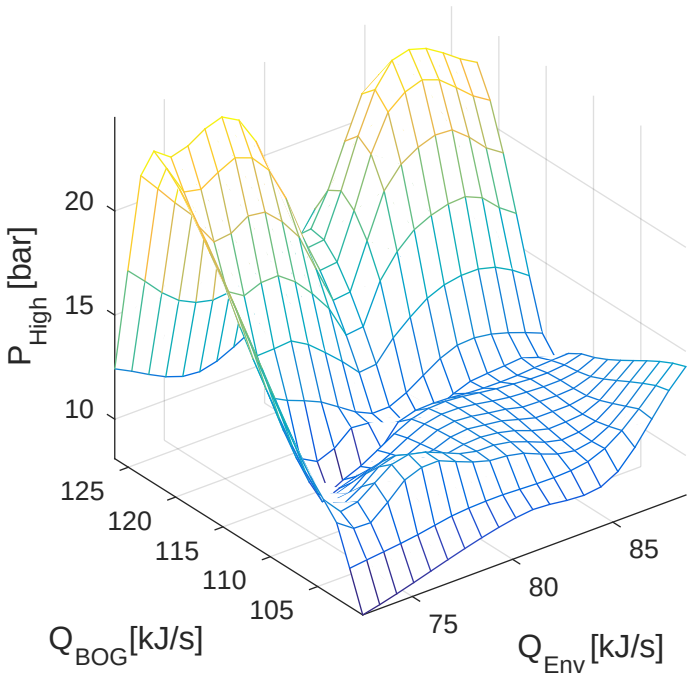


Figure 7.5: High pressure for different perturbations.

7.3.2 Refrigerant flow

The behaviour for the refrigerant flow is registered on Figure 7.7. The flow shown on Figure 7.7 is dimensionless. This means that the quantity displayed is the ratio between the actual flow and the base case. This is shown in equation (7.1)

$$Flow = \frac{F^{p}_{Ref}}{F^{bc}_{Ref}} \tag{7.1}$$

where $Flow$ is the dimensionless refrigerant flow, F^{p}_{Ref} is the refrigerant flow when a perturbation is present and F^{bc}_{Ref} is the refrigerant flow for the base case.

7.3.3 Temperature and split flow

The last two manipulated variables correspond to the duty of unit HA-01 and the split flow from stream 8 into 9 and 10.

The heat removed on unit HA-01 has a direct effect on the temperature of stream 3. The temperature and pressure of stream 3 set up the phase equilibria for the separator. Figure 7.8 shows the results obtained for the temperature of stream 3.

The values for the split flow vary from 0% to 100%, where 100% means that all the flow from stream 8 is sent to stream 10. Figure 7.9 shows the split flow values obtained.

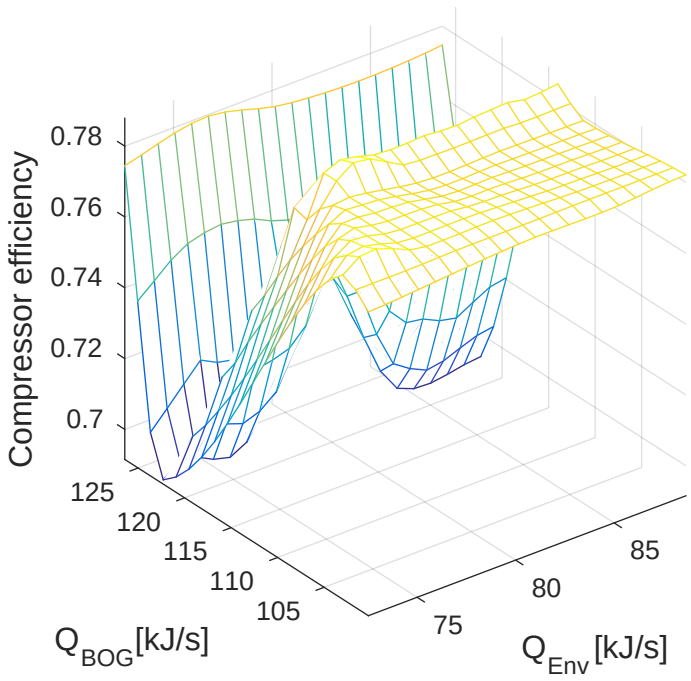


Figure 7.6: Compressor efficiency for different perturbations.

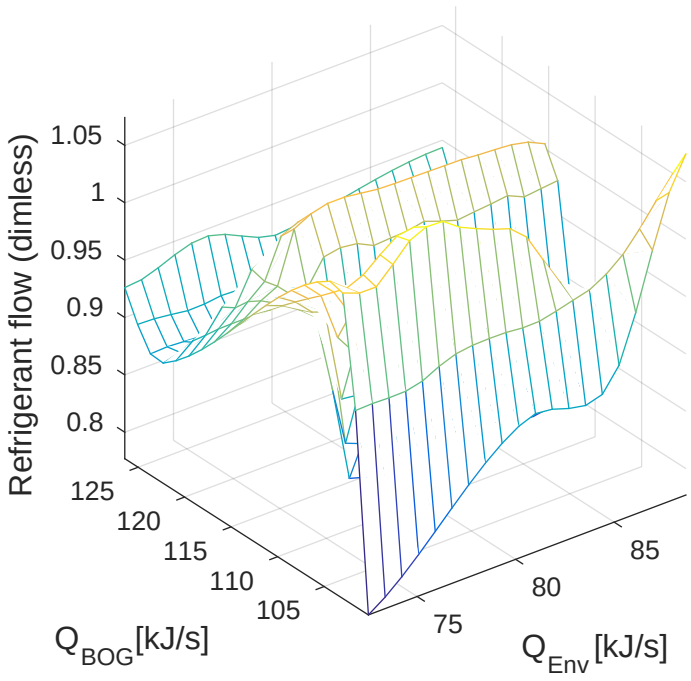


Figure 7.7: Refrigerant flow (dimensionless) for different perturbations.

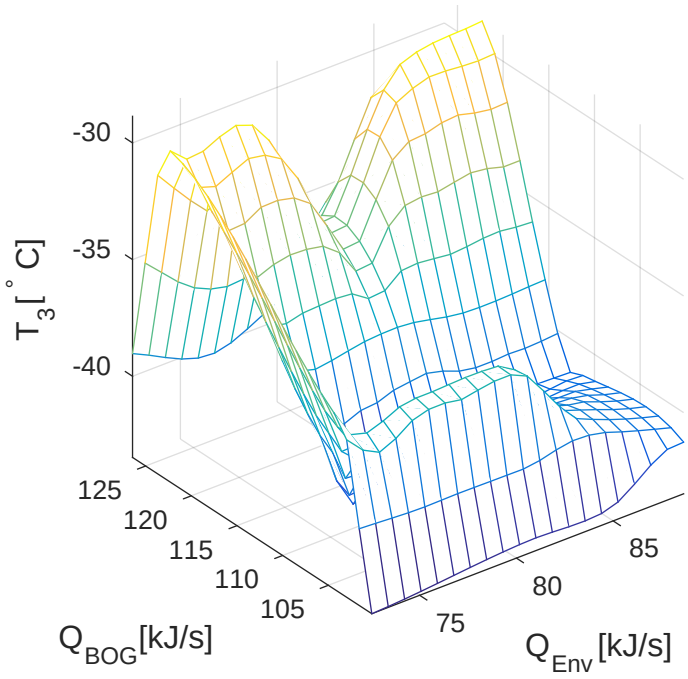


Figure 7.8: Temperature of stream 3 for different perturbations.

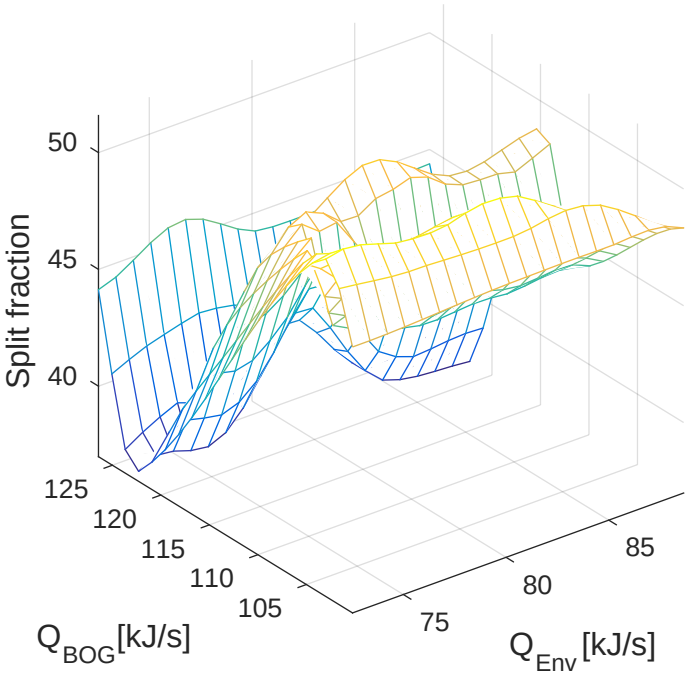


Figure 7.9: Flow split for different perturbations.

CHAPTER

EIGHT

DISCUSSION

8.1 Plant operation

8.1.1 Compressor charge

The expected behaviour of the process predicts a smaller compressor charge W_c when the environment is cold (Q_{Env} is low) and the amount of energy to remove is low (Q_{BOG} is low).

Figure 7.1 shows two important trends: first, the compressor charge increases for larger amounts of heat to remove and second, the influence of Q_{Env} is considerably lower than the effect of Q_{BOG}. These trends correspond to what is anticipated for refrigeration cycles, even though the effect of the environment is not significant for all points.

The compressor charge is the function of three parameters: the input pressure, output pressure and refrigerant flow. Considering only the compressor, optimal operation requires the lowest possible refrigerant flow and output pressure, while setting up the inlet pressure so that maximum efficiency is assured.

The following behaviours can be expected for W_c, if the interactions with the rest of the process are neglected:

- *Refrigerant flow.* It should be as low as possible while increasing the cooling demand.
- *Efficiency.* This value should be kept constant at the maximum point for all the cases, unless one of the pressures hits a boundary.
- *Pressures.* Both, input and output pressure for the compressor should keep the relationship for maximum efficiency while respecting their boundaries.

On Figures 7.4, 7.5, 7.6 and 7.7, it can be seen that the pressures, efficiency and flow follow the expected trend for values of Q_{BOG} lesser than 115 $kj/kmol$. For values

larger than 115 and smaller than 125 $kj/kmol$ the trend is different. This difference can be understood by the internal interactions in the process.

8.1.2 Internal interactions

To understand the internal interactions let us consider two control volumes as shown in Figure 8.1

Not all the variables have a direct effect on the compressor charge. The remaining T_3 and split flow have an indirect effect as they ensure feasibility of the operation. All the operational constraints should be met inside V-1, that is the heat exchanger sizes should be met without temperature crosses, while keeping the product on specification.

Cold pocket temperatures

There are two temperatures inside V-1 that are set; they are T_6 and T_8.

T_8 is set so that stream 8 is always liquid. 8 has to be liquid for the valves FCV-04 and FCV-05 to work. This configuration allowed to model the system without an additional slacked flash calculation.

To calculate units HA-2 and HA-04 the conditions of at least 3 of their streams should be known, as their area is unknown.

For HA-04, the conditions of stream 8 and value of the low pressure set up the state of stream 12. This gives only one side for HA-04.

For HA-02, only stream 5 is known. In this case fixing stream 6 allows to calculate HA-02 duty from 5 to 6 side and the duty of HA-04 (from the 6 to 8 side) and, given that 12 is known, also stream 13. In addition, streams 7 and 14 are also known so that the state of stream 15 can be calculated. Knowing the state of streams 15, 6 and 5 the whole unit HA-02 can be calculated.

Separator effect

The conditions inside the separator (P_3 and T_3) set up the flow and composition for the streams that enter V-2. There are two possible effects, while satisfying the same product specifications (Q_{BOG} is constant):

- *P_3 increases while T_3 is constant.* This means the pressure inside the separator is higher. Thus, the flow of 5 and 8 decreases. Given that T_6 and T_8 are constant, a larger portion of stream 8 has to be directed to 11 (split flow decreases).
- *T_3 increases while P_3 is constant.* This increment produces more vapour, which means the flow of stream 6 and 8 increases. Larger flows in stream 8 increases flow sent to stream 10 (higher split fraction).

Figure 7.5 shows that the pressure increases but at the same time, Figure 7.8 shows an increment in T_3. This means there is a combination of effects in order to satisfy the operational constraints.

Figure 7.9 displays the distribution of the flow. In this plot it can be seen that there is a reduction in the fraction or an increase in the flow of stream 9. Based on the previous discussion, the effect of P_3 was larger than that of T_3.

Low pressure

P_7 or the low pressure of the system has to be set considering the following guidelines:

- It should be larger than its operational limit (1.5 *bar*).
- It has to be set in relation to the high pressure so efficiency is maximized.
- It has be low enough to allow stream 11 to reach a cold enough temperature to satisfy the LNG specifications.

Based on these guidelines, the trend of this pressure is more difficult to predict. However, it can be expected that the variance remains in a restricted range, as it is subject to more constraints than the higher pressure. Figure 7.4 shows a more restricted variation for the low pressure, of around 1 *bar*, compared to 10 *bar* for the high pressure. Yet, Figure 7.4 is a very uneven surface which does not show a clear trend.

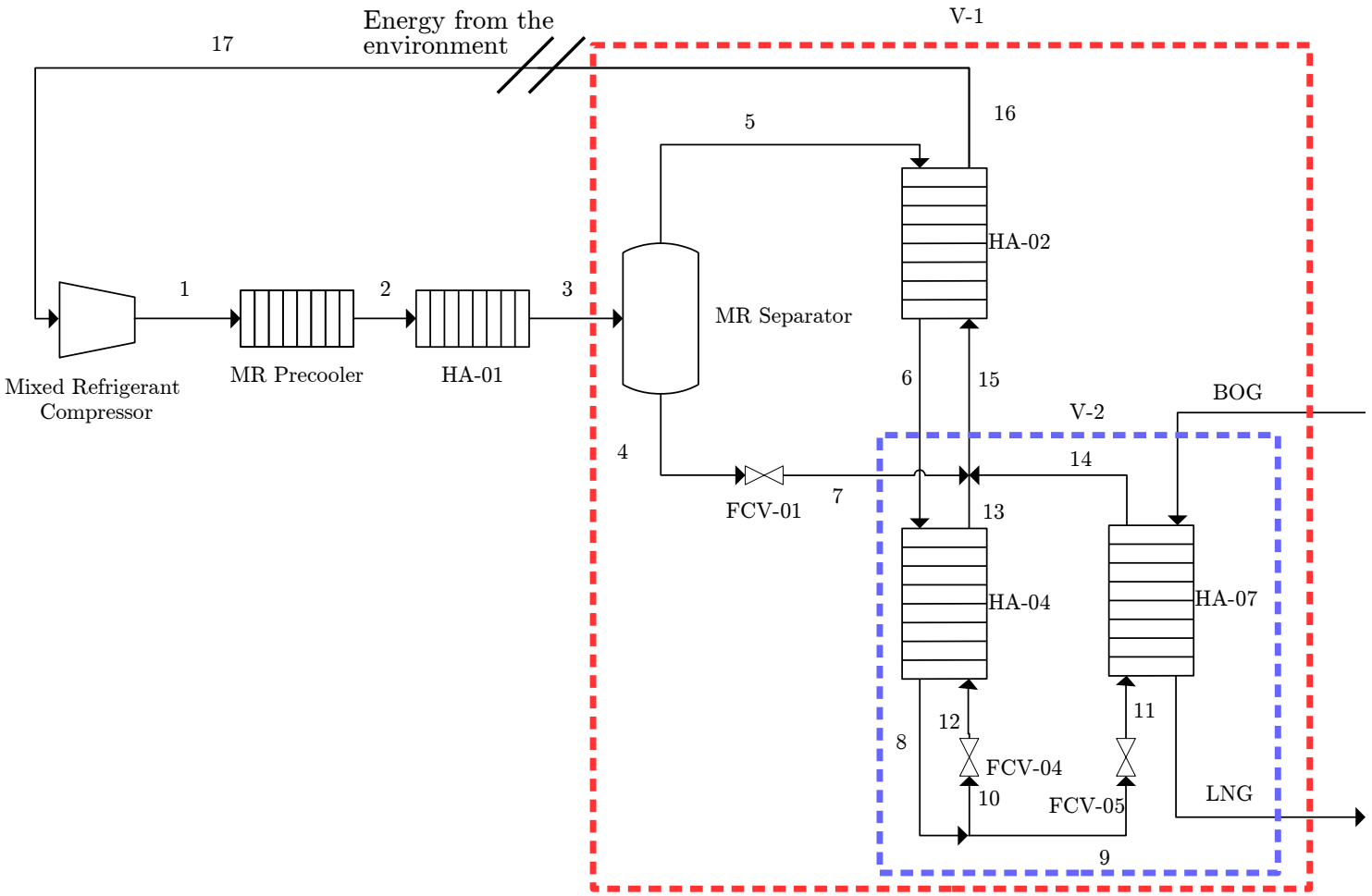


Figure 8.1: Control volumes for analysis.

Flow effect

The total refrigerant flow has an modifying effect on the system response to changes in P_3, T_3 and on the compressor charge. A larger flow can have different effects on the internal interactions of the process depending on the vapour fraction inside the compressor. If said fraction is higher than 0.5, the refrigerant flow will magnify T_3 and P_3 effect on streams 5 and 6. If the opposite case is true, then the effect of the total refrigerant flow will lower the impact of changes in T_3 and P_3.

Considering the results from Figures 7.5, 7.8 and 7.9, it is possible to notice that the effect of pressure on the split fraction was higher than that of temperature. This was increased by the flow as the vapour fraction inside the separator is higher than 0.5.

Additionally an increment in the flow makes Q_{Env} less representative. A larger flow of stream 16 requires more heat from the environment to have a significant effect on stream 17. However, this is not a decisive factor when modifying the flow.

It should be expected to have a flow that varies with the amount of heat to be removed by the system. On Figure 7.7, the flow does not follow this trend. The reason for this is the flow is adjusted so the effect of other interactions is diminished. However, it can be expected to be lower than the base case for most of the scenarios. As a lower flow reduces the amount of energy needed in the compressor. This corresponds to the results shown in Figure 7.7.

Another interesting effect is the relationship between the flow and high pressure. It is expected for high pressure points to have low flow to achieve an optimal compressor operation. Figures 7.5 and 7.7 do not show a clear trend. The main reason can be the combination of other effects (such as T_3) on the final result.

8.2 On the optimization problem

The discussion on the optimization problem will be in two parts. The first one involves the proposed model and its assumptions. The second one corresponds to the algorithm and numerical approach used for this problem.

8.2.1 Model

Previous work

The model formulation is a continuation of the specialization project done by Leguizamon (2015). However, there were some drawbacks for this project that needed to be addressed.

The first weak point was the thermodynamics formulation. The previous formulation did not have a proper behaviour on the points that are close to the two phase boundary (streams 16 and 17). This work dealt with this issue by implementing the relaxation procedure proposed by Kamath et al. (2010). The new formulation

granted enough flexibility to provide an optimal point for most of the perturbations.

An additional improvement on the previous formulation was the inclusion of the active charge as a DOF. Despite the impossibility to have an estimate for the active charge, it was possible to indirectly consider it throughout the optimization.

The final improvement corresponds to the model's convergence. On previous work the model did not converge. However, implementing an EO approach in the optimization problem allowed solving the model while optimizing it, without using tear streams.

Thermodynamics

This is the basis for the model, as thermodynamics describe the properties of the streams in the process. Implementing a reliable thermodynamic model was a priority in this work. The use of Kamath et al. (2010) demonstrated sufficient reliability for this process, as explained on Chapter 4.

The chosen approach involved the use of complementarity constraints. These constraints are far from linear and require special care when using an NLP algorithm.

The inclusion of complementarity constraints in the objective function proved to be sufficiently reliable. In a similar manner, the implementation can be expanded to model the liquid phase change.

Model detail

The models for the heat exchangers were based on a very pragmatic approach. However, the chosen approach is not closely representative of reality. They work on the assumption of ideal counter current and constant heat capacity.

In reality, the heat exchangers do not have ideal counter current flow and the fluids inside do not have constant heat capacity as there is a phase change. Therefore, the used assumptions are not the most suitable ones for this plant.

A solution to this is the implementation of a more detailed model of the heat exchangers. This model should take into consideration the phase change and non ideal counter current flow. However, this work can be used as an starting point for a plant model involving more detailed heat exchanger calculations.

An additional improvement that can be done is the modelling of the active charge. This can be done by including dimensions for the units and using more detailed unit models. The dimensions will set up the maximum capacities and the detailed models enable a more accurate estimation of the fluid's specific volume inside the units.

Having values for maximum capacities and volumes, it is possible to estimate the active charge. In this way the optimization problem can be refined by including the

operational constraints related to the active charge, instead of modelling it as an unbounded flow.

Having areas for the heat exchangers can be a great advantage. A more detailed model can be formulated using these areas. This will allow more realistic results.

A new model can let T_6 and T_8 calculated using the known area of the exchangers. Additionally for T_8, it is necessary to expand the thermodynamic calculations to also include the change from liquid only to the two phase region.

8.2.2 Algorithm

As it was explained in the discussion about optimal operation, this is a process which has several internal interactions. Those effects plus the thermodynamics concede a great degree of complexity and non linearity for the optimization problem. Despite these difficulties, it was possible to obtain results using the interior point method.

This is a local method and given the non convexity of the problem, it is possible that some of the results obtained correspond to local minima. An improvement for this work can be the use of a global optimization algorithm.

Other than using another algorithm, another possibility is to use different starting points for the optimization. For this case, the 'warm' initialization began using the base case, but it can begin with other points. Considering the complexity of the problem, using different starting points for the region of $115 \leq Q_{BOG} \leq 120$ can be an alternative to validate the behaviour in said area. In which given the amount of internal interactions it is very difficult to propose an accurate theoretical prediction

8.3 Active constraints regions

With this region (Figure 7.3) the next step in the plantwide control procedure is to use the optimal results to determine the optimal combination of measurements.

There is one area of active constraints. But this only involves the directly specified constraints. As it was discussed in Section 8.1, several operational constraints are set up in a dynamic way depending on the other constraints.

Due to the complexity of the model, there are some boundaries and behaviours that were not considered as explained in Section 8.2.1. Generating a more detailed model will allow to set up additional bounds from which it is possible to obtain more than one active constraint region.

Failed states

There were 10 failed states out of 45. Their occurrence is not limited to a specific area. The following are examples of failed states:

- They can be sub optimal points, as the algorithm can be stuck in a feasible point which is not optimal. This can happen if the steps are too large or there was a poor initialization.
- A failed state can happen when the algorithm takes too many steps without finding an optimal point. This can happen if search direction shifted from one local minimum to another, due to poor initialization.

A way to fix the poor initialization is to tune the 'warm' start parameter. It is the number of iteration used as initial point for the next perturbed problem. By choosing a large number, there could be a snowball effect as each successive problem can bring that initial point close to the barrier. This makes points more likely to be a poor initial point. On the other hand, choosing an early iteration the snowball effect can be reduced. However, for earlier points the gain in speed is reduced. Speed is a non trivial concern as there are many successive problems to solve.

The possibility of increasing the resolution of the grid by calculating more points should be considered. This can lead to see in more detail the effects of the internal interactions on the optimal points. However, for larger grids, speed becomes a more relevant concern. Thus, a careful study on the initialization is recommended.

CHAPTER

NINE

CLOSING REMARKS

9.1 Conclusions

This thesis covered the modelling and optimization of the mini-LNG plant. It was possible to provide a sufficiently reliable model. This is based on an accurate thermodynamic implementation. The model was optimized for a large space of disturbances.

The thermodynamic model implemented in this work can easily used for other natural gas applications. Additionally, the implementation can be modified to include a larger set of components.

The main result from this work is one region of active constraints. This is subject to the modelling assumptions. The behaviour of the system is heavily influenced by the internal interactions within the cold pocket.

In the context of plantwide control, the next step is to calculate the best combination of measured variables. This is subject to changes if an improved model predicts a different behaviour.

The interior-point method proved to be solve most of the disturbed optimization problems. Given the size of the feasible region and its non convexity, its results need to be validated using a global optimization method.

In summary, this model and optimization needs to be further validated and expanded. However, the obtained model and results set up a basis for further improvements in the model.

9.2 Further work

The model needs to count with more detailed calculations for the heat exchangers. These will give a more realistic results.

The use of more detailed models and sizes can set up realistic constraints for the active charge. This can lead to more than one active constraints regions.

The use of a global optimization algorithm is strongly recommended. The non convexity of this problem make it difficult to find global optimums without proper initialization.

All further improvements in detail should be handled with special care when optimizing, given all the internal interactions already present.

BIBLIOGRAPHY

Vidar Alstad, Sigurd Skogestad, and Eduardo S. Hori. Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19(1):138–148, 2009. ISSN 09591524. doi: 10.1016/j.jprocont.2008.01.002. URL http://dx.doi.org/10.1016/j.jprocont.2008.01.002.

E. M B Aske and Sigurd Skogestad. *Dynamic degrees of freedom for tighter bottleneck control*, volume 27. Elsevier Inc., 2009. doi: 10.1016/S1570-7946(09)70603-0. URL http://dx.doi.org/10.1016/S1570-7946(09)70603-0.

Bjørn Austbø, Sigurd Weidemann Løvseth, and Truls Gundersen. Annotated bibliography-Use of optimization in LNG process design and operation. *Computers and Chemical Engineering*, 71:391–414, 2014. ISSN 00981354. doi: 10.1016/j.compchemeng.2014.09.010. URL http://dx.doi.org/10.1016/j.compchemeng.2014.09.010.

Alireza Bahadori and Alireza Bahadori. Chapter 5 – Gas Compressors. In *Natural Gas Processing*, pages 223–273. 2014. ISBN 9780080999715. doi: 10.1016/B978-0-08-099971-5.00005-2.

Hande Y Benson, David F Shanno, and Robert J Vanderbei. Interior-point methods for nonconvex nonlinear programming. *Technical Report*, 2(2):1–20, 2002.

B.P. BP Energy Outlook 2035. Technical Report February, 2015.

E. Brendeng. Reciprocating compressors or screw compressors? *International Journal of Refrigeration*, 2(3):163–170, 1979. ISSN 01407007. doi: 10.1016/0140-7007(79)90060-4.

John P. O'Connell Bruce, E Poling, John M. Prausnitz. *Properties of Gases and Liquids, Fifth Edition - Access Engineering from McGraw-Hill*. 1987. URL http://accessengineeringlibrary.com/browse/properties-of-gases-and-liquids-fifth-edition.

A.C B De Araújo, Eduardo S. Hori, and Sigurd Skogestad. Application of plantwide control to the HDA process. II-regulatory control. *Industrial and Engineering Chemistry Research*, 46(15):5159–5174, 2007. ISSN 08885885. doi: 10.1021/ie061393z.

R. Dohrn and O. Pfohl. Thermophysical properties-Industrial directions. *Fluid Phase Equilibria*, 194-197:15–29, 2002. ISSN 03783812. doi: 10.1016/S0378-3812(01)00791-9.

Trygve M Eikevik. Positive Displacement Compressors, 2015.

Anders Forsgren, Philip E. Gill, and Margaret H. Wright. Interior Methods for Nonlinear Optimization. *SIAM Review*, 44(4):525–597, 2002. ISSN 0036-1445. doi: 10.1137/S0036144502414942.

Rafiqul Gani, Nuria Muro-Suñé, Mauricio Sales-Cruz, Claude Leibovici, and John P. O'Connell. Mathematical and numerical analysis of classes of property models. *Fluid Phase Equilibria*, 250(1-2):1–32, dec 2006. ISSN 03783812. doi: 10.1016/j.fluid.2006.09.010. URL `http://www.sciencedirect.com/science/article/pii/S0378381206004109`.

Mevludin Glavic and Louis Wehenkel. Interior Point Methods : A Survey , Short Survey of Applications to Power Systems , and Research Opportunities. Technical report, University of Liege, 2004.

Vipin Gopal and Lorenz T. Biegler. Smoothing methods for complementarity problems in process engineering. *AIChE Journal*, 45(7):1535–1547, jul 1999. ISSN 00011541. doi: 10.1002/aic.690450715. URL `http://doi.wiley.com/10.1002/aic.690450715`.

Prue Hatcher, Rajab Khalilpour, and Ali Abbas. Optimisation of LNG mixed-refrigerant processes considering operation and design objectives. *Computers and Chemical Engineering*, 41:123–133, 2012. ISSN 00981354. doi: 10.1016/j.compchemeng.2012.03.005. URL `http://dx.doi.org/10.1016/j.compchemeng.2012.03.005`.

International Gas Union IGU. Prospects for Natural Gas: Identifying the key developments that will shape the gas market in 2050. page 32, 2015a.

International Gas Union IGU. News , views and knowledge on gas – worldwide World LNG Report 2015 Edition. Technical report, 2015b.

Jørgen Bauck Jensen and Sigurd Skogestad. Optimal operation of simple refrigeration cycles. Part I: Degrees of freedom and optimality of sub-cooling. *Computers and Chemical Engineering*, 31(5-6):712–721, 2007. ISSN 00981354. doi: 10.1016/j.compchemeng.2006.12.003.

Jørgen Bauck Jensen and Sigurd Skogestad. Steady-State Operational Degrees of Freedom with Application to Refrigeration Cycles. *Industrial & Engineering Chemistry Research*, 48(14):6652–6659, jul 2009. ISSN 0888-5885. doi: 10.1021/ie800565z. URL `http://dx.doi.org/10.1021/ie800565z`.

Ravindra S. Kamath, Lorenz T. Biegler, and Ignacio E. Grossmann. An equation-oriented approach for handling thermodynamics based on cubic equation of state in process optimization. *Computers & Chemical Engineering*, 34(12):2085–2096, dec 2010. ISSN 00981354. doi: 10.1016/j.compchemeng.2010.07.028. URL `http://www.sciencedirect.com/science/article/pii/S0098135410002760`.

Susanne V. Krichel and Oliver Sawodny. Dynamic modeling of compressors illustrated by an oil-flooded twin helical screw compressor. *Mechatronics*, 21(1):77–84, 2011. doi: 10.1016/j.mechatronics.2010.08.004.

Alexander Leguizamon. Robust implementation of LNG refrigeration cycles. Technical report, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2015.

Mokhatab, Saeid., William A. Poe, and James G Speight. *Hand book for Natural Gas Transmission and Processing*. Elsevier, Oxford, UK, 1st edition, 2006. ISBN 13:978-0-75066-7776-9.

L. T. Narrawaf, G. W. Barton, and J. D. Perkins. Interaction between process design and process control: economic analysis of process dynamics. *J. Proc. Cont.*, 1: 243–250, 1991.

P. Nekså, E. Brendeng, M. Drescher, and B. Norberg. Development and analysis of a natural gas reliquefaction plant for small gas carriers. *Journal of Natural Gas Science and Engineering*, 2(2-3):143–149, July 2010. ISSN 18755100. doi: 10.1016/j.jngse.2010.05.001. URL `http://www.sciencedirect.com/science/article/pii/S1875510010000326`.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer Series in Operation Research and Financial Engineering, New York, NY, USA, second edi edition, 2006. ISBN 0387303030 (hd. bd.)\n9780387303031. doi: 10.1007/978-0-387-40065-5. URL `http://www.loc.gov/catdir/enhancements/fy0818/2006923897-d.html$\delimiter"026E30F$nhttp://www.loc.gov/catdir/enhancements/fy0818/2006923897-t.html`.

André Péneloux, Evelyne Rauzy, and Richard Fréze. A consistent correction for Redlich-Kwong-Soave volumes. *Fluid Phase Equilibria*, 8(1):7–23, jan 1982. ISSN 03783812. doi: 10.1016/0378-3812(82)80002-2. URL `http://www.sciencedirect.com/science/article/pii/0378381282800022`.

Prausnitz, John M., Poling Bruce E., and O'Connell John P. *The properties of gases & liquids*. Number 4. McGraw-Hill, fifth edition, 2001. ISBN 0071499997. doi: 10.1016/0894-1777(88)90021-0.

Jafar Sadeq, Horacio A. Duarte, and Robert W. Serth. Anomalous Results ? Technical report, Texas A&M University Kingsville, College Station, Texas, 1997.

J R Sauls and R Sauls. PERFORMANCE CHARACTERIST ICS OF FIXED VOLUME RATIO COMPRESSORS. In *International Compressor Engineering Conference*, pages 201–207, 1982. URL `http://docs.lib.purdue.edu/icechttp://docs.lib.purdue.edu/icec/394`.

Sigurd Skogestad. Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507, oct 2000. ISSN 09591524. doi: 10.1016/S0959-1524(00)00023-8. URL `http://www.sciencedirect.com/science/article/pii/S0959152400000238`.

Sigurd Skogestad. Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234, jan 2004. ISSN 00981354. doi: 10.1016/j.compchemeng.2003.08.002. URL http://www.sciencedirect.com/science/article/pii/S0098135403001984.

Sigurd Skogestad. *Chemical and Energy Process Engineering*. CRC Press, Boca Raton, Fla., 1st edition, 2009. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

Sigurd Skogestad. Economic Plantwide Control. In Pandu Rangaiah Gade and Kariwala Vinay, editors, *Plantwide Control: Recent developments and applications*, chapter 11, pages 229–251. John Wiley and Sons, Chichester, UK, 1st edition, 2012. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

Giorgio Soave. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chemical Engineering Science*, 27(6):1197–1203, jun 1972. ISSN 00092509. doi: 10.1016/0009-2509(72)80096-4. URL http://www.sciencedirect.com/science/article/pii/0009250972800964.

Inc. The Mathworks. Constrained nonlinear optimization algorithms, 2016a.

Inc. The Mathworks. Choosing a solver, 2016b.

Mengyu Wang, Rajab Khalilpour, and Ali Abbas. Operation optimization of propane precooled mixed refrigerant processes. *Journal of Natural Gas Science and Engineering*, 15:93–105, 2013. ISSN 18755100. doi: 10.1016/j.jngse.2013.09.007. URL http://dx.doi.org/10.1016/j.jngse.2013.09.007.

Wallace B Whiting. Effects of Uncertainties in Thermodynamic Data and Models on Process Calculations †. *Journal of Chemical & Engineering Data*, 41(96):935–941, 1996. ISSN 0021-9568. doi: 10.1021/je9600764. URL http://pubs.acs.org/doi/abs/10.1021/je9600764.

Appendices

APPENDIX

ONE

THERMODYNAMICS

A.1 MATLAB Codes

A.1.1 Thermodynamic calculations

A.1.2 Flash Calculations

```
function [cons,f_ine,h,penal] = ...
    flash(type,slack,z,x,y,T,P,VF,Zl,Zg,varargin)
% Flash calculations function. This function creates the vector of
% constraints for different possibilities of flash calculation.

% Inputs:
% Type: defines the type of flash calculation TP/PH
% Slack: defines the use of slack variables
% z: total composition of the stream
% x: liquid composition
% y: vapour composition
% T: temperature [K]
% P: pressure [bar]
% VF: vapour fraction
% Zl: liquid compressibility factor
% Zg: gas compressibility factor
% varargin: variable input which can be:
%  for PH flash
%     h: enthalpy [kJ/kmol]
%  for Slacks
%   s: slack (imaginary liquid composition)
%   beta: VLE relaxation factor

% Outputs:
% cons:list of equality constraints
% f_ine: list of inequality constraints(if slacks are on)
% h: enthalpy [kJ/kmol] for TP flash
% penal: Penalty constraint
```

```
28
29 % Initialize parameters
30 n_e = 1; % Number of equality constraints
31 n_i = 1; % Number of inequality constraints
32 NC = 5;  % Number of components
33 penal = [];
34
35 %% Constraints regardless of the type of flash and slacks
36 [phil,hl,~,errl,~]=srkseo(x,T,P,Zl);       % liquid
37 [phig,hg,~,errg,~]=srkseo(y,T,P,Zg);       % vapour
38
39 K = phil./phig; %Calculate fugacity
40
41 f{n_e}(1:NC-1) = z(1:NC-1) - VF.*y(1:NC-1) - (1-VF).*x(1:NC-1); ...
       n_e=n_e+1;   % = 0   Algebraic(NC-1): Component mass balances
42 f{n_e} = [abs(errl) abs(errg)];                                  ...
       n_e=n_e+1;   % = 0   Solution of CEOS
43
44 %% Energy balance: Uses it as a constraint for PH flash, ...
       claculates enthalpy for TP flash
45
46 if type == 'PH' %For PH flash the energy balance becomes a
47     a=1;            % Initialize index for variable inputs
48     h=varargin{a};  % Extract h from the variable input
49     f{n_e} = (h - VF*hg - (1-VF)*hl)*10^-3; ...
                                      n_e=n_e+1;   % = 0   ...
         Algebraic (1): Energy balance;
50
51 elseif type == 'TP' % For TP flash the energy balance is ...
       calculated based on both the liquid and gas enthalpies
52     a=0;    %Initialize index for variable inputs
53     [~,~,hl,~]=srks(x,T,P,1);       % liquid
54     [~,~,hg,~]=srks(y,T,P,2);       % vapour
55     h=hg*VF+hl*(1-VF); % Value of enthalpy for TP flash
56
57 else
58     disp('Error specifying flash')
59 end
60
61 %% Slack variables: relaxes the VLE constraints when slack ...
       variables are present
62 % For calculations that involve only one phase or are very close ...
       to it, it is necessary to relax the constraints
63 % The relaxation for this problem is based only on an imaginary ...
       liquid
64 % phase of composition [s]. The VLE calculations are relaxed by ...
       a factor
65 % [beta] which allows the VLE constraints to be continous at all ...
       points.
66 % The inequality constraint allows to carry out this relaxation ...
       and by
67 % meeting the slack constraint the value of beta becomes 1, thus ...
       meeting
68 % the VLE constraint.
69
70 if slack == 'on '
71     s=varargin{a+1};    % Extract s from the variable input
72     beta=varargin{a+2}; % Extract beta from the variable input
```

```
73      f{n_e} = y - beta.*K.*x;
74      n_e=n_e+1;    % = 0   Algebraic Relaxed(NC): VLE
75
76      f_in{n_i}(1:NC)=beta-ones-s;                                    ...
            n_i=n_i+1;    % < 0   Relaxation
77      % Complementary constraints
78      f_in{n_i}(1:NC)=-s;                                             ...
            n_i=n_i+1;    % = 0   Slack variables
79      f_in{n_i}(1:NC)=VF-1;                                           ...
            n_i=n_i+1;
80      penal = f_in{n_i-2}*(f_in{n_i-1})';
81
82
83  elseif slack == 'off' % If slack variables aren't needed the VLE ...
        is directly calculated
84      f{n_e} = y - K.*x;                                              ...
            n_e=n_e+1;    % = 0   Algebraic (NC): VLE
85      f_in=[0];           % Initialize inequality vector
86
87  else
88      disp('Error specifying slacks')
89  end
90
91  %% Extract vectors of constrains
92  % The constraints are transposed in order for them to be used by ...
        fmincon
93  % without further modification.
94  % The order is of them is the same as their order of appearance ...
        throughout the code.
95
96  cons=[];
97  n_e;
98  for i = 1:n_e-1
99      cons=[cons;f{i}'];
100 end
101
102 f_ine=[];
103 for i = 1:n_i-1
104     f_ine=[f_ine;f_in{i}'];
105 end
106
107 end
```

APPENDIX

TWO

EO MODEL

B.1 MATLAB Code

```
function [J,c,ceq,Wc,QHA_07,Cross,filename,THA_01]=f(w,p1,p2,p3)
%% Main Model code
% This code includes a steady state simulation of the whole mini ...
    LNG plant.
% In this code all the models for the different units are solved
% simultaneously. The initialization comes from the files Main.m and
% Initialpoint.m.

%%  Initialize variables

N_s=17;% Number of streams
%Initialize variables
P    = zeros(1,N_s); % Pressures [pa]
T    = zeros(1,N_s); % Stream temperature [K]
n    = ones(1,N_s);  % Flows in [kmol/s]
h    = zeros(1,N_s); % Molar enthalpy [Kj/kmol]
VF   = zeros(1,N_s); % Vapor Fraction
H    = zeros(1,N_s); % Total Enthalpy [Kj]
x    = zeros(N_s,5); % Liquid Composition
y    = zeros(N_s,5); % Vapor Composition
z    = zeros(N_s,5); % Total Composition
s    = zeros(N_s,5); % Liquid slack
beta = zeros(N_s,5); % Relaxation


% Set known compositions
z(1,:)  = [0.1074 0.3292 0.4096 0.1345 0.0193];
z(17,:) = [0.1074 0.3292 0.4096 0.1345 0.0193];


% Set known Temperatures
T(2) = 35 + 273.15;
```

```
T(6) = 273.15 - 75;
T(8) = 273.15 - 146.6;


% Set known vapor fractions
VF(1)  = 1;
VF(5)  = 1;
VF(4)  = 0;



% Define heat from the environment
Q_env = 80.5852*p3;


%% Calculate the BOG requirements

Tin_bog = -32 + 273.15+p1; %Inlet temperature [K]
m_bog=610*p2; %Gas mass flow [Kg/h] (Neskaa, 2010)
[ QHA_07] = BOGC( Tin_bog, m_bog );

%% Data extraction from the initial point vector
%Define sets for different kinds of flashes
TP  = [ 6 ];  %TP Flash
PH  = [ 3, 7, 11, 13, 14, 15]; %PH Flash
PHs = [ 16, 17];%PH Flash with slack variables
F = [TP, PH, PHs]; % Total set of flash calculations
FH= [PH,PHs]; % Set of all PH flash calculations

%Define important parameters
NC=5;

T(1) = w(1);
T1id = w(2);
a=2;
for i = F
    x(i,1:NC-1)=w(a+1:a+NC-1);
    x(i,NC)=1-sum(x(i,1:NC-1));
    y(i,1:NC-1)=w(a+NC:a+2*NC-2);
    y(i,NC)=1-sum(y(i,1:NC-1));
    VF(i)=w(a+2*NC-1);
    Zl(i)=w(a+2*NC);
    Zg(i)=w(a+2*NC+1);
    a=a+2*NC+1;
    if ismember(i,FH)
        T(i)=w(a+1); a=a+1;
        if ismember(i,PHs)
            s(i,:)=w(a+1:a+NC);
            beta(i,:)=w(a+NC+1:a+2*NC);
            a=a+2*NC;
        end
    end
end


T(10)= T(8);
T(9) = T(8);
THA_01 = T(3);
```

```
%% Optimization variables

% Set known flows
N = 0.0434*w(a+1); %Refrigerant flow [Kmol/s]
n = N*n;

% Define split fraction
Sf = w(a+2);

% Set the pressures for the whole system
lp = [7 12 11 13 14 15 16 17];
hp = [1 2 3 4 5 6 8 9 10];
P(lp) = 3e5*w(a+3);
P(hp) = 20e5*w(a+4);

% Define heat for HA-01
QHA_01 = 250*w(a+5);

%% Model Initial parameters

% The following section of the code includes the code for the ...
   model of the
% system, it sets all the equations of the system in terms of ...
   non linear
% equality constraints for the optimization
n_e = 1; %Initial number of equality constrains
n_i = 1; %Initial number of inequality constrains
PH='PH';
TP='TP';
Slack='on ';
noSlack='off';
%% Compressor

% Pressure relation calculation
rel = P(1)/P(17);
p = [0.0015   -0.0314    0.1911    0.4217]; %3rd degree polynomial.
eta = polyval(p,rel);


% Isoentropic calculation
[Z(17),~,h(17),~,s17]= srks(z(17,:),T(17),P(17),2);
[~,~,h1id,~,s1id]= srks(z(1,:),T1id,P(1),2);
f{n_e}= (s17 - s1id)*10^-2; n_e=n_e+1;

% Real calculations
Wc =(h1id-h(17))/eta;           % Work calculation for
h1real = h(17) + Wc;            % Real output enthalpy
[Z(1),~,h(1),~]  = srks(z(1,:),T(1),P(1),2);
Wc = Wc*n(17);
f{n_e}= (h1real - h(1))*10^-4; n_e=n_e+1;

%Assign variables
n(2)=n(1);
z(2,:)=z(1,:);
H(1) = h(1)*n(1);

%% MR Precooler

```

```
147 % Vapour only
148 z(2,:)=z(1,:);
149 y(2,:)=z(1,:);
150 [~,~,h(2),~]= srks(z(2,:),T(2),P(2),2);
151 H(2) = h(2)*n(2);
152 QMR = H(2)-H(1);
153 VF(2) = 0;
154
155 %Assign values for the next stream
156 z(3,:) = z(2,:);
157 n(3) = n(2);
158
159 %% HA - 01
160 % Energy balance
161 H(3) = H(2)-QHA_01;
162 h(3) = H(3)/n(3);
163
164 % PH Flash without slack variables
165 [f{n_e},~,~] = flash(PH,noSlack,z(3,:),x(3,:),y(3,:),T(3),P(3),...
166     VF(3),Zl(3),Zg(3),h(3));
167 n_e=n_e+1; %Increment constraints indeces
168
169 %Assign flows and compositions
170
171 %Vapour stream
172 z(5,:) = y(3,:);
173 y(5,:) = y(3,:);
174 n(5) = n(3)*VF(3);
175 T(5) = T(3);
176 [Z(5),~,h(5),~] = srks(z(5,:),T(5),P(5),2);
177 H(5)=h(5)*n(5);
178
179
180 %Liquid stream
181 z(4,:)=x(3,:);
182 x(4,:)=x(3,:);
183 n(4)=n(3)*(1-VF(3));
184 T(4)=T(3);
185 [Z(4),~,h(4),~] = srks(z(4,:),T(4),P(4),1);
186 H(4)=h(4)*n(4);
187
188 %% Valve FCV-01
189 % Isoenthalpic Valve
190 h(7) = h(4);
191 n(7) = n(4);
192 z(7,:) = z(4,:);
193
194 % PH flash without slack variables
195 [f{n_e},~,~] = flash(PH,noSlack,z(7,:),x(7,:),y(7,:),T(7),P(7),...
196     VF(7),Zl(7),Zg(7),h(7));
197 n_e=n_e+1; %Increment constraints indeces
198
199 H(7)=n(7)*h(7);
200
201
202 %% Heat exchanger HA-02 first side
203 %Calculate stream 6
204 z(6,:)=z(5,:);
```

```
n(6)=n(5);

% TP Flash without slack
[f{n_e},~,h(6)] = ...
    flash(TP,noSlack,z(6,:),x(6,:),y(6,:),T(6),P(6),...
     VF(6),Zl(6),Zg(6));
n_e=n_e+1; %Increment constraints indeces

% Energy Calculation
H(6)=h(6)*n(6);

%QHA_02 Duty
QHA_02=H(5)-H(6);

%% Calculate HA-04 first side
%Calculate stream 7
z(8,:) = z(6,:);
x(8,:) = z(8,:);
n(8) = n(6);
VF(8) = 0; % Liquid only
[~,~,h(8),~] = srks(z(8,:),T(8),P(8),1);
H(8) = h(8)*n(8);
QHA_04 = H(6)-H(8);

%Split

%Stream 10
z(10,:) = z(8,:);
x(10,:) = x(8,:);
VF(10) = VF(8);
n(10) = n(8)*Sf;
h(10) = h(8);
H(10) = h(10)*n(10);
T(10) = T(8);

%Stream 9
z(9,:) = z(8,:);
x(9,:) = x(8,:);
VF(9) = VF(8);
n(9) = n(8)*(1-Sf);
h(9) = h(8);
H(9) = h(9)*n(9);
T(9) = T(8);

%% FCV-04 // FCV-05
%Assign values for stream 11
z(11,:)=z(9,:);
n(11)=n(9);
h(11)=h(9);

% PH flash without slack variables
[f{n_e},~,~] = ...
    flash(PH,noSlack,z(11,:),x(11,:),y(11,:),T(11),P(11),...
     VF(11),Zl(11),Zg(11),h(11));
n_e=n_e+1; %Increment constraints indeces

% Total enthalpy
H(11)=n(11)*h(11);
```

```
261
262 %Stream 12 - Properties are the same as for stream 11
263 z(12,:) = z(10,:);
264 n(12) = n(10);
265 h(12) = h(10);
266 T(12) = T(11);
267 x(12,:) = x(11,:);
268 y(12,:) = y(11,:);
269 VF(12) = VF(11);
270 H(12) = n(12)*h(12);
271
272 %% HA-04 Second side
273 n(13)=n(12);
274 H(13)=H(12)+QHA_04;
275 h(13)=H(13)/n(13);
276 z(13,:)=z(12,:);
277
278 %PH flash without slack variables
279 [f{n_e},~,~] = flash(PH,noSlack,z(13,:),x(13,:),y(13,:),T(13),...
280     P(13),VF(13),Zl(13),Zg(13),h(13));
281 n_e=n_e+1; %Increment constraints indeces
282
283 H(13)=n(13)*h(13);
284
285 %% HA-07 Refrigerant side
286 z(14,:)=z(11,:);
287 n(14)=n(11);
288 H(14)=H(11)-QHA_07;
289 h(14)=H(14)/n(14);
290
291 %PH flash without slack variables
292 [f{n_e},~,~] = flash(PH,noSlack,z(14,:),x(14,:),y(14,:),T(14),...
293     P(14),VF(14),Zl(14),Zg(14),h(14));
294 n_e=n_e+1; %Increment constraints indeces
295
296
297 %% Mixer
298 %Balances
299 n(15)=n(7)+n(13)+n(14);
300 z(15,:)= (n(7).*z(7,:)+n(13).*z(13,:)+n(14).*z(14,:))/n(15);
301 H(15)=H(7)+H(13)+H(14);
302 h(15)=H(15)/n(15);
303
304 %PH flash without slack variables
305 [f{n_e},~,~] = flash(PH,noSlack,z(15,:),x(15,:),y(15,:),T(15),...
306     P(15),VF(15),Zl(15),Zg(15),h(15));
307 n_e=n_e+1; %Increment constraints indeces
308
309
310 %% QHA_02 second side
311 z(16,:)=z(15,:);
312 n(16)=n(15);
313 H(16)=H(15)+QHA_02;
314 h(16)=H(16)/n(16);
315
316 %PH flash with slack variables
317 [f{n_e},fi{n_i},~,Pen1] = ...
       flash(PH,Slack,z(16,:),x(16,:),y(16,:),T(16),...
```

```
318     P(16),VF(16),Zl(16),Zg(16),h(16),s(16,:),beta(16,:));
319 n_e=n_e+1; n_i=n_i+1; %Increment constraints indeces
320
321
322 %% Heat Transfer from the environment
323 %It is calculated by carrying out an energy balance
324
325 H(17) = Q_env + H(16); % This corresponds to the heat from the ...
       environment
326 n(17) = n(16);
327 h(17) = H(17)/n(17);
328
329 %PH flash with slack variables
330 [f{n_e},fi{n_i},~,Pen2] = ...
       flash(PH,Slack,z(17,:),x(17,:),y(17,:),T(17),...
331     P(17),VF(17),Zl(17),Zg(17),h(17),s(17,:),beta(17,:));
332 n_e=n_e+1; n_i=n_i+1; %Increment constraints indeces
333
334 x(1,:)=x(17,:);
335 y(1,:)=y(17,:);
336 VF(1)=VF(17);
337 %% Operational constratints
338
339 % Only vapour entering to the compressor
340 f{n_e} = 1-VF(17);n_e=n_e+1;
341 N0 = n_i;
342
343 % Constraints regarding pressure relation boundaries 2 <= rel <=10
344 fi{n_i} = rel - 10; n_i=n_i+1;% 31
345 fi{n_i} = 2 - rel;  n_i=n_i+1;% 32
346
347 % Boundaries for the splitflow
348 fi{n_i} = n(9)/(n(10)+n(9)) - 1;n_i=n_i+1; %33
349 fi{n_i} = 0 - n(9)/(n(10)+n(9));n_i=n_i+1; %34
350
351 % Constraints regarding pressure values
352
353 % Low pressure between 2 and 4 bar
354 fi{n_i} = 1.5/3 - w(a+3);n_i=n_i+1; % 35
355
356 % High pressure between 18 and 25 bar
357 fi{n_i} = w(a+4) - 25/20;n_i=n_i+1; % 36
358
359 % Stream 3 output T between -44 C and 35 C
360 fi{n_i} = T(3) - (35 + 273.15);n_i=n_i+1; % 37
361 fi{n_i} = (273.15 -44) - T(3);n_i=n_i+1; % 38
362
363 N1= n_i-1;
364 %% Temperature crosses
365 Cross=[];
366
367 %BOG Temperatures from BOG.m
368 TBOGin = Tin_bog; %Inlet temperature [K]
369 TBOGout = -154 +273.15; %Outlet temperature [K]
370
371 % HA - 02
372 Cross(1) = -T(16) + T(5);
373 Cross(2) = -T(15) + T(6);
```

```
374
375 % HA - 04
376 Cross(3) = - T(13) + T(6);
377 Cross(4) = -T(12) + T(8);
378
379 % HA - 07
380 Cross(5) = - T(14) + TBOGin;
381 Cross(6) = - T(11) + TBOGout;
382
383 P_c=sum(abs(Cross));
384 %% Heat exchanger sizes
385 %Load calculated values
386 filename='HXSizing.mat';
387 load(filename)
388
389 %The following contraints make sure that the value for UA for each
390 %exchanger is constant
391
392 %HA-02
393 [ UA_HA_02_c ] = hxsizing( QHA_02, T(5), T(6), T(15), T(16) );
394 f{n_e} = UA_HA_02-UA_HA_02_c;n_e=n_e+1;
395
396 %HA-04
397 [ UA_HA_04_c ] = hxsizing( QHA_04, T(6), T(8), T(12), T(13) );
398 f{n_e} = UA_HA_04-UA_HA_04_c;n_e=n_e+1;
399
400 %HA-07
401 [ UA_HA_07_c ] = hxsizing( -QHA_07,  T(11), T(14), TBOGin, TBOGout);
402 f{n_e} = (UA_HA_07-UA_HA_07_c);n_e=n_e+1;
403
404 %% Objective function
405 W0 = 250; %Initial value for dimensionless work
406 M1 = 10^6;
407 M2 = 10^3;
408
409 J = (Wc/W0)+M1*(Pen1+Pen2)+M2*P_c;
410
411 %% Define the constraints vectors
412 const_i=[];
413 for i = 1:n_i-1
414     const_i=[const_i;fi{i}];
415 end
416
417 const_e=[];
418 for i = 1:n_e-1
419     const_e=[const_e;f{i}];
420 end
421 c=[const_i]; ceq=[const_e];
```

B.2 Compressor curves

```
%% Compresor curves
clear all
close all
% This script carries out the regression for the compressor curves
% Curve points
x      = [ 2    5    6.67 8    9    10];
eta    = [ 0.69 0.78 0.74 0.70 0.69 0.68];
lambda = [ 0.98 0.93 0.90 0.89 0.87 0.86];

% Regression
p = polyfit(x,eta,3)
l = polyfit(x,lambda,1);

x2 = 2:0.2:10;
eta2 = polyval(p,x2);

%% Plot curves

fig(1) = figure(1);
plot(x2,eta2)
hold on
scatter(x,eta)
title('Compressor curve','Interpreter','latex')
xlabel('$P_{out}/P_{in}$','Interpreter','latex')
ylabel('$\eta$','Interpreter','latex')
legend('Regression', 'Points')
```

B.2.1 Heat exchangers

```
function [ UA, NTU_A, NTU_B ] = hxsizing( Q, T1in, T1out, T2in, ...
    T2out )
%This is hxsizing.m - This file corresponds to the recursive ...
    routine to
%determnine UA values for all heat exchangers.
% %   Input arguments
%   Q = Heat duty [Kj/s]
%   T1in = Inlet temperature for side 1 [K]
%   T1out = outlet temperature for side 1 [K]
%   T2in = Inlet temperature for side 2 [K]
%   T2out = Outlet temperature for side 2 [K]
% %  Output arguments
% UA = Overall heat transfer coefficient*Area [Kj/(sK)]

% Find cold and hot side



% Calculate LMTD
% Counter-current: DeltaA: T1in - T2out. DeltaB: T2in - T1out.
DeltaA = abs(T1in - T2out);
DeltaB = abs(T2in - T1out);
```

```
21
22 LMTD=(DeltaA-DeltaB)/log(DeltaA/DeltaB);
23
24 % Calculate UA based on Q=UALMTD
25
26 UA=Q/LMTD;
27
28 % Calculate NTU
29
30 NTU_A = DeltaA/LMTD;
31 NTU_B = DeltaB/LMTD;
32
33 end
```

B.2.2 BOG Calculations

```
1  function [ QBOG,n_bog ] = BOGC( Tin_bog, m_bog )
2  % Calculation of the energy requirements of the BOG for a ...
       variable inlet
3  % temperature and flow.
4
5  z_bog=[0.11 0.89 0 0 0]; %Gas Mole compositions
6  M_N= 28; %Nitrogen (N2) molecular mass [Kg/Kmol]
7  M_C1=16; %Methane (CH4) molecular mass [Kg/Kmol]
8  n_bog= m_bog/(z_bog(1)*M_N+z_bog(2)*M_C1)*(1/3600); %BOG Molar ...
       Flow [Kmol/s]
9
10
11 Tout_bog = -154 +273.15; %Outlet temperature [K]
12 P_bog = 18.2e5;% Pressure [Pa] (Assumed constant through the ...
       exchanger)
13
14 [~,~,Hin,~]= srks(z_bog,Tin_bog,P_bog,2); %Inlet SRK ...
       Calculations. Initial phase vapor only.
15 [~,~,Hout,~]= srks(z_bog,Tout_bog,P_bog,1); %Outlet SRK ...
       Calculations. Outlet liquid only.
16
17 QBOG= n_bog*(Hout-Hin); %Heat flow from the gas [Kj/s]
18 end
```

APPENDIX

THREE

OPTIMIZATION

C.1 MATLAB Code

C.1.1 Optimization function

```
function [x,f,eflag,outpt,p0] = ...
    runobjconst(x0,lb,ub,opts,p1,p2,p3,fobj)
    % This function runs the optimization problem and sets it up ...
       for a more
    % efficient use of parallel computation.
    % Inputs:
    %   x0: Initial point
    %   lb: Lower bound
    %   ub: Upper bound
    %   opts: Algorithm options
    %   Perturbations:
    %       p1: Increment in the inlet temperature of BOG [K]
    %       p2: Modifier for the flow of BOG
    %       p3: Modifier for the incoming environment heat
    %   fobj: optimization mode:
    %       fobj = 1: normal optimzation
    %       fobj = 2: dummy objective function (simulation)
    %
    % Outputs:
    %   x: Optimal point
    %   f: Optimal objective function value
    %   eflag: Optimizers flag
    %   outpt: Optimizer output summary
    %   p0: point at iteration nr 10 for future initialization
    %% Initial parameter definition

    if nargin == 1
        opts = [];
        lb = [];
        ub = [];
```

```
29      end
30      xLast = [];
31      myf = []; % Use for objective at xLast
32      myc = []; % Use for nonlinear inequality constraint
33      myceq = []; % Use for nonlinear equality constraint
34      p0 = []; % Intermediate state for next iterations
35      opts.OutputFcn=@outfun;
36
37
38      fun = @objfun; % the objective function, nested below
39      cfun = @constr; % the constraint function, nested below
40
41      %% Optimization
42      switch fobj
43          case 1
44              [x,f,eflag,outpt] = fmincon( @(w)fun(w,p1,p2,p3)  ...
                    ,x0,[],[],[],[],lb,ub,@(w)cfun(w,p1,p2,p3),opts);
45          case 2
46              [x,f,eflag,outpt] = fmincon( ...
                    '1',x0,[],[],[],[],lb,ub,@(w)cfun(w,p1,p2,p3),opts);
47      end
48
49      %% Definition of the objective function based on the model
50      function y = objfun(x,p1,p2,p3)
51          if ~isequal(x,xLast) % Check if computation is necessary
52              [myf,myc,myceq] = MainEO(x,p1,p2,p3);
53              xLast = x;
54          end
55          y = myf;
56      end
57      %% Definition of the constraints based on the model
58      function [c, ceq] = constr(x,p1,p2,p3)
59          if ~isequal(x,xLast) % Check if computation is necessary
60              [myf,myc,myceq] = MainEO(x,p1,p2,p3);
61              xLast = x;
62          end
63          c = myc;
64          ceq = myceq;
65      end
66      %% Function to extract an intermediate point data
67      function stop = outfun(x,optimValues,state)
68          stop=false;
69          if optimValues.iteration == 15
70              p0 = x;
71          end
72      end
73  end
```

C.1.2 Initial point

```
%% Initial point script
clear all
clc

% Load information from the steady state results
filename='SSresults.mat';
[w0,lb,ub] = Initialp(filename);

% Run optimization for the base case
tic
options = optimoptions('fmincon','Algorithm', 'interior-point', ...
    'Diagnostics','on','PlotFcns',{...
@optimplotfval,@optimplotfirstorderopt,@optimplotconstrviolation},...
'FunValCheck','on' ,'TolCon', 1e-4,'TolFun', 1e-4, 'MaxIter', ...
    16000,...
'MaxFunEvals', 150000, 'TolX', 1e-10, ...
    'ScaleProblem','obj-and-constr',...
'UseParallel',true,'FinDiffType','forward');
[w,f,eflag,outpt,w0] = runobjconst(w0,lb,ub,options, 0,1 ,1,1);
t=toc

% Save results
file='EOSS.mat';
save(file)
```

C.1.3 Perturbed calculations

```
%% Main optimization
close all
clear all
clc

%% Set up initial point and boundaries
load('EOSS.mat','w0','lb','ub')
filename='SSresults.mat';
%[w0,lb,ub] = Initialp(filename);

%% Set up results structure
ActCon=struct('Flag',[],'T', [],'MBOG',[],'QEnv',[], 'Act', [],...
    'Objective',[],'QBOG', ...
        [],'Cross',[],'Flow',[],'SplitFlow',[],...
    'LowP',[],'HighP',[],'T3',[],'Output',[],'w', [],'Time',[]);
tol = 1e-3; % Tolerance for a constraint to be active
m=1;

% Set up perturbations order based on QBOG
QB = [];
for p1= 0:5:10
    for p2= 0.9:0.1:1.1
        Tin_bog = -32 + 273.15+p1; %Inlet temperature [K]
        m_bog=610*p2; %Gas mass flow [Kg/h] (Neskaa, 2010)
```

```
24          QB(m,1) = -BOGC( Tin_bog, m_bog );
25          QB(m,2)=p1;
26          QB(m,3)=p2;
27          m=m+1;
28      end
29  end
30  QB = sortrows(QB,1);
31
32  %% Solve perturbed optimization problems
33  m =1;
34  for j= 1:size(QB,1) % Vary on QBOG
35      for k= 0.9:0.05:1.1 % Vary on QEnv
36          options = optimoptions('fmincon','Algorithm', ...
                'interior-point',...
37              'Diagnostics','on','PlotFcns',{@optimplotfval,...
38              @optimplotfirstorderopt,@optimplotconstrviolation,...
39              @optimplotstepsize},'FunValCheck','on' ,'TolCon',...
40              1e-4,'TolFun', 1e-5, 'MaxIter', 1000,'MaxFunEvals', ...
                    150000,...
41              'TolX', 1e-8, 'ScaleProblem','obj-and-constr',...
42              'UseParallel',true,'FinDiffType','forward',...
43              'OutputFcn',@outfun);
44
45          %Solve optimization
46          tic
47          [w,f,eflag,outpt,w0] = runobjconst(w0,lb,ub,options,...
48              QB(j,2),QB(j,3),k,1);
49          t=toc
50
51          % Save optimal and feasible points
52          if eflag == 1 || eflag == 2
53              [~,cons,~,Wc,QBOG,Cross,fil,THA_01]=MainEO(w,...
54                  QB(j,2),QB(j,3),k);
55              A = [0];
56              for n=drange(31:1:38)
57                  if abs(cons(n))<= tol
58                      A = [A n];
59                  end
60              end
61          else
62              A = [-1];
63              Wc= [-1];
64          end
65
66          % Save information in a structure
67          ActCon(m).Flag=eflag;
68          ActCon(m).T=-32 + 273.15+QB(j,2);
69          ActCon(m).MBOG=610*QB(j,3);
70          ActCon(m).QEnv=80.5852*k;
71          ActCon(m).Act=[A];
72          ActCon(m).Objective=Wc;
73          ActCon(m).QBOG=-QBOG;
74          ActCon(m).Cross=Cross;
75          ActCon(m).Output=outpt;
76          ActCon(m).Flow=w(130);
77          ActCon(m).SplitFlow=w(131);
78          ActCon(m).LowP=w(132);
79          ActCon(m).HighP=w(133);
```

```
80          ActCon(m).T3=THA_01;
81          ActCon(m).w=w;
82          ActCon(m).Time=t;
83          m=m+1
84      end
85  end
86
87  %% Save results to file
88  file='Actcons.mat';
89  save(file)
```

APPENDIX

FOUR

RESULTS AND ANALYSIS

D.1 MATLAB Code

```
%% Extract data
clear
clc
close all
load('Actcons.mat','ActCon')
%load('AC.mat','ActCon')
X = [];
Y = [];
Z = [];
A = [];
C = [];
c=[];
m = 1;
l = ones(1,6);
w = 1;
ceq=[];
FsX =[];
FsY =[];
for i= 1:1:size(ActCon,2)
    if ActCon(i).Flag == 1
        X(m) = ActCon(i).QEnv;
        Y(m) = ActCon(i).QBOG;
        Z(m) = ActCon(i).Objective;
        A{m} = ActCon(i).Act;
        I(m) = ActCon(i).T- 273.15;
        J(m) = ActCon(i).MBOG;
        P1(m) = ActCon(i).HighP*20;
        P2(m) = ActCon(i).LowP*3;
        F(m) = ActCon(i).Flow;
        Sf(m) = ActCon(i).SplitFlow*65;
        T(m) = ActCon(i).T3 - 273.15;
        [~,~,ceq(i,:)]=MainEO(ActCon(i).w,(ActCon(i).T-273.15+32),...
            ActCon(m).MBOG/610,ActCon(m).QEnv/80.5852);
```

```
34          % Active constraints
35          a = A{m};
36          if a == [0] % No active constraints
37              n = 1;
38              QE_{n}(l(n)) = X(m);
39              QBOG_{n}(l(n)) = Y(m);
40              c(n,:) = [1 0 0]; % Red
41              l(n)=l(n)+1;
42          end
43          if size(a,2) == 2
44              if a == [0,35] % Lower bound High P
45                  n = 6;
46                  QE_{n}(l(n)) = X(m);
47                  QBOG_{n}(l(n)) = Y(m);
48                  c(n,:)=[0 0 1];
49                  l(n)=l(n)+1;% Blue
50              elseif a == [0,36] % Lower bound Active Charge
51                  n = 3;
52                  QE_{n}(l(n)) = X(m);
53                  QBOG_{n}(l(n)) = Y(m);
54                  c(n,:)=[0 1 0]; % Green
55                  l(n)=l(n)+1;
56              elseif a == [0,39] % Upper bound High P
57                  n = 5;
58                  QE_{n}(l(n)) = X(m);
59                  QBOG_{n}(l(n)) = Y(m);
60                  c(n,:) = [0 1 1];% Yellow
61
62                  l(n)=l(n)+1;
63
64              end
65          elseif size(a,2) == 3
66              if a == [0,36,40] % Lower bound Active Charge  Lower ...
                    bound High P
67                  n = 2;
68                  QE_{n}(l(n)) = X(m);
69                  QBOG_{n}(l(n)) = Y(m);
70                  c(n,:) = [1 0 0]; % Red
71                  l(n)=l(n)+1;
72              elseif a == [0,36,39] % Lower bound Active Charge % ...
                    Upper bound High P
73                  n = 4;
74                  QE_{n}(l(n)) = X(m);
75                  QBOG_{n}(l(n)) = Y(m);
76                  c(n,:) = [1 0 1]; % Magenta
77
78                  l(n)=l(n)+1;
79              end
80          end
81          m = m+1;
82      else
83          FsX(w) = ActCon(i).QEnv;
84          FsY(w) = ActCon(i).QBOG;
85          w = w+1;
86      end
87  end
88
89
```

```
%% Plot 1
np = 20;
plotres( X, Y, Z, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'W_{Compressor}[kj/s]',1,'Wc'  )

%% Plot 2

ilin = linspace(min(I),max(I),np);
jlin = linspace(min(J),max(J),np);
[i, j] = meshgrid(ilin,jlin);

k = griddata(I',J', Y', i,j,'cubic');

figure(2)

axis tight
hold on
grid on

xlabel('Temperature_{BOG,In}[^\circC]', 'FontSize',12)
ylabel('Flow_{BOG}[kmol/s]', 'FontSize',12)
zlabel('Q_{BOG}[kJ/s]')

contour(i,j,k, 'ShowText','on')
clabel([] ,[] ,'FontSize',12)
printfig(figure(2),'Qbog')


%% Plot 3

figure(3)
axis tight;
hold on
grid on
xlabel('Q_{Env}[kJ/s]', 'FontSize',12)
ylabel('Q_{BOG}[kJ/s]', 'FontSize',12)
% Active constraints
for n = 1:size(QE_,2)
    scatter(QE_{n}',QBOG_{n}',60,c(n,:),'filled')
end
scatter(FsX',FsY',60,[0 0 0],'filled')
legend('No active constraints', 'Failed states','Location', ...
    'best', 'FontSize',12)
legend('show')
printfig(figure(3),'Actcon')

%% Plot 4
plotres( X, Y, P2, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'P_{Low}[bar]',4,'LowP'  )
%% Plot 5
plotres( X, Y, P1, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'P_{High}[bar]',5,'HighP'  )
%% Plot 6
plotres( X, Y, F, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'Refrigerant flow (dimless)',6,'Flow'  )
%% Plot 7
plotres( X, Y, T, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'T_{3}[^\circ C]',7,'Temp3'  )
```

```
%% Plot 8
plotres( X, Y, Sf, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', 'Split ...
    fraction',8,'Splitf'  )

%% plot
Eff = (P1)./(P2);
p = [0.0015   -0.0314    0.1911    0.4217]; %3rd degree polynomial.
Eff = polyval(p,Eff);

plotres( X, Y, Eff, np,'Q_{Env}[kJ/s]' , 'Q_{BOG}[kJ/s]', ...
    'Compressor efficiency',9,'Compeff'  )
```

D.2 Plot figure

```
function [  ] = plotres( X, Y, Z, np, xl, yl, zl,i,name  )
    % This function prints the plot for a given set of ...
       perturbations and a
    % given input
    xlin = linspace(min(X),max(X),np);
    ylin = linspace(min(Y),max(Y),np);
    [x, y] = meshgrid(xlin,ylin);
    z = griddata(X',Y', Z', x,y,'cubic');

    figure(i)
    fig= figure(i);
    mesh(x,y,z)

    axis tight;
    hold on
    grid on



    FSz = 12;

    xlabel(xl,'FontSize', FSz)
    ylabel(yl,'FontSize', FSz)
    zlabel(zl,'FontSize', FSz)

    printfig(fig,name)


end
```

D.2.1 Print results

```
function [ ] = printfig( fig, name )
    % Prints figures

    units = 'centimeters';
```

```
5       pos = [0 0 10 10];
6       mode = 'Manual';
7
8
9       fig.PaperUnits = units;
10      fig.PaperPosition = pos;
11      fig.PaperPositionMode = mode;
12      fig.PaperSize = [pos(3),pos(4)];
13      fpath = '../Document/img/';
14      print( fig, fullfile(fpath,name), '-dpdf','-r0')
15  end
```