Modelling and optimization of the C3MR process for liquefaction of natural gas

Dag-Erik Helgestad

December 10, 2009

Abstract

A simulation model of the C3MR liquefaction process for LNG production was built in UniSim Design software. The model was built in parallel with PhD-student and co-supervisor Magnus Glosli Jacobsen, arriving at almost identical simulation models. The process simulation was based on a process train with production capacity of 8.4 MTPA. The C3MR process model in UniSim Design had problems with robustness in terms of allowing very limited changes in process variables before having problems converging. The tolerances of process unit calculations were set as tight as possible with regards to maintaining a converging flowsheet.

The potential degrees of freedom for the process were identified according to the reference tables of Skogestad and Jensen.[1] The process specifications in the simulation model were counted and were found to be in accordance with the number of potential DOF's. The degrees of freedom available for optimization were determined by subtracting specified variables as well as variables that had no steady-state effect on the process.

Optimization of the process was attempted using both the built-in optimizer of UniSim Design and the MATLAB function *fmincon* by interfacing the two software. Problems arose during optimization due to the loose tolerances in the simulation model, resulting in inaccurate objective function calculations. The inaccuracy in UniSim Design affected the optimization routine by providing incorrect gradients and thus causing the optimizer to fail. Additionally, the UniSim optimizer was not able to meet the inequality constraints posed on the optimization problem. Results were obtained when the optimization was lifted out to MATLAB, though they were inaccurate and did not satisfy the Karush-Kuhn-Tucker conditions for optimality. Nevertheless, the results give an indication of the optimal region of the process, as well as proving the design to be close to optimum.

The UniSim process model was neither accurate nor robust enough to be well suited for optimization and detailed analysis of the C3MR process. With regards to further work, it is recommended to rebuild the model using alternative simulation software, preferrably providing a higher level of transparency with respect to process equations and optimization procedures.

A self-optimizing control structure for the process was not determined in this project, but the procedure and simplifying assumptions were discussed.

Acknowledgement

I would like to thank my supervisor for this project, Professor Sigurd Skogestad at the Process Systems Engineering group at NTNU, Trondheim, for support regarding the project work throughout the semester. Equally, I wish to thank my co-supervisor and *almost*-daily contact, P.h.D. student Magnus Glosli Jacobsen, for help and insight in LNG processes as well as troubleshooting when needed.

Finally, I would like to thank my fellow students Elisabeth Hyllestad, Martin Buus Jensen and Anders Haukvik Røed for keeping a good spirit and healthy, fun atmosphere at our office. It has been a pleasure.

Contents

List of Figures

List of Tables

1 Introduction

The world's energy demand is expected to increase by 40% between 2007 and 2030.[4] The rate of new oil field discoveries in the world has been sinking rapidly since the 1960'ies, while the world production has been increasing.[5] Since the millenium, most non-OPEC and non-FSU countries have peaked in production and are declining.[6] Even though output from Russia is growing and the OPEC countries have excess capacity, there is no doubt the world will have to rely on other energy sources in the future.[7]

Natural gas is a viable energy source, but is dependent on existing pipeline infrastructure in order to reach the consumers. Due to the large volume it is not practical or economical to transport gas by vehicles or ships. Unfortunately, the dependency of pipelines renders many proven gas reserves infeasible to exploit, as it will be too costly to extend pipelines to these remote places. Until recently, natural gas produced at fields without such infrastructure was flared as it was practically valueless.

Liquefied natural gas (LNG) is condensed natural gas at atmospheric pressure and approximately -162 °C. Natural gas in this state takes up only one 600'th of the original gaseous volume, and therefore eases transportation issues. The LNG may be transported by land or sea to a terminal connected to a pipeline infrastructure, where the LNG is re-gasified and distributed to the customers.

The process of cooling natural gas to -162 °C is highly energy demanding. Several process designs have been developed in order to optimize the production of LNG. The focus of this project is the propane pre-cooled mixed refrigerant (C3MR) process developed by Air Products and Chemicals Inc (APCI). The C3MR process is the dominant liquefaction cycle in LNG production, and has been for many years.[8]

The goal of this study was to create a model of the C3MR process in Honeywell's UniSim Design software, analyze the degrees of freedom avaible and subsequently optimize the operation of the process with given design parameters. After successful optimization, a self-optimizing control structure can be determined for the process, and alternative control structures can be tested in dynamic simulations.

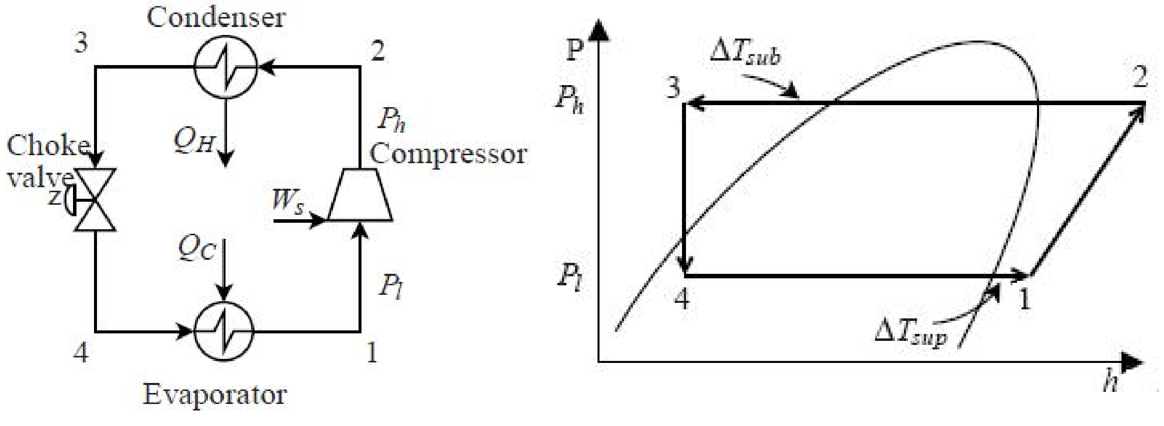
2 Background

This section introduces the theory behind refrigeration systems, using the vapor compression cycle as an example, as well as discussing the value chain of liquefied natural gas (LNG) and introducing the numerical optimization techniques used in the project work. It is important to note that none of these sections are meant to completely cover the topic at hand, rather give a short introduction and provide understanding for the later sections of the report. References to more extensive descriptions and studies have been given.

2.1 A simple refrigeration cycle

Conventional household refrigerators and air-conditioners transfer heat from an area with low temperature to an area with higher temperature. The most common process for these familiar applications is the vapor compression cycle. Variations of the same process are used in industry in order to cool process streams to a temperature lower than that of readily available cooling media, e.g. seawater.

The vapor compression cycle is fairly simple, and consists of only four components: compressor, condenser, expansion valve and evaporator. A simple process flow diagram and the corresponding pressure-enthalpy diagram are shown in Figures 2.1a and 2.1b, respectively. The cycle has four thermodynamic states which are denoted in the figures and are explained below.



(a) Flowsheet

(b) Pressure-enthalpy diagram

Figure 2.1: The vapor compression cycle with corresponding pressure-enthalpy diagram.[1]

The working fluid, or refrigerant, is evaporated at low pressure (P_l) in step 4 → 1 by heat exchange with the cold source (i.e. the system that is to be cooled). As indicated in the pressure-enthalpy diagram, the vapor may be super-heated in order to assure no liquid is fed to the succeeding compressor. The degree of superheating, ΔT_{sup}, is the temperature difference between the temperature at the outlet of the evaporator and the boiling point of the refrigerant at the given pressure.

The vapor is compressed to a higher pressure (P_h) in step 1 → 2 and cooled, condensed and subcooled by heat exchange with the hot sink (e.g. ambient air or cooling water) in step 2 → 3.

The degree of sub-cooling, ΔT_{sub}, is defined similarly as the degree of super-heating, namely the difference between the temperature of the liquid and the saturation temperature (boiling point). The liquid is expanded through a choke valve back to the low pressure (P_l) in order to provide a low temperature two-phase mixture that is fed to the evaporator.

The efficiency of a refrigeration cycle is measured by a coefficient of performance (COP) defined in Equation 2.1.

$$COP = \frac{|Q_C|}{W_s} \tag{2.1}$$

Q_C represents the heat removed from the system by the refrigerant (h_1 - h_4), while W_s represents the compressor work (h_2 - h_1). The COP for a refrigeration cycle is restricted by the Carnot efficiency as shown in Equations 2.2, 2.3 and 2.4.

$$\text{For a reversible Carnot process: } \frac{Q_H}{T_H} = \frac{Q_C}{T_C} \tag{2.2}$$

$$\text{Energy balance of the machine: } W_s = Q_H - Q_C = \left(\frac{T_H}{T_C} - 1\right) Q_C \tag{2.3}$$

$$\text{Substituted into Equation 2.1: } COP_{Carnot} = \frac{T_C}{T_H - T_C} \tag{2.4}$$

Thus, a refrigeration process is favoured by a small difference between T_H and T_C.

2.2 Natural gas liquefaction

LNG processes

The process of liquefying natural gas is basically the same process as described above. In LNG production, different refrigerant fluids are used to cool and condense the natural gas to approximately -162 ° C. Sea water is used to cool the compressed refrigerant streams which are subsequently expanded to provide cooling to the natural gas. The processes are extremely energy demanding, and involve large compressors. Therefore, large savings could be made by just slightly improving operating conditions.

Several different process designs have been developed, ranging significantly in complexity and capacity. The simplest processes involve a single loop of refrigeration (i.e. the PRICO process), while the more complex processes consist of multiple cascaded refrigeration circuits with different refrigerants (e.g. the Statoil-Linde MFC process). To illustrate the difference in terms of process design, simple flowsheets of the PRICO process and the Mixed Fluid Cascade (MFC) process are shown in Figures 2.2 and 2.3, respectively.

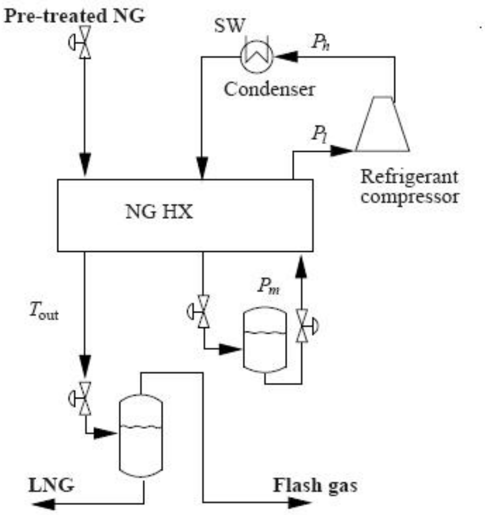


Figure 2.2: Simple flowsheet of the PRICO process.[1]

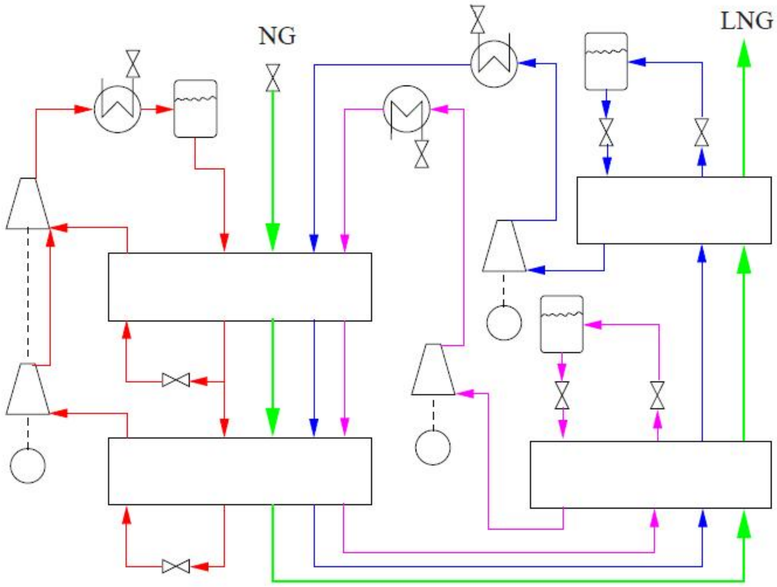


Figure 2.3: Simple flowsheet of the mixed fluid cascade (MFC) process.[1]

The simple PRICO process has a lower thermodynamic efficiency than the larger, more complex processes, and is therefore preferred for smaller plants (e.g. peak shaving plants) with production rates up to 2 MTPA (million tons per annum).

LNG processes may be operated with pure or mixed refrigerants, though mixed refrigerants have the advantage of providing a much closer-fitting cooling curve in heat exchange with the natural gas. This is shown in Figure 2.4. A pure refrigerant gives a large temperature difference in the warm end of the heat exchanger and therefore a low COP. It is necessary to use multiple cascades of pure refrigerant cycles to obtain results to be comparable to those of mixed refrigerants.

The propane pre-cooled mixed refrigerant (C3MR) process involves a single mixed refrigerant cycle. However, as it's name suggests, the process also consists of a propane pre-cooling cycle. This single component refrigerant eases the duty of the mixed refrigerant by cooling the natural

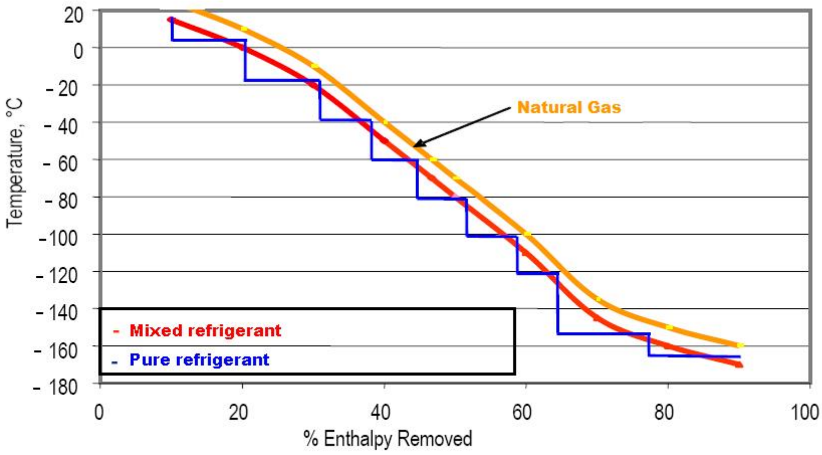


Figure 2.4: Cooling curves of pure- and mixed refrigerants vs. natural gas

gas and the refrigerant itself before the main cryogenic heat exchanger. The C3MR process will be explained further in Section 3.1.

The LNG Value Chain

Significant natural gas reserves are present in areas where there is no market, or where the quantity of the natural gas resources greatly exceed the local demand. For some areas, such as northern parts of Russia and North America, large gas pipeline infrastructure has been developed to transport the product from the remote source to the market. The development of such infrastructure calls for large capital costs, and is not always practical to implement (e.g. transport across oceans). The most exonomic way of transporting natural gas over large distances is in it's liquid form, LNG, by specially designed tank ships. See Figure 2.5.

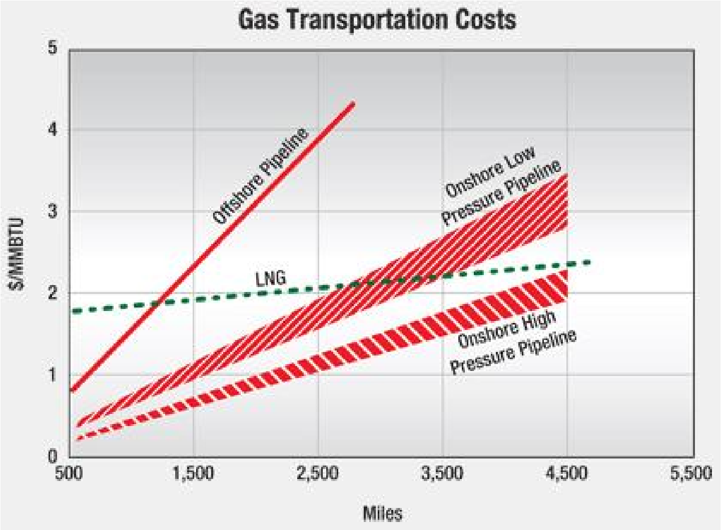


Figure 2.5: Comparison of the cost of transporting gas through pipeline as opposed to using LNG.[2]

Obviously, the first steps of the LNG value chain are exploration and production. Natural gas is found either as associated gas in oil fields or non-associated gas isolated in natural gas fields. The gas is transported to the LNG production plant on land by pipeline where it is pre-treated before the liquefaction process.

An interesting concept under development is known as *Floating* LNG plants (FLNG). Similar to Floating Production, Storage and Offloading (FPSO) units used in oil production, these FLNG plants are ships with a LNG production plant on deck. The use of FLNG plants will eliminate the transport of natural gas from well to plant (and vice versa the transport of CO_2 back to the well for storage), and will render it possible to exploit natural gas fields far out to sea. However, the world has yet to see the first FLNG unit in production.

Following the liquefaction process, the LNG is shipped to the market and stored in specially designed tanks. However, storage of LNG requires significant amounts of energy to keep the low temperature, so some amounts of the product is lost over time. When needed, the LNG is re-gasified at a terminal connected to the local infrastructure and distributed to the customers.

2.3 Introduction to numerical optimization

Optimization problems are seen in various applications, such as stock portfolios, chemical processes and transportation logistics. Optimization is also present in various levels of typical industry companies, from management to design to operation. The purpose of any optimization is to find the values of the variables corresponding to the best possible value of a given objective function. An optimization problem function can be linear or non-linear, and may be confined by various constraints.

A general optimization problem can be defined as follows:

$$\begin{aligned} \text{Minimize (or maximize):}\ & J = f(x) \\ \text{Subject to:}\ & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{2.5}$$

In Equation 2.6, J represents the objective function, which is a function of variable(s) x. The optimization problem may be subject to inequality constraints *g(x)* and equality constraints *h(x)*.

Different optimization methods have been developed in order to solve problems such as above. In the case where both objective function and constraints are linear functions of the variables, the optimization becomes a linear programming problem. If either objective function or constraints are non-linear functions of the variables, the problem is non-linear and more sophisticated methods are required to solve the problem.

One of the most popular and robust methods for non-linear optimization is the sequential quadratic programming (SQP) algorithm. The SQP algorithm handles both equality and inequality constraints, and is reduced to Newton's method for finding a point where the gradient of the objective is zero if the problem is unconstrained. The method constructs and solves a local model of the optimization problem and yields a step towards the solution of the original problem. The SQP algorithm uses a quadratic model for the objective function and linear models for the constraints. This is called a *quadratic program* (QP). The quadratic programs are solved sequentially, by minimizing the Lagrangian function with the linear approximation of the constraints in order to reach the optimum for the problem.[9] The optimum is defined by the Karush-Kuhn-Tucker conditions, a generalization of the method of Lagrangian multipliers to inequality constraints.[10]

The Karush-Kuhn-Tucker conditions are analogous to the condition that the gradient of the objective function must be zero at optimum, modified to take constraints into account. The conditions are based on the method of Lagrange multipliers, with the inclusion of inequality constraints rather than being restricted to equality constraints. The Lagrange function for a constrained optimization problem is presented in Equation 2.6.

$$L(x,\lambda) = f(x) + \sum \lambda_{g,i} g_i(x) + \sum \lambda_{h,i} h_i(x) \tag{2.6}$$

The vector λ is the concatenation of vectors λ_g and λ_g, and is the vector of Lagrange multipliers. The KKT conditions are presented in Equations 2.7 through 2.11:

$$\nabla_x L(x,\lambda) = 0 \tag{2.7}$$

$$\lambda_{g,i} g_i(x) = 0 \; \forall \; i \tag{2.8}$$

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

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

$$\lambda_{g,i} \geq 0 \tag{2.11}$$

Equation 2.7 represents the condition of a zero gradient of the Lagrangian function, while Equation 2.8 represents the complementary slackness. Equations 2.9 and 2.11 require that the inequalities and equalities constraints are met, while Equation 2.11 requires that the Lagrangian multipliers associated with the inequality constraint functions are positive.[10]

Another algorithm for solving non-linear optimization problems is the BOX method. The BOX method is a sequential search method that does not require any derivatives. For an optimization problem involving *n* variables, the BOX method creates a *n+1* complex around the center of the feasible region and calculates the objective function at each point. The point with the highest function value is replaced by it's opposite by extrapolating through the center of the complex. If the new reduces the value of the objective function, a new extrapolation is performed and the algorithm repeated until all points result in higher values of the objective function. It is clear that the BOX method is very simple and robust, but requires a large amount of iterations to converge. The BOX method does not handle equality constraints.[11]

3 Modelling

This chapter will introduce the C3MR process in more detail, explain estimations and approximations of process parameters and explain how the simulation model was built. It is important to note that the design of the process was not optimized, as the scope of the project was to optimize *operation* of a LNG plant. Thus, optimal design is not necessary and is neither always the case in real life. Equipment costs are significant, which in turn leads to the use of existing or cheaper equipment than what would be optimal for the process.

3.1 Process Description

The propane pre-cooled mixed refrigerant (C3MR) process is the dominant process for liquefaction of natural gas. It is developed by Air Products and Chemicals Inc. and was first run in 1972 at Shell's LNG plant in the Sultanate of Brunei. Several enhancements and extensions have been made to the process to increase capacity and/or efficiency. These will receive further mention later in this section. However, the main process remains the same, and is the focus of this report. The C3MR process is of medium complexity relative to the PRICO process and the MFC process showed in Section 2.2. A simplified flowsheet is shown in Figure 3.1. As implied by the name of the process, the primary step is cooling by propane. The natural gas is pre-cooled to approximately -36 °C by the propane cycle before it is passed through the main cryogenic heat exchanger (MCHE) where it is liquefied and sub-cooled to approximately -157 °C by the mixed refrigerant cycle. The natural gas fed to the process is usually at pressure around 40 bar, so the cooling down to the LNG product specification of approximately -162 is obtained by a isenthalpic expansion through a valve.

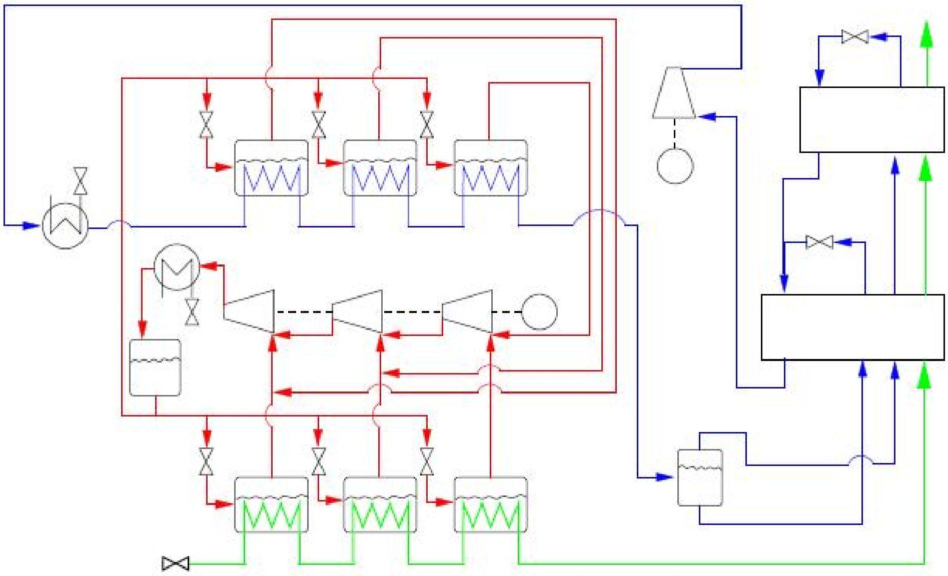
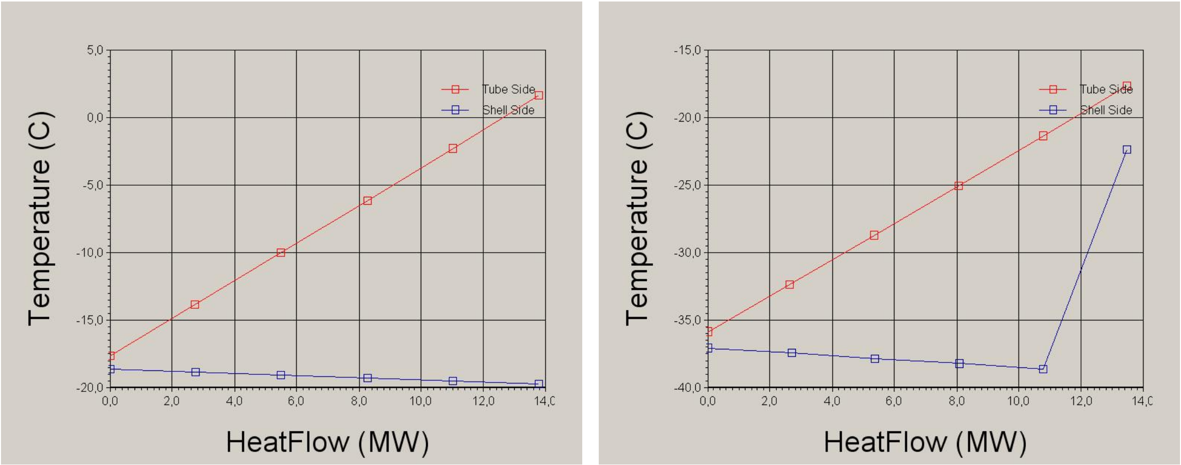


Figure 3.1: Simple C3MR Flowsheet. The propane (C3) cycle is shown in red, the mixed refrigerant (MR) cycle in blue, and the natural gas (NG) in green.[1]

The propane cycle is also used to pre-cool the mixed refrigerant. Propane is compressed to a high enough pressure in order to be condensed by cooling water. In other words, the pressure must be high enough for propane to be in liquid phase at the temperature achieved by the cooling. The liquid propane stream is let down in pressure and vaporized by heat exchange with natural gas and mixed refrigerant. An example of a temperature profile for a heat exchanger with boiling propane providing cooling to natural gas is presented in Figure 3.2a. The pressure let-down and heat exchange is performed in three stages, where the propane vapor is sent back to compression after each stage. The final heat exchangers in the propane cycle must super-heat the propane in order to avoid liquid being fed to the first compressor. A temperature profile for a propane heat exchanger with super-heating is presented in Figure 3.2b. After pre-cooling, the mixed refrigerant is partially condensed and is sent to a high pressure separator prior to entering the main cryogenic heat exchanger (MCHE). The vapor and liquid MR streams pass through separate circuits in the MCHE and are cooled, liquefied and sub-cooled by internal heat exchange along with the natural gas. The two sub-cooled refrigerant streams are let down in pressure, reducing their temperature to provide necessary cooling to their respective areas of the MCHE. As indicated in Figure 3.1, the liquid refrigerant stream is taken out and expanded at a point other than that for the MR vapor. As the low pressure refrigerant streams flow down the MCHE, it is vaporized and super-heated by the cooling of natural gas (and the MR streams). The super-heated low pressure mixed refrigerant is then recompressed and cooled by water to complete the cycle. The result of this process is a high pressure natural gas stream of approximately -157 °C, which is let down to atmospheric pressure to give LNG at 1 atm and -162 °C.



(a) Without super-heating

(b) With super-heating

Figure 3.2: Temperature profile for cooling natural gas by propane vaporization with and without super-heating.

Since the birth of the C3MR process in 1972, several enhancements have been made in the main cryogenic heat exchanger and refrigerant compressors and drivers to increase efficiency and capacity. Air Products and Chemicals Inc. have also developed an extension to the C3MR process, known as the AP-X process. The AP-X is basically the C3MR with the addition of a nitrogen expander. The nitrogen expander takes care of the final sub-cooling of the LNG, allowing the MCHE a outlet temperature of approximately -115 °C rather than -157 °C, thus allowing larger throughput. However, the AP-X process is not considered in this report.

3.2 Modelling in UniSim design

A model of the C3MR process was built using UniSim Design simulation software by Honeywell. The simulation flowsheet is shown in Figure 3.3 and is also included in larger format in Appendix A. The workbook for the design case containing all of the process data is attached in Appendix B. The extraction of heavy components from the natural gas has not been considered in this report. This may be performed upstream or integrated in the liquefaction process.

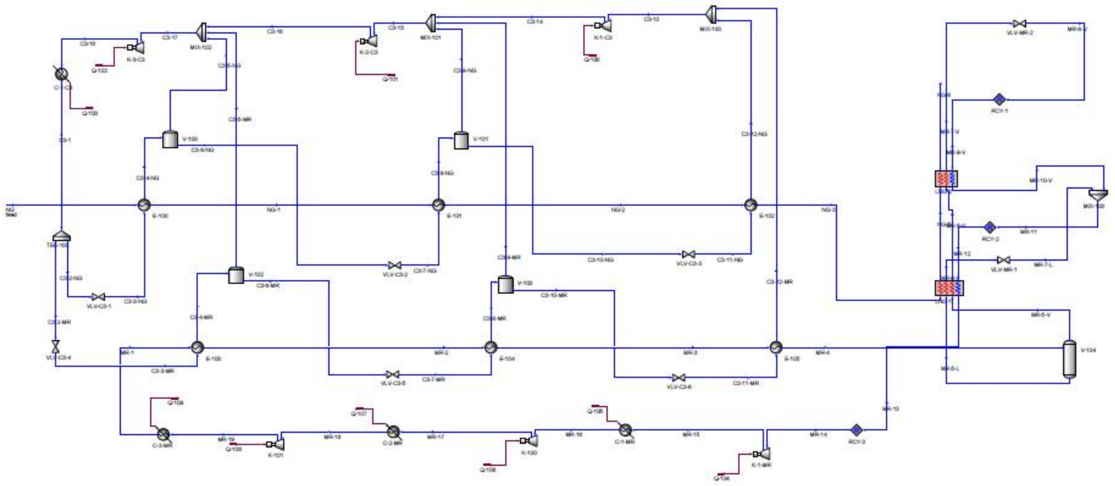


Figure 3.3: C3MR process flowsheet in UniSim Design.

The thermodynamic fluid package of Peng-Robinson was used for the simulation. The natural gas composition, the composition of the mixed refrigerant and other process parameters that were used as basis for the simulation model are given in Tables 3.1, 3.2 and 3.3, respectively.

Table 3.1: Natural gas composition

Component	Fraction
Methane (C1)	89.7 %
Ethane (C2)	5.5 %
Propane (C3)	1.8 %
n-Butane (n-C4)	0.1 %
Nitrogen (N2)	2.9 %

Table 3.2: Mixed refrigerant composition

Component	Fraction
Methane (C1)	45.0 %
Ethane (C2)	45.0 %
Propane (C3)	2.0 %
n-Butane (n-C4)	-
Nitrogen (N2)	8.0 %

The feed rate of natural gas in the simulation is 60 000 kmol/h, corresponding to a LNG production rate of approximately 8.4 MTPA (million tons per year). This is a large, but reasonable

Table 3.3: Process parameters

Parameter	Value
Natural gas	
NG inlet pressure	40 bar
NG inlet temperature	30 °C
NG feed rate	60 000 $\frac{kmol}{h}$
Propane (C3)	
Temperature after sea water cooling	30 °C
Mixed refrigerant (MR)	
Temperature after sea water cooling	30 °C
Heat exchangers in propane cycle	
ΔP tube side	0.5 bar
ΔP shell side	0.1 bar
MCHE (per part)	
ΔP hot stream	5 bar
ΔP cold stream	0.5 bar

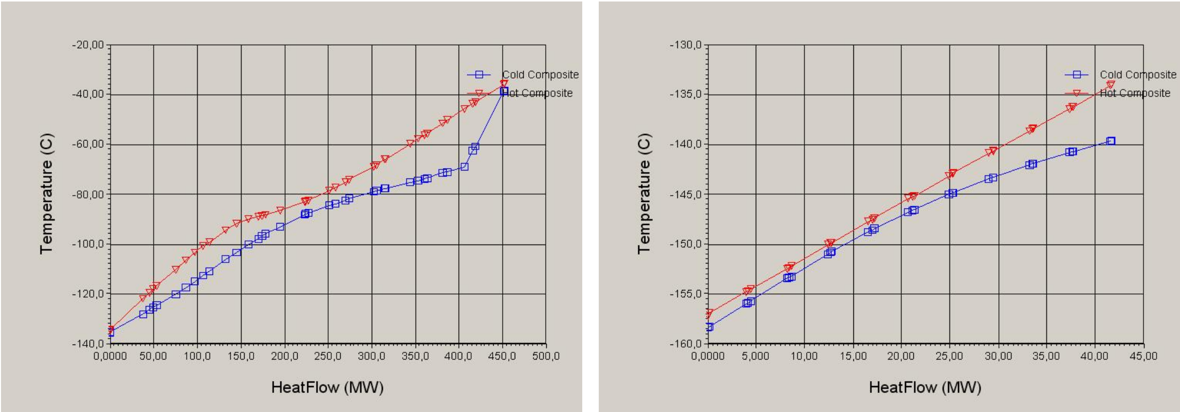
size for a C3MR process train. [8]

As seen in the design flowsheet in Figure 3.3, the flow of propane is split into two streams that provide pre-cooling to the natural gas and the refrigerant. A notable difference from the flowsheet presented in Figure 3.1 is that the UniSim model does not operate with splits before each of the propane heat exchangers. The simulation model involves only one splitter unit, but the remaining splits are taken care of flash tanks succeeding each heat exchanger, sending the vapor to recompression and the liquid phase to the next step in the precooling cycle.

The pressure assignment of the mixing units was set to the option *Equalize all.* This choice implies that the propane cycle has an equal let-down in pressure per valve for the natural gas side and the refrigerant side. In other words, it is only necessary to set one pressure value, and the mixer automatically sets the other pressures. This choice was made in order to simulate a real mixing process, where three pipes are connected in a pipe joint. The same pressure will apply to all the streams that are connected.

The heat exchangers in the pre-cooling cycle are modelled as shell-tube heat exchangers, while the main cryogenic heat exchanger (MCHE) is modelled as a combination of two LNG-type exchangers. The LNG exchangers allows for multiple streams, which is necessary for the MCHE. In the case of multiple streams, an interative approach is used to determine the solution that satisfies not only the energy balance, but also any constraints, such as temperature approach or UA.[11] The tolerance of the iterations is set by the user in UniSim Design. The tolerance was set as low as possible (10^{-4}) with respect to maintaining a sufficiently robust flowsheet design. In other words, the tolerance had to be high enough to allow slight changes in process parameters without rendering a flowsheet unable to converge. The tolerance of the heat exchangers turned out to cause serious problems for optimization using MATLAB. This will be presented in Secion 4.

All heat exchangers are modelled using a *Weighted* counter-current design model. The Weighted model is an excellent model to apply to non-linear heat curve problems such as the phase change of pure components in one or both heat exchanger sides. With the Weighted model, the heating curves are broken into intervals, and an energy balance is performed along each interval. A logarithmic mean temperature difference (LMTD) and UA are calculated for each interval and summed to calculate the overall exchanger UA.[11] All heat exchangers in the process were designed to have a minimum temperature approach (ΔT_{min}) greater than 0.5 °C. The last two heat exchangers in the propane cycle and the bottom part of the MCHE were designed to super-heat the propane and mixed refrigerant, respectively, to avoid liquid in compressor feeds. Examples of the temperature profiles of the propane heat exchangers can be seen in Figures 3.2a and 3.2b in Section 3.1. The temperature profile of the two parts of the MCHE are shown in Figures 3.4a and 3.4b.



(a) Bottom part

(b) Top part

Figure 3.4: Temperature profile showing the cold and hot composite curves for the two parts of the MCHE.

Three recycle units had to be added to selected streams in UniSim Design for the simulation software to converge. The software solves unit operations subsequently as an independent modular simulator, so the recycle units are necessary for the software to make an initial guess for the two parts of the MCHE as well as the first MR compressor.

The expansion of the sub-cooled natural gas to atmospheric pressure was omitted from the design. Instead, the sub-cooled pressurized natural gas stream was required to have a temperature of -157°C. This temperature is sufficient to give saturated liquid natural gas at approximately -162°C at atmospheric pressure.

Instead of using a single compressor, the recompression of mixed refrigerant was split over three compressors with intercooling. This is typical for existing designs.[8] The three compressor design restricts the compressor outlet temperature from being too high with respect to material considerations, as well as reducing the total compressor workload.

Internal heat exchange in the MCHE makes the design very sensitive to changes in process parameters. For a given set of parameters, the range for changing a single variable is very limited without causing problems for the convergence of the flowsheet. Section 4 of the report shows that this lack of robustness in the model caused serious problems for optimization procedures.

4 Optimization

This section introduces the ideas behind the degrees of freedom (DOF) of a process, and determines the number of DOF's available in the C3MR process. The determination of the objective function for optimization is discussed, as well as the operational constraints. Further, the attempts made at optimizing the operation of the process are described in detail, with both some theory and practical implementation. Optimization was attempted using both the optimizer included in UniSim Design as well as interfacing and optimizing with MATLAB.

4.1 Degrees of freedom (DOF) analysis

General

In general, a degree of freedom (DOF) is a single scalar number describing a micro-state of a system. The system is then completely described by all it's degrees of freedom. For a process design, the number of steady-state DOF's is the number of variables (parameters) that must be specified to completely define the process. The degrees of freedom can be calculated by subtracting the number of specified variables (equations) from the number of process variables, as shown in Equation 4.1.

$$N_{SS} = N_{var} - N_{SV} \tag{4.1}$$

N$_{var}$ represents the number of process variables, and N$_{SV}$ represents the number of specified variables (equations). However, counting equations is not a very efficient procedure. The steady-state degrees of freedom for a process may also be determined by counting the manipulated variables N$_{MV}$ (valves) and subtracting the variables with no steady-state effect and the process specifications as shown in Equation 4.2. N$_{0,SS}$ includes purely dynamic DOF's such as heat exchanger bypass streams and controlled variables without steady-state effect such as liquid levels in tanks.

$$N_{SS} = N_{MV} - N_{0,SS} \tag{4.2}$$

It is essential to identify the steady-state degrees of freedom because they represent the degrees of freedom available to process optimization.

C3MR process

The potential degrees of freedom for the C3MR process according to the tables of Skogestad and Jensen [1] are presented in Table 4.1. The specifications in the simulation model in UniSim Design are shown in Table 4.2. The composition of the mixed refrigerant involves N_C *- 1* degrees of freedom. The MR in the C3MR process consists of only four components, however, it is assumed that it may also contain n-Butane. Thus, 5 - 1 = 4 degrees of freedom are available related to the MR composition. The 'Equalize all' option on mixing units consumes *N - 1* specifications of the simulation model, with *N* being the number of streams being mixed. As expected, the number of potential degrees of freedom and the specifications in UniSim Design (with consideration of the mixers) add up to the same number.

Table 4.1: Potential degrees of freedom for process units in the C3MR process.

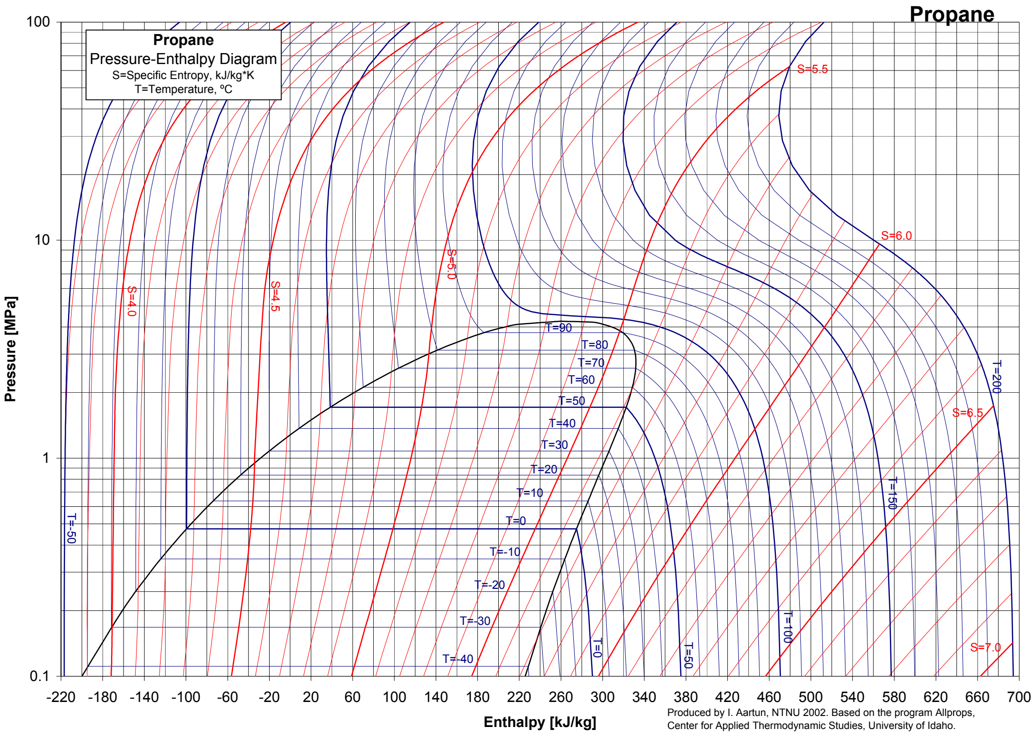
Process unit	**Potential DOF**
NG feed	1
C3 splitter	1
Compressors	6
Heat exchangers	15
Choke valves	8
Holdup in closed cycles	2
Composition of MR	4
Total	**37**

Table 4.2: Process specifications in UniSim Design.

Process unit	**Specification**
NG feed rate	1
C3 flows	2
C3 pressures	4
C3 temperature	1
MR flow	1
MR pressures	4
MR temperatures	3
MR composition	4
Heat exchangers	11
Mixers (Equalize all)	6
Total	**37**

Some assumptions were made in order to reduce the amount of degrees of freedom for optimization. The optimization was based on a given feed rate, so that was eliminated as a DOF. The composition of mixed refrigerant was specified as described in Section 3.2, and is therefore not a degree of freedom in the optimization of the process. It was assumed that the sea water coolers had capacity to cool the process streams to 30 °C. In other words, the flow of cooling water was used to control the temperature of the process streams at 30 °C. The degrees of freedom related to heat exchangers (bypass) have no steady-state effect and are therefore not counted as actual degrees of freedom. This was achieved in UniSim Design by keeping the UA values constant for all heat exchangers. Additionally, six degrees of freedom were consumed by the 'Equalize all' option of the mixing units.

The results of the DOF-analysis described above give 11 degrees of freedom available to the C3MR process simulation model in UniSim Design. Further analysis and thermodynamic reasoning implies that one of these DOF's should be specified at the minimum value allowed for process feasibility. The highest pressure of the propane cycle needs to be higher than the saturation pressure of propane at 30 °C in order to ensure a two-phase mixture after expansion. However, increasing the pressure beyond this point will not benefit the operation in any way, as the isotherm is practically vertical in the liquid phase for pressures within reasonable range. A pressure-enthalpy digram for pure propane is shown in Figure 4.1 illustrating this fact. On the contrary, an increase in pressure will require more work for the compressor(s). The diagram in Figure 4.1 shows that the saturation pressure at 30 °C is approximately 1.1 MPa. Calculated in UniSim Design, the exact saturation pressure at 30 °C was retrieved to be 1091 kPa.



Produced by I. Aartun, NTNU 2002. Based on the program Allprops, Center for Applied Thermodynamic Studies, University of Idaho.

Figure 4.1: Pressure-enthalpy diagram of pure Propane.

A typical design of a condenser used in refrigeration cycles is shown in Figure 4.2. The vapor refrigerant is cooled in a tank with a cooling coil, so the condensed liquid is drained to the bottom and therefore is not sub-cooled. As the temperature after cooling is specified to 30 °C and the liquid is saturated, the pressure is given as the saturation pressure. It follows that the pressure is not a degree of freedom when using this type of condenser, strengthening the suggestions in the previous paragraph. As a result, the simulation model has ten degrees of freedom available.

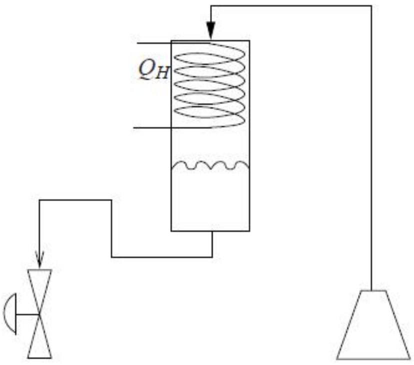


Figure 4.2: Common design of condenser with saturation at outlet giving no sub-cooling.[1]

The temperature of the sub-cooled natural gas is required to be lower than -157 °C. The temperature was not specified in the simulation model, but the process design was adjusted to meet this constraint. For the optimization of the process, the temperature criteria was added as an inequality constraint. This is discussed in the next section.

4.2 Objective function and constraints

In order to optimize operation of the C3MR process, it is necessary to minimize the operating costs. In other words, the operating costs represent the objective function for this optimization problem. The derivation of the operating costs is shown in Equation 4.3.

$$\min\ J = f(x) = \sum (P_{W_s} W_s) + \sum (P_{SW} Q_C) + P_{NG}\dot{m}_{NG} - P_{LNG}\dot{m}_{LNG} \tag{4.3}$$

By assuming that the cost of cooling by sea water is neglected and that the feed and production rate are equal and constant, the objective function can be simplified to Equation 4.4. In words, the operation cost of the process is solely a function of the compressor work, which is again a function of the process variables.

$$\min\ J = f(x) = \sum W_s \tag{4.4}$$

The optimization problem also involves a series of constraints in order to give a feasible operation of the process. First of all, the temperature of the sub-cooled LNG must be less than -157 °C. The temperature in the simulation design was slightly above, at -156.98 °C. The optimization requires a starting point that meets constraints, so the sub-cooled LNG temperature constraint was relaxed to -156.9 °C. The minimal difference in temperature should have no significance with regards to the optimization of the process. The heat exchangers can are constrained by a minimum temperature approach requirement (ΔT_{min}) in order to give a reasonable exchanger area. The constraint value for ΔT_{min} was roughly estimated based on realistic scenarios. Also, the outlet propane streams of the last heat exchangers in the pre-cooling cycle must be superheated in order to avoid liquid being fed to the compressor. A minimum degree of super-heating ($\Delta \mathrm{T}_{sup}$) of 10 °C was set as a constraint. This constraint also applies for the mixed refrigerant exiting the warm side of the main cryogenic heat exchanger. The optimization constraints are presented in Table 4.3.

Table 4.3: Optimization constraints.

Property	**Constraint**
T_{LNG}	$<$ -156.9 °C
$\Delta\mathrm{T}_{min}$ (all HEX's)	$>$ 0.5 °C
$\Delta\mathrm{T}_{sup}$ (two C3 streams and MR)	$>$ 10 °C

4.3 Optimization in UniSim Design

The steady state optimizer included in UniSim Design offers several optimization methods. The *Mixed* method is a hybrid method that attempts to take advantage of the global convergence characteristics of the BOX method and the efficiency of the SQP method. The BOX and SQP methods are described in Section 2.3. The BOX method is used initially with a very loose tolerance to find an optimal region, whilst the SQP is subsequently used to pinpoint the optimal solution.[11] The stand-alone BOX and SQP methods are also available, as is Newton's method and other less known algorithms. The optimization attempts in this report were made using both the Mixed method and the SQP algorithm.

The process parameters were adjusted and tuned manually to find a near-optimal solution in order to assist the optimizer. Internal heat exchange in the MCHE makes the design very sensitive to changes in process variables that affect the performance of the MCHE. In order to maintain a converged flowsheet, the range that variables can be changed is quite limited. Variables were changed one by one consecutively in order to determine the approximate ranges for the variables while maintaining a converged flowsheet. In the case of combinations of multiple deviated variables causing difficulties for the flowsheet, the respective variable ranges were readjusted accordingly.

The optimizer in UniSim Design operates with penalty values for constraints. In other words, the optimizer may break some constraints provided that the value of the objective function will improve. The penalty value is a way the user may suppress this from happening. The constraint function is multiplied with the penalty value in the optimization calculations. [11]

10 variables

Optimization was attempted using the 10 degrees of freedom determined in Section 4.1 as optimization variables and the constraints showed earlier in Table 4.3. The results varied attempt after attempt, but did not give a reasonable solution to the optimization problem. Often, the optimizer failed to meet the constraints: Either the LNG temperature was too high, temperature crosses occurred in the heat exchangers, or super-heating was not obtained where needed and liquid was fed to the compressors.

The penalty values for the constraints were raised to accordingly, even to heights excessively over what should be necessary for the value of the objective function. However, the optimizer either failed at finding an optimal solution or gave the initial conditions as the optimal point of operation. The latter was tested by changing the initial conditions and re-running the optimizer with the same parameters. The result was that the new initial conditions were optimal. The tolerance of the optimizer was set according to the deviation in the objective function, and was even varied from unreasonably loose to excessively tight without any improvements. Clearly, the optimizer was not functioning properly, perhaps due to tolerance issues in the flowsheet. These issues will be discussed in the summary of this section.

14 variables

In the previous optimization attempt, the heat exchangers in the UniSim Design simulation model had specified UA values. The outlet temperatures of each heat exchanger were calculated based on the inlet temperatures and the UA value using Equation 4.5. The process is iterative, as the software has to make an initial guess of the outlet temperatures before calculating the logarithmic mean temperature difference (ΔT_{lm}) and comparing the values of heat transferred (Q).

$$Q = UA\Delta T_{lm} \tag{4.5}$$

On the contrary, if all except one outlet temperatures are provided as specifications, the calculation of the heat exchanger equations would be more efficient. The transferred heat could be calculated relatively easily, retrieving the last outlet temperature and thereafter calculating the UA values. It was suspected that such an change in the process simulation would render a more robust flowsheet in terms of deviations of process variables. In other words, allowing a greater range of variables without resulting in an non-converged flowsheet.

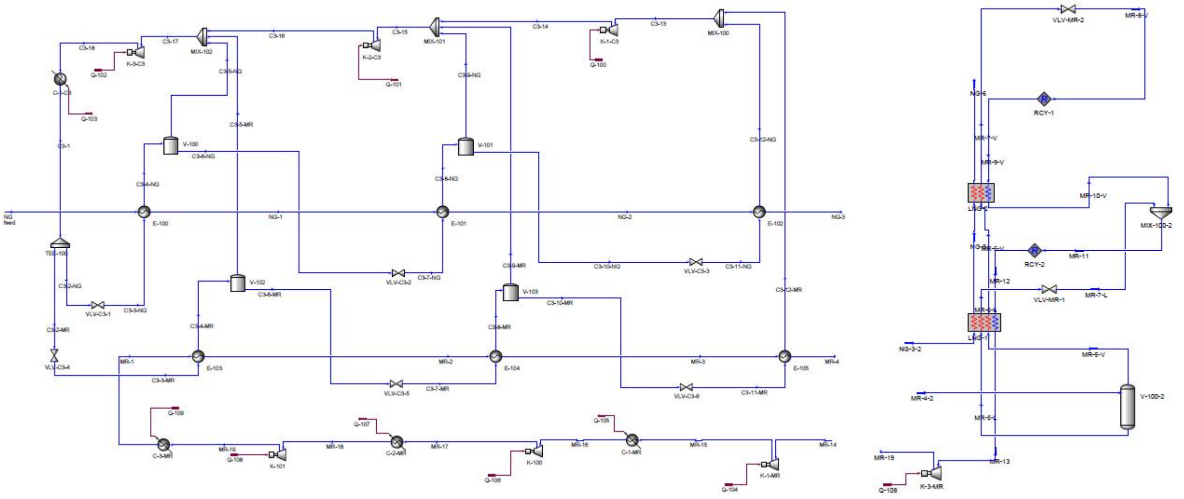
The simulation model was modified to include four additional process variables: Three intermediate temperatures of the MCHE, as well as the outlet temperature of MR at the cold end of the MCHE. Five specifications of UA values were removed from the MCHE parts (see Section 3.2), while the temperature of the sub-cooled LNG was specified at -157 °C. This way, the LNG was guaranteed to meet it's temperature specification. Since the UA values of the heat exchangers should not be altered from the original design values, these were added as equality constraints to the optimization problem. The UA values of the heat exchangers in the pre-cooling cycle were left as specifications, as they had evidently not caused any problems for the optimization. Anyhow, setting sufficient ranges for the resulting outlet temperature variables would not be possible without causing the flowsheet to crash during optimization (the ranges would have to be dynamic with respect to the value of the preceding heat exchanger).

The simulation model definitely proved to be more robust, and the MCHE required significantly shorter time to converge at a steady state solution after altering process variables. However, the optimization was still not producing any reasonable results. The same problems were encountered as in the case with 10 variables. The constraints on the UA values of the MCHE were severely broken, as were the original constraints on minimum approaches and super-heating. The results were irregular, either returning an error message or producing an optimal solution that could be proven incorrect by simply changing a variable in a logical direction.

Split Flowsheet

The flowsheet was divided into two parts, as shown in Figures 4.3a and 4.3b, to attempt to optimize the pre-cooling cycle and the MCHE separately. The two parts of the process are dependent of each other, but assumptions could be made allowing for independent optimization of the MCHE. The two flowsheets could subsequently be combined and the whole process could be optimized with respect to the remaining variables in the pre-cooling part.

It was assumed that the optimal operation of the entire process was achieved by pre-cooling the natural gas and the mixed refrigerant to -36 °C. This assumption was based on process descriptions in literature [8], as well as the saturation point of propane at around 120-130 kPa pressure (an estimate of the lowest pressure in the propane cycle) while considering the minimum temperature approach in the heat exchanger.



(a) Pre-cooling and recompression (b) MCHE

Figure 4.3: The C3MR process divided into two separate flowsheets.

Five degrees of freedom were available for the optimization of the MCHE: The flow of MR, the pressure drop in one MR valve, and the three pressures of the MR stream after the recompression steps. The UA values of the MCHE were specified, so there were totally 10 manipulated variables in the entire process. Optimization was attempted, but the results were similar to those obtained in the previous attempts. Without an optimized MCHE, there was no point in optimizing the pre-cooling cycle. This would imply optimizing with respect to a non-optimal flow of MR with a non-optimal pressure, and would not bring us any closer to the solution.

However, significant work was put into analyzing various case studies on how the different variables affected eachother, the temperature of sub-cooled LNG, and the compressor workload. Based on these simple studies, the variables were altered to give a lower value of the objective function. In other words, the process was manually optimized to some extent, though no theoretical proof can be provided that the operating conditions were close to optimum.

Summary

The optimization attempts using the built-in UniSim Design optimizer were not successful. It seems that the optimizer is not capable of meeting the constraints of this optimization problem, even with excessive manipulation of the penalty values. Little documentation is available about the optimizer and it provides limited room for user input and control. In addition, the output of the optimizer provides limited information about the optimization routine, thus making it difficult to troubleshoot.

Since the tolerance of the heat exchangers (mainly the MCHE) were set relatively loose (10^{-4}) in order for the flowsheet to converge, the objective function calculations were subject to uncertainty. It is suspected that the objective function is quite flat near the optimum, thus the optimization is subject to the 'noise' in the objective function calculations. This is problematic since the SQP algorithm uses gradients to determine the next step in the optimization routine. Due to the 'noise' in the values of the objective function, small steps may provide a inaccurate gradient and lead the optimizer to a incorrect solution. This scenario is illustrated in Figure 4.4.

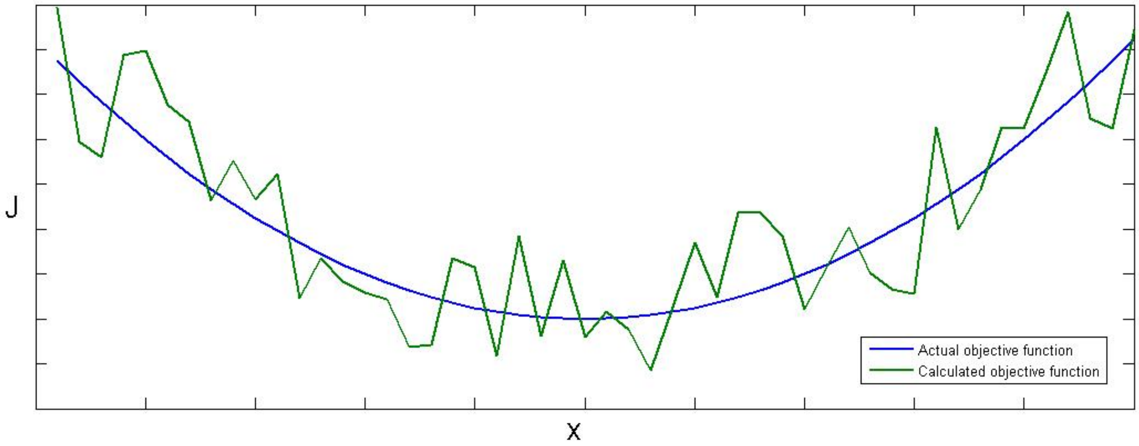


Figure 4.4: Inaccuracy in objective function calculations affecting the optimization. Taking small steps will give incorrect gradients.

The UniSim Design optimizer did not give reasonable results and does not offer the level of customization needed for this optimization problem. It was therefore determined to be unsuitable and optimization must be attempted using separate software.

4.4 Optimization in MATLAB

Due to the problems encountered using the built-in optimizer of UniSim Design, it was decided to attempt optimization using MATLAB. The MATLAB function *fmincon* is a solver for non-linear constrained minimization problems. The *active-set* algorithm of the *fmincon* function uses a sequential quadratic programming (SQP) method, solving a quadratic program (QP) at each iteration and updating an estimate of the Hessian of the Lagrangian.[12] MATLAB offers a wide range of user options to customize the optimization routine for the problem at hand, as well as offering a detailed output display clearly showing where and why the optimization failed. Most importantly, previous experience from optimization with *fmincon* in MATLAB had not been subject to difficulties meeting constraints.

Interfacing MATLAB and UniSim

In order to optimize the process using MATLAB, the program must be able to communicate with the simulation in UniSim Design. The process model in UniSim Design was still used for simulating the process, but the optimization routines were lifted out to MATLAB. The two programs can be interfaced from MATLAB by creating an ActiveX/COM automation server running UniSim Design. Further, MATLAB connects to the server as a client and is able to access the internal structure of the simulation software. In other words, MATLAB can withdraw any information from the flowsheet and similarly assign new data and run the solver. Using MATLAB to control UniSim Design in this way presents a variety of new applications, as UniSim Design procedures may be implemented into MATLAB scripts and functions and do not require manual input. The scripts and functions used for interfacing MATLAB and UniSim Design are attached in Appendix C.

The MATLAB function *fmincon* determines the values of the variables that minimize the objective function with respect to equality and inequality constraints. The syntax of the function is shown in Equation 4.6.

$$x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon) \tag{4.6}$$

The function *fun* is the objective function, which is calculated at each iteration by sending the variables x to UniSim Design and running the simulation. *x0* represents the initial estimate of x, while $A \cdot x \leq b$ and $Aeq \cdot x = beq$ represent linear inequalities and equalities. *lb* and *ub* represent the lower and upper bounds of the variables, respectively. The function *nonlcon* calculates the non-linear constraints, by calling the UniSim Design solver yet again. For each iteration, MATLAB calls UniSim Design twice in order to calculate the objective function and the constraints.

Optimization

Optimization was carried out with both 10 and 14 variables, similarly as when using the built-in optimizer in UniSim Design. The interface proved to be successful, as the simulation was running and the values of variables and the objective function were presented in MATLAB. However, familiar problems arose regarding the flowsheet not being able to converge and thus causing an optimization failure in MATLAB. The variable bounds were adjusted accordingly to bypass the convergence problems, but optimization was yet not successful. The optimizer kept running as if taking extremely small steps in search of the optimum, but returned without reasonable results. The initial estimate was often returned as a feasible optimum, as were results with broken constraints or no feasible solution at all.

Repetive attempts were made altering the objective function tolerance as well as the minimum change in variables for finite differences gradients. As described in Section 4.3, the objective function is subject to 'noise' or inaccuracy from loose tolerances in the simulation in UniSim Design. The SQP algorithm calculates the gradients of the objective function with respect to the different variables in order to determine the next step in the optimization routine. In case the optimizer takes too small changes in variables to calculate the gradients, the resulting gradients may be extremely inaccurate compared to those of the actual objective function. The inaccurate gradients will in turn cause the optimizer to settle at a wrong solution, or in some cases, not move at all. See Figure 4.4 in Section 4.3 for a visualization of the theory.

Due to the problems explained above, the tolerances in the heat exchangers were tightened to the verge of barely allowing the flowsheet to converge in a reasonable amount of iterations, and the minimum change in variables for calculating gradients (DiffMinChange) was raised to 10^{-4}. The exact function call can be seen in Appendix C. The optimization routine was successful at converging at a solution, and the results are shown in Table 4.4 along with the design values (initial estimate). The detailed results of the optimization are presented in Appendix D.

Table 4.4: Results of optimization in MATLAB. Note: Pressure indeces are numbered from low to high pressure.

Objective function	**Optimized value**	**Initial estimate**	**Unit**
$J = \sum W_s$	326 664	328 482	kW
Manipulated variable	**Optimized value**	**Initial estimate**	**Unit**
C3 flow to NG pre-cooling	13 540	13 800	kmol/h
C3 flow to MR pre-cooling	68 550	69 000	kmol/h
P_1 Propane	126.3	126.5	kPa
P_2 Propane	256.8	253.1	kPa
P_3 Propane	483.2	482.2	kPa
MR flow	117 100	118 000	kmol/h
P_1 MR	540.2	540.0	kPa
P_2 MR	2296.2	2295	kPa
P_3 MR	3383.0	3380.0	kPa
P_4 MR	4800.1	4800.0	kPa

Table 4.5: Details about the optimization.

Number of iterations	5
Function Evaluations	99
Maximum constraint violation	$-6.657 \cdot 10^{-4}$
First order optimality measure	3097.6
Active inequalities	-

Table 4.4 shows that the design (with manual optimization) was very close to optimum calculated by MATLAB. Some of the details regarding the optimization are shown in Table 4.5. The optimization ran for five iterations, which is a low but reasonable number considering how small changes have been made to the variables and how little the objective function has changed. The first-order optimality measure for constrained optimization is based on the KKT conditions that were presented in Section 2.3. The optimality measure is the maximum of the infinity norm of Lagrangian function and the infinity norm of the inequality constraint function. These are presented in Equations 4.7 and 4.8.

$$\parallel \nabla_x L(x,\lambda) \parallel = \parallel \nabla f(x) + \sum \lambda_{g,i} \nabla g_i(x) + \sum \lambda_{h,i} \nabla h_i(x) \parallel \tag{4.7}$$

$$\parallel \lambda_g \vec{g}(x) \parallel \tag{4.8}$$

The equality constraints $h(x)$ are neglected in our case, as none are present. At optimum, the optimality measure should by definition be equal to zero. This is not the case for the optimization in MATLAB, clearly telling us that something is not correct. However, the optimality measure is based on the gradients of the objective function and the constraints, which likely are extremely inaccurate due to inaccuracy in the flowsheet calculations. Thus, all results must be given reasonable doubt.

The optimum constraint values are presented in Table 4.6. The inequality constraint on the subcooled LNG temperature of -156.9 °C is regarded to be active since the deviation is only 0.01 °C. It is unlikely that excess cooling would be beneficial, and the difference may be a result of loose optimization tolerance in MATLAB. In any case, based on process reasoning and the optimization results, the sub-cooled LNG temperature should be considered an active constraint.

Table 4.6: Constraint values at optimum.

Constraint	**Value**
LNG outlet temperature	-156.91 °C
ΔT_{min} HEX E-100	1.00 °C
ΔT_{min} HEX E-101	0.98 °C
ΔT_{min} HEX E-102	1.23 °C
ΔT_{min} HEX E-103	1.03 °C
ΔT_{min} HEX E-104	1.00 °C
ΔT_{min} HEX E-105	1.35 °C
ΔT_{min} MCHE bottom part	0.98 °C
ΔT_{min} MCHE top part	0.93 °C
ΔT_{sup} of C3 in HEX E-102	16.69 °C
ΔT_{sup} of C3 in HEX E-105	19.34 °C
ΔT_{sup} of MR in MCHE	29.72 °C

The minimum approach temperatures in the heat exchangers are reasonable, as are the degrees of super-heating of propane and mixed refrigerant.

Scaling

Scaling of the objective function, variables and constraints was considered to be an issue for the optimization problems. However, the SQP algorithm with BFGS update of the Hessian is considered scaling invariant.[13] Scaling should not have any significant effect on the number of iterations needed, and was therefore not considered in more detail. In retrospect, more time and research should perhaps be devoted to this area, as the optimization turned out be problematic. Scaling of the objective function and variables would result in a smaller condition number for the Hessian matrix, possibly easing the work of the optimizer.[13] Yet, scaling is not considered to be the primary cause of the problems that arose during optimization.

Summary

Optimization using the function textitfmincon in MATLAB interfaced with the UniSim Design flowsheet proved to be difficult, but yet more user-friendly than optimizing directly in UniSim Design. The optimization is obviously critically dependant on whether the flowsheet is able to converge or not. A non-converged flowsheet led to a failed optimization as the objective function nor constraints could be calculated. By restricting the tolerances in the simulation flowsheet as well as manipulating the optimization options, some optimization results were obtained and were presented in the preceding section. However, caution must be exercised when analyzing the results. While the optimization was successful at reducing the objective function and meeting the constraints, the obtained results should be subject to criticism. The first-order optimality measure gives a clear indication that the optimization is not entirely correct, though being relatively small compared to the magnitude of the objective function.

In general, optimization of the C3MR process simulation model proved to be difficult to perform. It is suspected that the difficulties arise from the robustness of the simulation model in UniSim Design. Nevertheless, the results have been presented and discussed.

4.5 Self-optimizing control

The results of the optimization of the C3MR process with 10 degrees of freedom yielded a single active constraint, the sub-cooled LNG temperature. Following the procedures for plantwide control, each active constraints must be controlled by one DOF, leaving nine DOF's for self-optimizing control. The self-optimizing control structure should preferrably be determined using the maximum scaled gain (minimum singular value) procedure.[3]

For each of the nine remaining DOF's, the gains of selected control variables (CV) should be calculated by introducing small perturbations to the manipulated variable (MV). Following, the process should be reoptimized for various disturbances that may be relevant, such as deviations in the temperature of cooling water or composition of the natural gas. The scaled gains may then be calculated by scaling the nominal gain values with the span of each control variable. The span represents the variation of each control variable for expected disturbances and implementation errors. Following the maximum scaled gain rule, the control variable with the largest scaled gain from a given manipulated variable should be controlled at it's nominal optimum point by the respective MV.[3] Though it would be ideal to determine the self-optimizing control structure using the maximum scaled gain rules, the procedure requires reoptimization after introducing realistic disturbances. Since the optimization routine presented earlier in this section proved to be difficult and inaccurate, the maximum scaled gain method was ruled out as an efficient alternative.

Instead, the self-optimizing control structure may be determined using the brute-force method. The brute-force method consists of systematically checking the objective function value for every candidate solution at each disturbance. The result will be similar to the graph presented in Figure 4.5. For the given manipulated variable, the lower graph will provide a smaller loss in comparison with the reoptimized process, and be the better control variable to keep at a constant setpoint.

According to the PhD thesis of Alstad [14], the self-optimizing control structure is not affected by the point of operation. Even if operated at a non-optimal setpoint, the average loss will be the same as for the optimal case, and the self-optimizing structure will uphold. This is fortunate in our case, as it is likely that the optimization has not yielded the true nominal optimum for the process, but the self-optimizing structure can yet be determined.

The C3MR process has nine manipulated variables available for which a self-optimizing control variables must be determined. While depending on the amount of possible control variables, it is nevertheless obvious that determining the self-optimizing control structure using brute-force evaluation will be extremely tedious work.

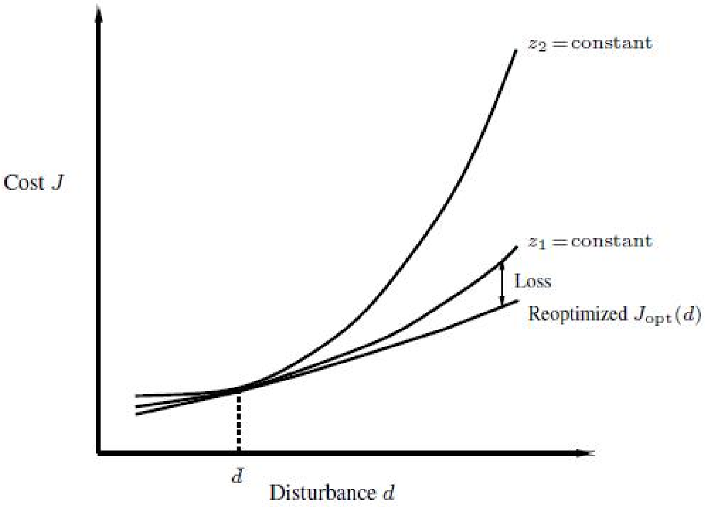


Figure 4.5: Objective function response to disturbances with self-optimizing variables z_1 and z_2 kept at constant setpoints compared to the reoptimized process.[3]

Addition of equality constraints to reduce self-optimizing variables

The previous section determined that the inequality constraint of the sub-cooled LNG temperature is active at the nominal optimum and should be controlled, thus consuming one degree of freedom. This leaves nine degrees of freedom that need assigned control variables for self-optimizing control. If there existed yet more active constraints, or equality constraints in that case, more DOF's would be consumed controlling these constraints and would ease the task of determining a self-optimizing control structure using the brute-force method. This section involves a brief analysis and discussion of reducing the number of DOF's by adding equality constraints.

The degrees of freedom in the C3MR process are summarized in Table 4.7. The two intermediate pressures in the three-stage recompression of mixed refrigerant have no influence on the overall process and should possibly have been removed from the optimization altogether. However, it is reasonable to specify these two pressures to their optimal values given in Section 4.4.

Table 4.7: Degrees of freedom in C3MR process simulation.

Degrees of freedom
C3 flow to NG pre-cooling
C3 flow to MR pre-cooling
P_1 Propane
P_2 Propane
P_3 Propane
MR flow
P_1 MR
P_2 MR
P_3 MR
P_4 MR

It is also a reasonable assumption that the degrees of super-heating of propane are controlled at a given setpoint (typically 10 °C), because super-heating is not optimal for the propane cycle as it does not involve internal heat exchange. The mixed refrigerant cycle on the other hand does involve internal heat exchange, thus it may be optimal with super-heating.[1] The degree of super-heating of mixed refrigerant is therefore not assumed to be controlled.

With these assumptions, totally five degrees of freedom may be ommitted from the self-optimizing control structure out of the 10 original DOF's. This reduction will help ease the work with a brute-force evaluation method determining the remaining self-optimizing control variables. Whether or not the assumptions are valid can only be determined by carrying out the method and evaluating the results, but the reasoning seems to hold. However, performing the brute-force method still requires more time than is readily available for the scope of this project.

5 Discussion

This section will describe the problems experienced throughout this project in further detail, as well as discuss possible alternatives that would provide a more accurate model of the C3MR liquefaction process. Several aspects have been introduced in earlier sections, but will be taken up to discussion again to summarize and effectively identify possible improvements for further work.

5.1 Modelling in UniSim Design

The process model was built in the UniSim Design simulation software by Honeywell based on a previous model in Aspen HYSYS and literature regarding the C3MR process. The software is a graphical interface flowsheeting package which is user-friendly and provides an excellent overview of the process. The model was built in parallel with PhD-student and co-supervisor Magnus Glosli Jacobsen, arriving at almost identical simulation models. This should provide some instance of quality assurance. Several simplifications have been made to the model, such as the omission of NGL extraction and the expansion of sub-cooled LNG, in order to yield a model that would be easier to work with for further optimization and analysis. Many of the process parameters were based on the work done by Jensen in his PhD thesis.[1] Alternative designs of the process simulation are not discussed as the focus of this project was to optimize the operation of an *existing* process.

UniSim Design solves each process unit independently based on the unit and stream specifications. The nature of this approach makes it necessary to break up internal heat exchange streams on the MCHE and add iteration blocks in order to solve the flowsheet. The cold input stream of the MCHE is directly dependent on the cold outlet stream as they represent the same physical stream. The recycle iteration block provides an initial estimate and iterates the process unit until convergence within a specified tolerance. The consequence of this approach is the probability of the flowsheet not being able to converge.

As described in Section 3.2, the heat exchangers in the process simulation are solved in an iterative procedure with respect to user-defined tolerances. It is obviously advantageous with regards to accuracy in calculations to operate with a very tight tolerance, but it will on the other hand cause problems for the flowsheet in terms of convergence when changing the manipulated variables. Many iterations will be required, and the flowsheet may not be able to meet the required tolerance at all. In the C3MR process, or any process involving multiple heat exchangers as the predominant process equipment, tight tolerances create significant difficulties for the robustness of the simulation. The multi-stream LNG heat exchangers proved to be specially vulnerable for this lack of robustness. In order to counter the problems with non-convergence, it was necessary to increase the tolerance in the heat exchangers. The consequence was inaccuracy in flowsheet calculations that naturally were reflected in the value of the objective function and constraints.

5.2 Optimization

Instead of being a fairly smooth curve, it is suspected that the objective function calculated by UniSim Design was subject to significant amount of *noise*. Figure 5.1 illustrates this suspicion, but the graph or values themselves have no connection to the C3MR process. The optimum itself is suspected to be relatively flat, which in turn increases the effect of the inaccuracy in the calculations.

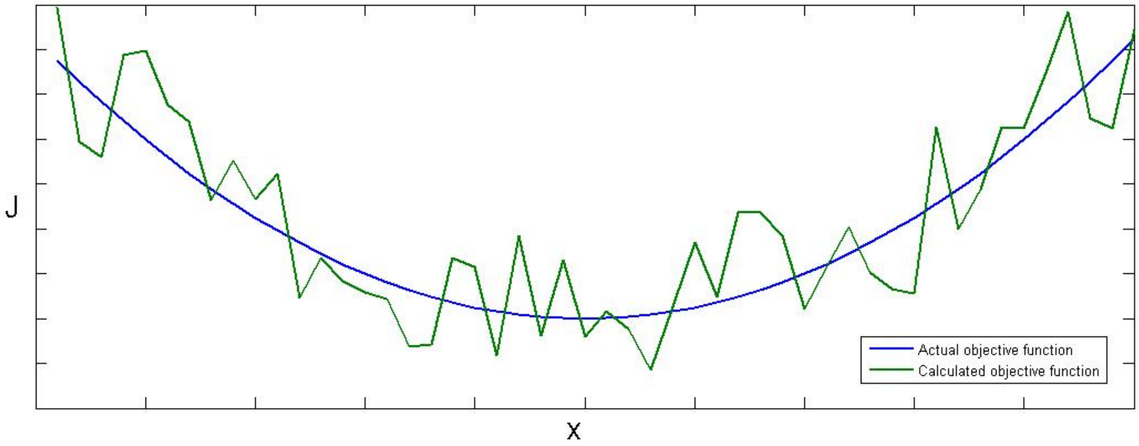


Figure 5.1: Inaccuracy (*noise*) in objective function calculations affecting the optimization. Taking small steps will give incorrect gradients.

The optimization of the C3MR process was attempted using the built-in optimizer in UniSim Design and the MATLAB function *fmincon*. Both optimizers use the sequential quadratic programming (SQP) algorithm for non-linear constrained optimization. The SQP method solves a quadratic program (QP) at each step to determine the search direction for the optimization routine. The quadratic programs are solved using the gradients of the objective function and the constraint functions with respect to the process variables. The gradients are calculated by numerical differentiation by making small steps in the variables, which are again determined by the algorithm but may also be set by the user. Figure 5.1 clearly shows that small steps when numerically determining the gradients will give very inaccurate results for the gradients due to the inaccuracy in the objective function calculations. The incorrect gradients will in turn lead the optimization routine in the wrong direction, thus resulting in meaningless results for the optimization. It is necessary to increase the step size to the extent that the noise in the objective function no longer has a significant effect on the gradient. This method is equivalent to using the average gradient over several smaller steps in the respective range in the variable.

However, using a larger step in variables for numerical determination of the gradients does not necessarily solve the problems regarding optimization. While narrowing down the variance, the larger step may still lead to incorrect gradient and the succeeding problems for the optimizer. The larger step may fail to recognize a minimum in the objective function by stepping over it, therefore not being able to find the true optimum. The magnitude of the variance in the objective function and constraint function calculations compared to the slope of the actual function is the main factor influencing the accuracy of the gradient calculations. The actual 'noise' in this case has not been analyzed, so it is difficult to speculate on the source of the problem. In any case, it would be most beneficial to minimize the amount of noise in the objective function and constraint function calculations in UniSim Design. The necessary action is to tighten the tolerances on process unit calculations, but as described earlier, this was not possible to achieve while maintaining a feasible simulation model. It was necessary for the simulation to be tolerant enough to converge while changing manipulated variables during optimization, as well as tolerate the introduction of disturbances and reoptimization for later analysis of a self-optimizing control structure.

UniSim Design

The *Original* optimizer option in UniSim Design involves a user-friendly interface with most of the desired optimization options available. Unfortunately, the optimizer does operate 'behind the scenes', in terms of not providing the user any insight to the optimization routine or detailed error messages when applicable. The numerical derivatives of the objective function and constraint functions are calculated based on the 'Shift A' and 'Shift B' available in the 'Parameters' tab of the optimizer. Supplementary information can be found in the UniSim Design Operations Guide. [11]

The UniSim Design optimizer operates with user-defined penalty values for the various constraint functions. Each constraint function is multiplied by it's respective penalty value in the optimization calculations, so a higher penalty value implies more weight on meeting that constraint.[11] The most critical problem with the UniSim optimizer throughout the work of this project was the optimizer's inability of meeting constraints. Repetive optimization attempts resulted in subcooled LNG temperatures significantly higher than the constraint value of -156.9 °C, as well as temperature crosses or minimum approach temperatures close to zero in heat exchangers (the constraint was > 0.5 °C). The constraints regarding super-heating of propane were also violated repeatedly. Yet, the optimization claimed to have been successful. The penalty values were varied from their default value of 1 to values excessively larger than the proposed values in the operations guide [11] without any improvement of the results.

The 'Shift A' and 'Shift B' options were varied in order to render a succesful optimization, but regardless of the iterations in the optimization the constraint functions were persistently violated. The reason for the violations is not known, but due to similar experiences of other attempts regarding optimization in UniSim, it was decided to move the optimization to MATLAB. The optimization routines available in MATLAB provide a greater degree of transparency to the user in terms of troubleshooting and debugging, and involve a greater variety of options to the user.

MATLAB

The optimization of the C3MR process in MATLAB was performed using the function for nonlinear constrained optimization. An interface was creating using an ActiveX/COM server running UniSim Design with MATLAB connected as a client. The *fmincon* functions uses the design variables as an initial estimate (x_0) for the optimization, and calls the UniSim Design twice per iteration: Once to calculate the objective function and once to calculate the constraint functions. The double-calling is necessary due to the nature of the *fmincon* function. The objective function and constraint functions must be located in function files that are separate inputs to the optimizer. The two calls to UniSim Design per iteration does not necessarily make the optimization routine any slower, as the calls are made with identical variables and the simulation does not have to redo it's calculations.

MATLAB provides a wide range of options available to the user regarding the optimization with *fmincon*, such as various tolerances on the objective function and constraints, step lengths, iterations, user-defined gradients, etc. Detailed error messages and exit flags are also a benefit. In terms of troubleshooting, the *fmincon* function itself is displayed when errors are encountered, and the problem is pinpointed in the function script.

The drawbacks using an external optimizer such as MATLAB are related to the possibility of an non-converged flowsheet. If UniSim is unable to provide a calculated value for the objective function or constraints to MATLAB, the MATLAB function will fail. Also, MATLAB has no bearing of what equations are behind the calculation of the objective function and thus is operating 'blindly', relying entirely on the results obtained from UniSim Design. In other words, MATLAB is fully dependant on UniSim providing reasonable results in order for *fmincon* to be able to converge to the optimum.

In the case of inaccurate values of the objective function, the optimization options had to be determined carefully. The minimum step size for numerically calculating the gradients of the objective function and constraints had to be raised until a optimization routine was successful. Also, the tolerance of the objective function had to raised to a reasonable value. Contrary to the optimizer in UniSim Design, *fmincon* was able to meet the constraints and provide a result for the optimization. However, the options specified above contribute to an inaccurate solution, as the optimizer is restricted from finding an accurate value for the optimum. Thus, the first-order optimality measure in the obtained results is not equal to zero (as it should be at the true optimum). There is reason to believe, however, that the optimization results are relatively close to optimum. The optimization did yield iterations that succesfully reduced the value of the objective function while meeting the constraints. The process design itself was expected to be close to optimum, as the complicated design with internal heat exchange does not leave much room for variation of process parameters.

The results of the optimization were presented in Section 4.4, but should not be interpreted as accurate results for the nominal optimum for operation of a C3MR process. Nevertheless, the results give an indication of the optimal region, and may be used as a basis for determining a self-optimizing control structure.

5.3 Self-optimizing control

The results of Alstad [14] show that the self-optimizing control structure is not affected by non-optimal nominal setpoints. The maximum scaled gain method is not suitable in this case, as it requires reoptimization after each disturbance that is introduced. The results of the maximum scaled gain method would inherit the inaccuracy from the optimization. Instead, a brute-force evaluation could be performed, systematically testing every combination of control variables at constant setpoints for the expected disturbances.

With nine degrees of freedom left for self-optimizing control (described in Section 2.3), a brute-force evaluation would be tedious work since the process simulation requires significant amounts of 'user-maintenance' in order to converge. By assuming equality constraints on the degrees of super-heating (T_{sup}) of propane as well as specifying the two intermediate pressures of mixed refrigerant, four of the DOF's would be consumed satisfying the mentioned constraints. Five DOF's would be left to determine a self-optimizing control, reducing the work related to the brute-force evaluation. The assumptions seem reasonable. For processes without internal heat exchange, super-heating is non-optimal. However, in order to avoid feeding liquid to the following compressor, the super-heating is controlled at a given setpoint (typically 10 °C).[1] The intermediate pressures of the mixed refrigerant have no effect on the remaining process, and thus should be controlled at their optimal setpoints. Nevertheless, it is necessary to perform the evaluation and test the control structure in dynamic simulation to be able to evaluate whether or not the assumptions hold.

6 Conclusions and further work

6.1 Conclusions

The simulation model of the C3MR process built in UniSim Design does not offer the desired accuracy for optimization and detailed analysis of the process. While being a user-friendly product for process design, UniSim does not provide any transparency towards the equations behind the process units or the procedures of the built-in optimizer. The simulation model proved to be weak in terms of robustness when the tolerances of process calculations were tightened, causing flowsheet convergence failure during optimization. On the other hand, loosening the constraints rendered it difficult to extract accurate information of the objective function and constraints.

The UniSim Design optimizer was not successful at meeting the constraints imposed on the optimization problem. Yet when subject to excessively large penalty values, the optimizer returned with violated constraints and an infeasible solution. Eventually the optimization was carried out in MATLAB. However, the inaccuracy from the UniSim model was transmitted to the optimization in MATLAB, necessitating looser optimizer options and thus giving incorrect results.

The problems encountered due to UniSim Design suggest that the process model should be rebuilt using an alternative simulator, preferrably an equation based simulator providing better robustness and a higher level of transparency towards the process equations. While UniSim Design apparently is an efficient and easy tool for some process simulations, such as distillation systems, it is evidently not the most suitable simulator for complicated LNG processes such as the C3MR process. The multi-stream LNG exchangers with internal heat exchange involve a iterative approach with large room for error. Though simulation and optimization of other LNG processes, such as the PRICO process [15], has been successful using in UniSim/HYSYS, the C3MR process is significantly more complex in terms of number of process equipment.

The problems regarding UniSim Design as an accurate tool for LNG process simulations were suspected prior to the work of this project, and have proven well justified. UniSim Design is not the correct tool for accurate simulation of complicated LNG processes.

6.2 Further work

As described above, more accurate optimization and studies of self-optimizing control structures for the C3MR liquefaction process require a more accurate simulation model. The process model should be built using a better simulation software product, preferrably providing access to the equations and internal structure of the simulator.

An alternative approach to the optimization of the C3MR process could be to build a dynamic model in UniSim Design based on the existing steady-state mode, and control the model at various setpoints to manually search for the optimum steady-state.

Potentially, further work may be done based on the optimization results obtained in this report, though not being truly accurate. It would however be interesting to perform a brute-force evaluation for obtaining a self-optimizing control structure. The results could subsequently be compared to those obtained from an alternative simulation model.

References

[1] Jørgen Bauck Jensen. *Optimal Operation of Refrigeration Cycles.* PhD thesis, NTNU, Faculty of Natural Sciences and Technology, Department of Chemical Engineering, 2008.

[2] Michael J. Economides and Saeid Mokhatab. *Compressed Natural Gas: Monetizing Stranded Gas.* Energy Tribune, `http://www.energytribune.com/articles.cfm?aid=643`.

[3] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, Ltd, second edition, 2005.

[4] *World Energy Outlook.* International Energy Agency, 2009.

[5] Bob Williams. *Future Energy Supply - 1: Oil Depletion.* Oil and Gas Journal, 2003.

[6] Werner Zittel and Jorg Schindler. *Future World Oil Supply.* International Summer School, Salzburg, 2002.

[7] *Strategic Significance of America's Oil Shale Resource. Volume I: Assessment of Strategic Issues.* Office of Naval Petroleum and Oil Shale Reserves. U.S. Department of Energy, 2004.

[8] Mark Pillarella et al. *The C3MR Liquefaction Cycle: Versatility for a fast growing, ever changing LNG industry.* Air Products and Chemicals, Inc.

[9] Mark S. Gockenbach. *Lecture: Introduction to sequential quadratic programming.* Michigan Technological University, `http://www.math.mtu.edu/$\sim$msgocken/ma5630spring2003/lectures/sqp1/sqp1.pdf`.

[10] Moritz Kuhn. *The Karush-Kuhn-Tucker Theorem.* CDSEM Uni Mannheim, 2006.

[11] *UniSim Design Operations Guide.* Honeywell, 2005.

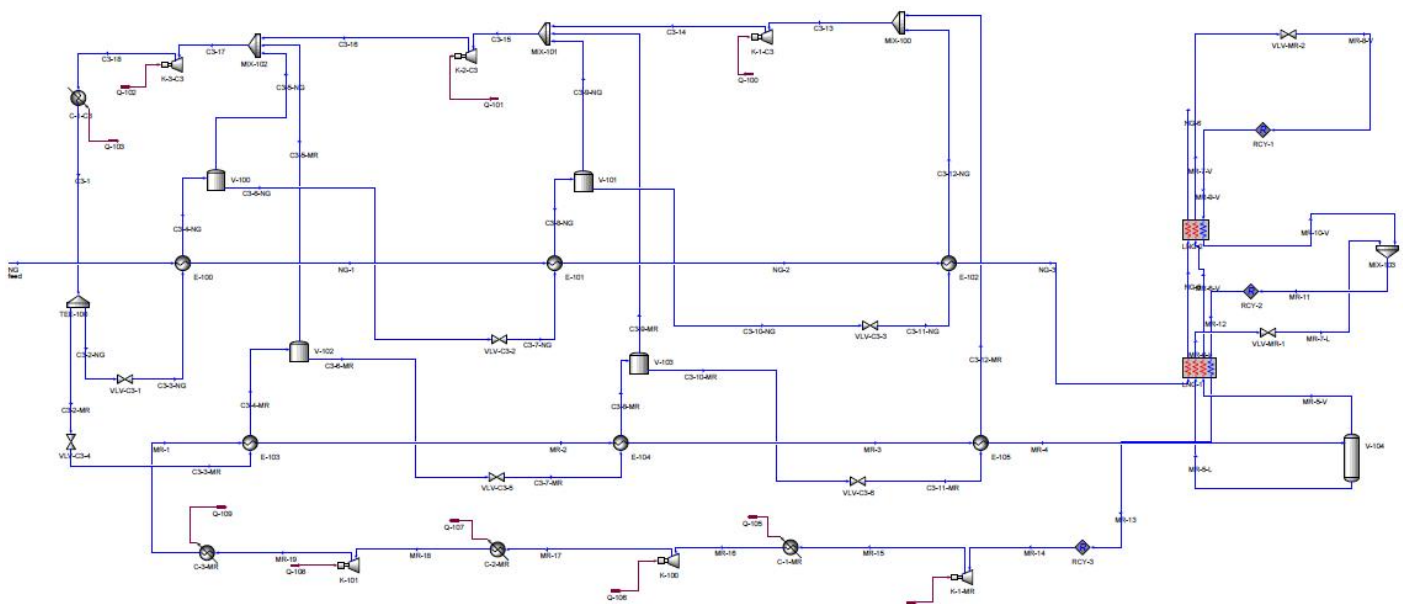
[12] *MATLAB R2009a Documentation.* The MathWorks, Inc., 2009.

[13] Tore Lid. *Data reconciliation and optimal operation - With applications to refinery processes.* PhD thesis, NTNU, Faculty of Natural Sciences and Technology, Department of Chemical Engineering, 2007.

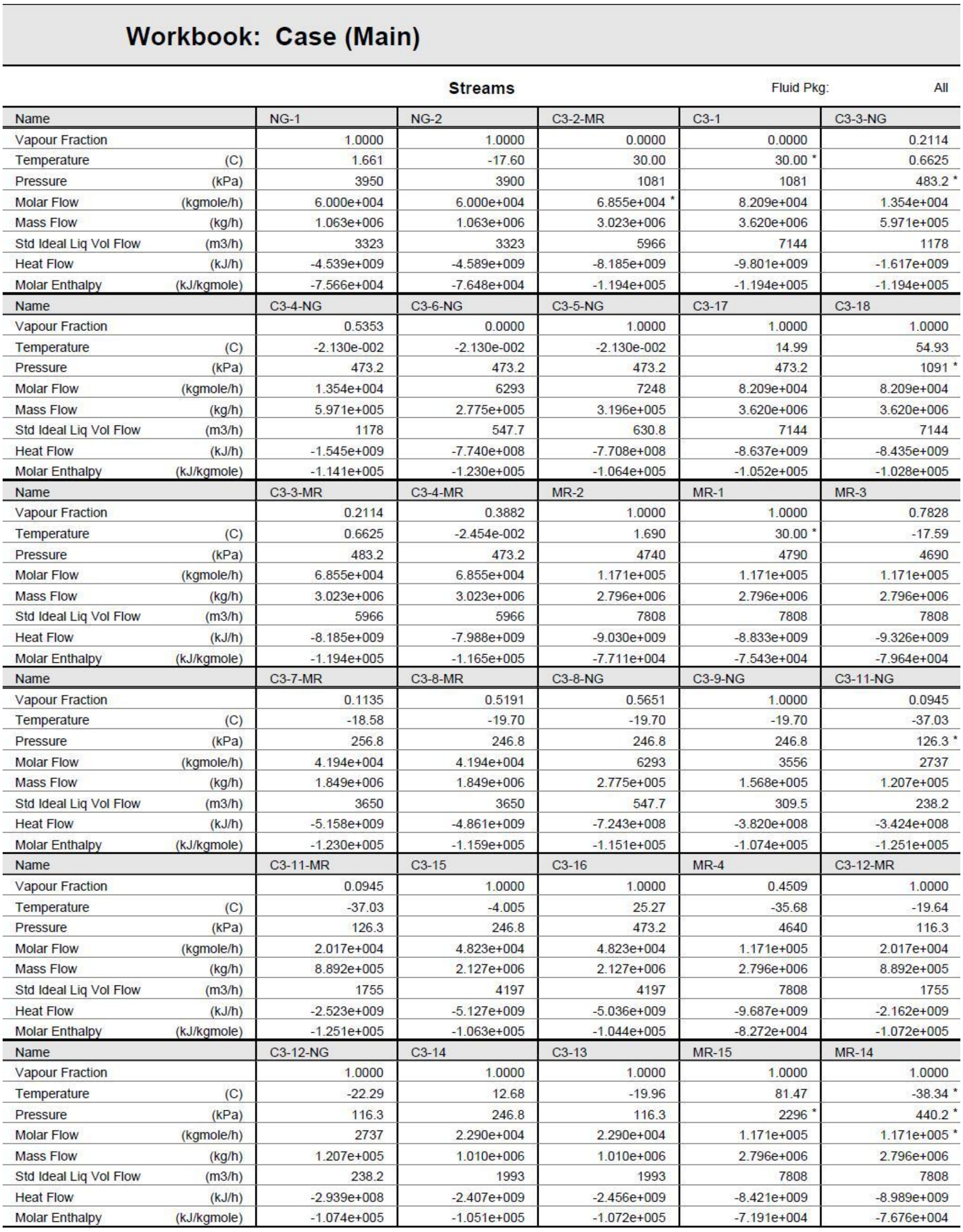
[14] Vidar Alstad. *Studies on selection of controlled variables.* PhD thesis, NTNU, Faculty of Natural Sciences and Technology, Department of Chemical Engineering, 2005.

[15] Luber Perez. *Optimal operation of a LNG process.* MSc thesis, NTNU, Faculty of Natural Sciences and Technology, Department of Chemical Engineering, 2009.

A UniSim Design Flowsheet



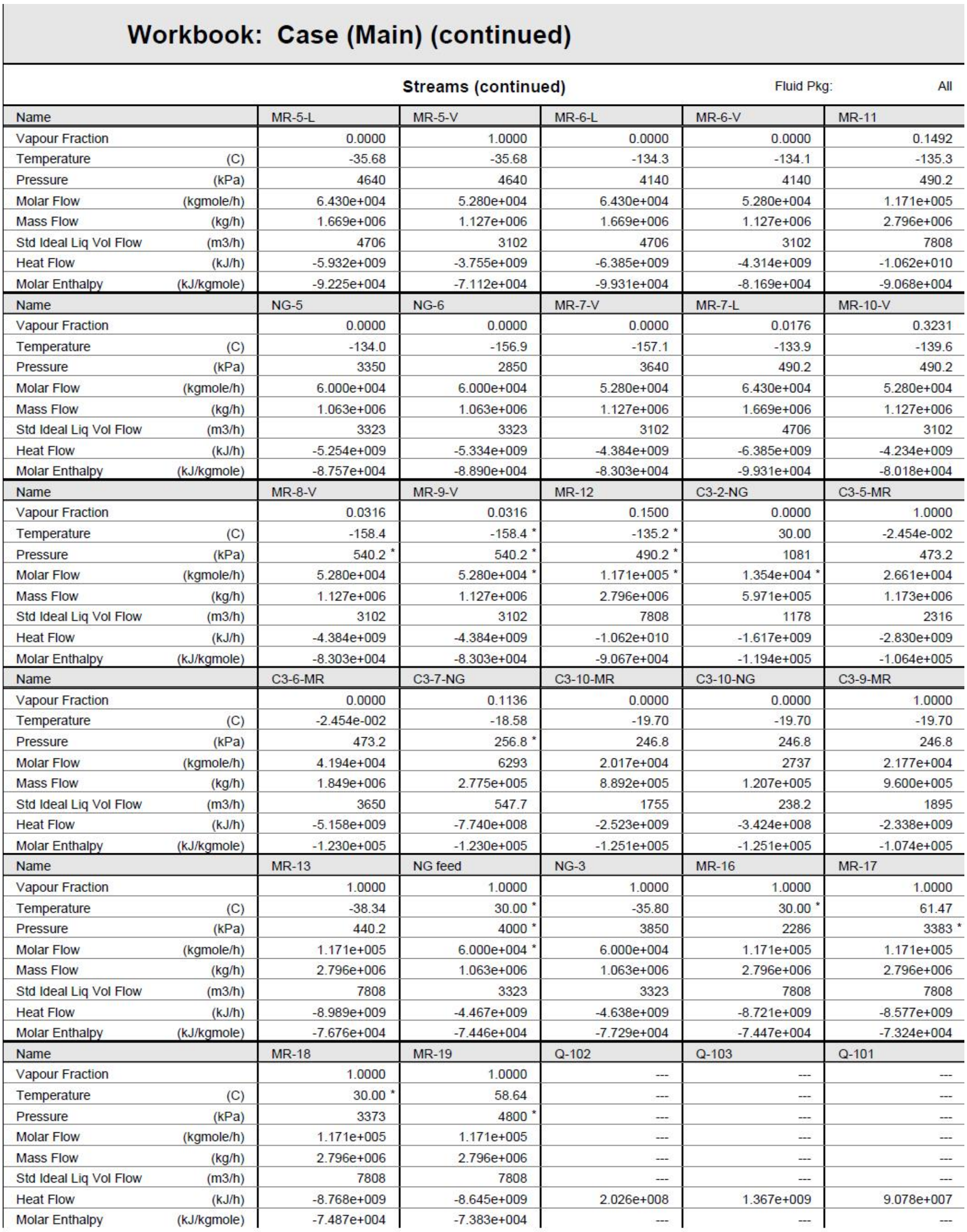
B UniSim Design Workbook



Workbook: Case (Main)

Streams Fluid Pkg: All

Name		NG-1	NG-2	C3-2-MR	C3-1	C3-3-NG
Vapour Fraction		1.0000	1.0000	0.0000	0.0000	0.2114
Temperature	(C)	1.661	-17.60	30.00	30.00 *	0.6625
Pressure	(kPa)	3950	3900	1081	1081	483.2 *
Molar Flow	(kgmole/h)	6.000e+004	6.000e+004	6.855e+004 *	8.209e+004	1.354e+004
Mass Flow	(kg/h)	1.063e+006	1.063e+006	3.023e+006	3.620e+006	5.971e+005
Std Ideal Liq Vol Flow	(m3/h)	3323	3323	5966	7144	1178
Heat Flow	(kJ/h)	-4.539e+009	-4.589e+009	-8.185e+009	-9.801e+009	-1.617e+009
Molar Enthalpy	(kJ/kgmole)	-7.566e+004	-7.648e+004	-1.194e+005	-1.194e+005	-1.194e+005
Name		C3-4-NG	C3-6-NG	C3-5-NG	C3-17	C3-18
Vapour Fraction		0.5353	0.0000	1.0000	1.0000	1.0000
Temperature	(C)	-2.130e-002	-2.130e-002	-2.130e-002	14.99	54.93
Pressure	(kPa)	473.2	473.2	473.2	473.2	1091 *
Molar Flow	(kgmole/h)	1.354e+004	6293	7248	8.209e+004	8.209e+004
Mass Flow	(kg/h)	5.971e+005	2.775e+005	3.196e+005	3.620e+006	3.620e+006
Std Ideal Liq Vol Flow	(m3/h)	1178	547.7	630.8	7144	7144
Heat Flow	(kJ/h)	-1.545e+009	-7.740e+008	-7.708e+008	-8.637e+009	-8.435e+009
Molar Enthalpy	(kJ/kgmole)	-1.141e+005	-1.230e+005	-1.064e+005	-1.052e+005	-1.028e+005
Name		C3-3-MR	C3-4-MR	MR-2	MR-1	MR-3
Vapour Fraction		0.2114	0.3882	1.0000	1.0000	0.7828
Temperature	(C)	0.6625	-2.454e-002	1.690	30.00 *	-17.59
Pressure	(kPa)	483.2	473.2	4740	4790	4690
Molar Flow	(kgmole/h)	6.855e+004	6.855e+004	1.171e+005	1.171e+005	1.171e+005
Mass Flow	(kg/h)	3.023e+006	3.023e+006	2.796e+006	2.796e+006	2.796e+006
Std Ideal Liq Vol Flow	(m3/h)	5966	5966	7808	7808	7808
Heat Flow	(kJ/h)	-8.185e+009	-7.988e+009	-9.030e+009	-8.833e+009	-9.326e+009
Molar Enthalpy	(kJ/kgmole)	-1.194e+005	-1.165e+005	-7.711e+004	-7.543e+004	-7.964e+004
Name		C3-7-MR	C3-8-MR	C3-8-NG	C3-9-NG	C3-11-NG
Vapour Fraction		0.1135	0.5191	0.5651	1.0000	0.0945
Temperature	(C)	-18.58	-19.70	-19.70	-19.70	-37.03
Pressure	(kPa)	256.8	246.8	246.8	246.8	126.3 *
Molar Flow	(kgmole/h)	4.194e+004	4.194e+004	6293	3556	2737
Mass Flow	(kg/h)	1.849e+006	1.849e+006	2.775e+005	1.568e+005	1.207e+005
Std Ideal Liq Vol Flow	(m3/h)	3650	3650	547.7	309.5	238.2
Heat Flow	(kJ/h)	-5.158e+009	-4.861e+009	-7.243e+008	-3.820e+008	-3.424e+008
Molar Enthalpy	(kJ/kgmole)	-1.230e+005	-1.159e+005	-1.151e+005	-1.074e+005	-1.251e+005
Name		C3-11-MR	C3-15	C3-16	MR-4	C3-12-MR
Vapour Fraction		0.0945	1.0000	1.0000	0.4509	1.0000
Temperature	(C)	-37.03	-4.005	25.27	-35.68	-19.64
Pressure	(kPa)	126.3	246.8	473.2	4640	116.3
Molar Flow	(kgmole/h)	2.017e+004	4.823e+004	4.823e+004	1.171e+005	2.017e+004
Mass Flow	(kg/h)	8.892e+005	2.127e+006	2.127e+006	2.796e+006	8.892e+005
Std Ideal Liq Vol Flow	(m3/h)	1755	4197	4197	7808	1755
Heat Flow	(kJ/h)	-2.523e+009	-5.127e+009	-5.036e+009	-9.687e+009	-2.162e+009
Molar Enthalpy	(kJ/kgmole)	-1.251e+005	-1.063e+005	-1.044e+005	-8.272e+004	-1.072e+005
Name		C3-12-NG	C3-14	C3-13	MR-15	MR-14
Vapour Fraction		1.0000	1.0000	1.0000	1.0000	1.0000
Temperature	(C)	-22.29	12.68	-19.96	81.47	-38.34 *
Pressure	(kPa)	116.3	246.8	116.3	2296 *	440.2 *
Molar Flow	(kgmole/h)	2737	2.290e+004	2.290e+004	1.171e+005	1.171e+005 *
Mass Flow	(kg/h)	1.207e+005	1.010e+006	1.010e+006	2.796e+006	2.796e+006
Std Ideal Liq Vol Flow	(m3/h)	238.2	1993	1993	7808	7808
Heat Flow	(kJ/h)	-2.939e+008	-2.407e+009	-2.456e+009	-8.421e+009	-8.989e+009
Molar Enthalpy	(kJ/kgmole)	-1.074e+005	-1.051e+005	-1.072e+005	-7.191e+004	-7.676e+004



Workbook: Case (Main) (continued)

Streams (continued) Fluid Pkg: All

Name		MR-5-L	MR-5-V	MR-6-L	MR-6-V	MR-11
Vapour Fraction		0.0000	1.0000	0.0000	0.0000	0.1492
Temperature	(C)	-35.68	-35.68	-134.3	-134.1	-135.3
Pressure	(kPa)	4640	4640	4140	4140	490.2
Molar Flow	(kgmole/h)	6.430e+004	5.280e+004	6.430e+004	5.280e+004	1.171e+005
Mass Flow	(kg/h)	1.669e+006	1.127e+006	1.669e+006	1.127e+006	2.796e+006
Std Ideal Liq Vol Flow	(m3/h)	4706	3102	4706	3102	7808
Heat Flow	(kJ/h)	-5.932e+009	-3.755e+009	-6.385e+009	-4.314e+009	-1.062e+010
Molar Enthalpy	(kJ/kgmole)	-9.225e+004	-7.112e+004	-9.931e+004	-8.169e+004	-9.068e+004

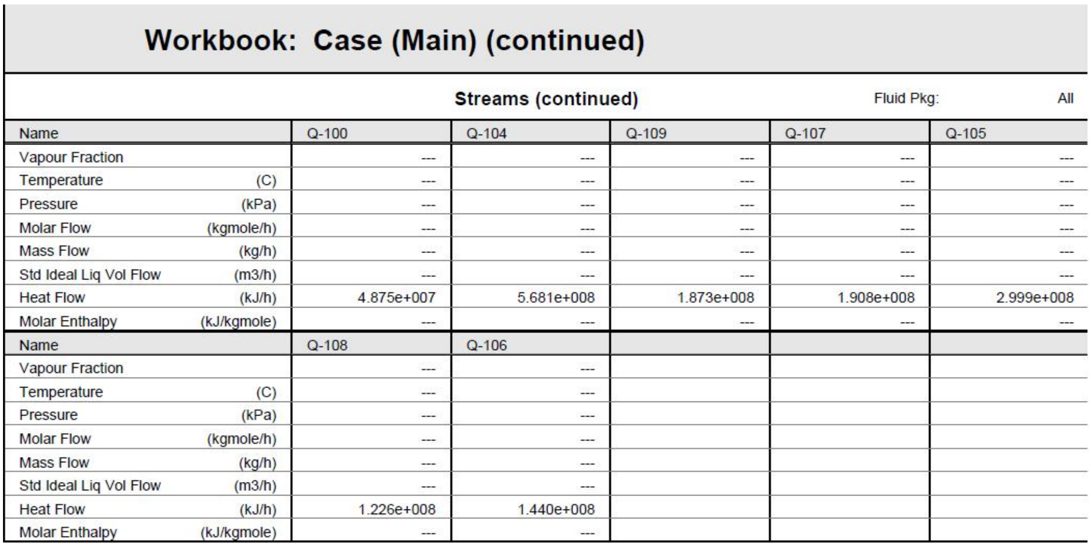
Name		NG-5	NG-6	MR-7-V	MR-7-L	MR-10-V
Vapour Fraction		0.0000	0.0000	0.0000	0.0176	0.3231
Temperature	(C)	-134.0	-156.9	-157.1	-133.9	-139.6
Pressure	(kPa)	3350	2850	3640	490.2	490.2
Molar Flow	(kgmole/h)	6.000e+004	6.000e+004	5.280e+004	6.430e+004	5.280e+004
Mass Flow	(kg/h)	1.063e+006	1.063e+006	1.127e+006	1.669e+006	1.127e+006
Std Ideal Liq Vol Flow	(m3/h)	3323	3323	3102	4706	3102
Heat Flow	(kJ/h)	-5.254e+009	-5.334e+009	-4.384e+009	-6.385e+009	-4.234e+009
Molar Enthalpy	(kJ/kgmole)	-8.757e+004	-8.890e+004	-8.303e+004	-9.931e+004	-8.018e+004

Name		MR-8-V	MR-9-V	MR-12	C3-2-NG	C3-5-MR
Vapour Fraction		0.0316	0.0316	0.1500	0.0000	1.0000
Temperature	(C)	-158.4	-158.4 *	-135.2 *	30.00	-2.454e-002
Pressure	(kPa)	540.2 *	540.2 *	490.2 *	1081	473.2
Molar Flow	(kgmole/h)	5.280e+004	5.280e+004 *	1.171e+005 *	1.354e+004 *	2.661e+004
Mass Flow	(kg/h)	1.127e+006	1.127e+006	2.796e+006	5.971e+005	1.173e+006
Std Ideal Liq Vol Flow	(m3/h)	3102	3102	7808	1178	2316
Heat Flow	(kJ/h)	-4.384e+009	-4.384e+009	-1.062e+010	-1.617e+009	-2.830e+009
Molar Enthalpy	(kJ/kgmole)	-8.303e+004	-8.303e+004	-9.067e+004	-1.194e+005	-1.064e+005

Name		C3-6-MR	C3-7-NG	C3-10-MR	C3-10-NG	C3-9-MR
Vapour Fraction		0.0000	0.1136	0.0000	0.0000	1.0000
Temperature	(C)	-2.454e-002	-18.58	-19.70	-19.70	-19.70
Pressure	(kPa)	473.2	256.8 *	246.8	246.8	246.8
Molar Flow	(kgmole/h)	4.194e+004	6293	2.017e+004	2737	2.177e+004
Mass Flow	(kg/h)	1.849e+006	2.775e+005	8.892e+005	1.207e+005	9.600e+005
Std Ideal Liq Vol Flow	(m3/h)	3650	547.7	1755	238.2	1895
Heat Flow	(kJ/h)	-5.158e+009	-7.740e+008	-2.523e+009	-3.424e+008	-2.338e+009
Molar Enthalpy	(kJ/kgmole)	-1.230e+005	-1.230e+005	-1.251e+005	-1.251e+005	-1.074e+005

Name		MR-13	NG feed	NG-3	MR-16	MR-17
Vapour Fraction		1.0000	1.0000	1.0000	1.0000	1.0000
Temperature	(C)	-38.34	30.00 *	-35.80	30.00 *	61.47
Pressure	(kPa)	440.2	4000 *	3850	2286	3383 *
Molar Flow	(kgmole/h)	1.171e+005	6.000e+004 *	6.000e+004	1.171e+005	1.171e+005
Mass Flow	(kg/h)	2.796e+006	1.063e+006	1.063e+006	2.796e+006	2.796e+006
Std Ideal Liq Vol Flow	(m3/h)	7808	3323	3323	7808	7808
Heat Flow	(kJ/h)	-8.989e+009	-4.467e+009	-4.638e+009	-8.721e+009	-8.577e+009
Molar Enthalpy	(kJ/kgmole)	-7.676e+004	-7.446e+004	-7.729e+004	-7.447e+004	-7.324e+004

Name		MR-18	MR-19	Q-102	Q-103	Q-101
Vapour Fraction		1.0000	1.0000	---	---	---
Temperature	(C)	30.00 *	58.64	---	---	---
Pressure	(kPa)	3373	4800 *	---	---	---
Molar Flow	(kgmole/h)	1.171e+005	1.171e+005	---	---	---
Mass Flow	(kg/h)	2.796e+006	2.796e+006	---	---	---
Std Ideal Liq Vol Flow	(m3/h)	7808	7808	---	---	---
Heat Flow	(kJ/h)	-8.768e+009	-8.645e+009	2.026e+008	1.367e+009	9.078e+007
Molar Enthalpy	(kJ/kgmole)	-7.487e+004	-7.383e+004	---	---	---



Workbook: Case (Main) (continued)

Streams (continued)

Fluid Pkg: All

Name		Q-100	Q-104	Q-109	Q-107	Q-105
Vapour Fraction		---	---	---	---	---
Temperature	(C)	---	---	---	---	---
Pressure	(kPa)	---	---	---	---	---
Molar Flow	(kgmole/h)	---	---	---	---	---
Mass Flow	(kg/h)	---	---	---	---	---
Std Ideal Liq Vol Flow	(m3/h)	---	---	---	---	---
Heat Flow	(kJ/h)	4.875e+007	5.681e+008	1.873e+008	1.908e+008	2.999e+008
Molar Enthalpy	(kJ/kgmole)	---	---	---	---	---
Name		Q-108	Q-106			
Vapour Fraction		---	---			
Temperature	(C)	---	---			
Pressure	(kPa)	---	---			
Molar Flow	(kgmole/h)	---	---			
Mass Flow	(kg/h)	---	---			
Std Ideal Liq Vol Flow	(m3/h)	---	---			
Heat Flow	(kJ/h)	1.226e+008	1.440e+008			
Molar Enthalpy	(kJ/kgmole)	---	---			

C Interfacing and Optimization in MATLAB

The main script and functions for optimization in MATLAB.

main.m:

```
%Optimizer for C3MR process

% The "import/export" units between Unisim and Matlab are as follows:
% Pressure: kPa
% Temperature: degrees C
% Molar flow: kmol/s

%Access UniSim flowsheet
h=ActXServer('UnisimDesign.Application');
hyCase = h.Activedocument;
sol=hyCase.Solver;
f = hyCase.Flowsheet;

%Get initial values from UniSim base case
x0 = zeros(10,1);
x0(1) = f.MaterialStreams.Item('C3-11-NG').PressureValue; %P1_C3
x0(2) = f.MaterialStreams.Item('C3-7-NG').PressureValue; %P2_C3
x0(3) = f.MaterialStreams.Item('C3-3-NG').PressureValue; %P3_C3
x0(4) = f.MaterialStreams.Item('4').PressureValue; %P3_MR
x0(5) = f.MaterialStreams.Item('MR-8-V').PressureValue; %Pc_MR
x0(6) = f.MaterialStreams.Item('C3-2-NG').MolarFlowValue; %flow1_C3
x0(7) = f.MaterialStreams.Item('C3-2-MR').MolarFlowValue; %flow2_C3
x0(8) = f.MaterialStreams.Item('MR-14').MolarFlowValue; %flow_MR
x0(9) = f.MaterialStreams.Item('MR-15').PressureValue; %P1_MR
x0(10) = f.MaterialStreams.Item('2').PressureValue; %P2_MR


%Set lower and upper bounds so UniSim is certain to converge
lb = [
    125
    240
    470
    4800
    540
    13400/3600
    68300/3600
    116000/3600
    2200
    3000
    ];

ub = [
    130
    260
    490
    5000
    560
    14000/3600
    69000/3600
    118000/3600
    2400
    3700
    ];

%Call optimizer function fmincon to minimize function objaektiv with given
%bounds and constraints in C3MR_constraints.
options = optimset('TolFun',1e-6,'DiffMinChange',1e-3,'MaxFunEval',1000,
'Display','iter','TypicalX', x0, 'PlotFcns', @optimplotfirstorderopt);
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(@objective,x0,[],
[],[],[],lb,ub,@constraints,options)

sol.CanSolve = 0 ;
```

objective.m:

```
function J = objective(x)

% Calculate objective function.

h=ActXServer('UnisimDesign.Application');
hyCase = h.Activedocument;
sol=hyCase.Solver;
f = hyCase.Flowsheet;

sol.CanSolve = 0 ;

%We'll work with 10 variables.

P1_C3 = x(1) ; % Lowest pressure in propane cycle
P2_C3 = x(2) ; % Second lowest pressure in propane cycle
P3_C3 = x(3) ; % Third lowest pressure in propane cycle
P3_MR = x(4) ; % Pressure out of K105, highest pressure in MR cycle
Pc_MR = x(5) ; % Pressure in cold end of plant, after J-T expansion of MR
flow1_C3 = x(6) ; % Propane flow to NG precooling
flow2_C3 = x(7) ; % Propane flow to MR precooling
flow_MR = x(8) ;  % Flow of mixed refrigerant
P1_MR = x(9) ; % Pressure out of K1-MR
P2_MR = x(10) ; % PRessure out of K-100


f.MaterialStreams.Item('C3-11-NG').PressureValue = P1_C3;
f.MaterialStreams.Item('C3-7-NG').PressureValue = P2_C3;
f.MaterialStreams.Item('C3-3-NG').PressureValue = P3_C3;
f.MaterialStreams.Item('4').PressureValue = P3_MR;
f.MaterialStreams.Item('MR-8-V').PressureValue = Pc_MR;
f.MaterialStreams.Item('C3-2-NG').MolarFlowValue = flow1_C3;
f.MaterialStreams.Item('C3-2-MR').MolarFlowValue = flow2_C3;
f.MaterialStreams.Item('MR-14').MolarFlowValue = flow_MR;
f.MaterialStreams.Item('MR-15').PressureValue = P1_MR;
f.MaterialStreams.Item('2').PressureValue = P2_MR;


sol.CanSolve = 1 ;


J = f.Operations.Item('SPRDSHT-2').Imports.Item('A5: A1:').CellValue;

end
```

constraints.m:

```
function [C Ceq] = constraints(x)
%Constraints - calculate the constraint values of the C3MR model for
%optimization

h=ActXServer('UnisimDesign.Application');
hyCase = h.Activedocument;
sol=hyCase.Solver;
f = hyCase.Flowsheet;

sol.CanSolve = 0 ;

P1_C3 = x(1) ; % Lowest pressure in propane cycle
P2_C3 = x(2) ; % Second lowest pressure in propane cycle
P3_C3 = x(3) ; % Third lowest pressure in propane cycle
P3_MR = x(4) ; % Pressure out of K105, highest pressure in MR cycle
Pc_MR = x(5) ; % Pressure in cold end of plant, after J-T expansion of MR
flow1_C3 = x(6) ; % Propane flow to NG precooling
flow2_C3 = x(7) ; % Propane flow to MR precooling
flow_MR = x(8) ;  % Flow of mixed refrigerant
P1_MR = x(9) ; % Pressure out of K1-MR
P2_MR = x(10) ; % PRessure out of K-100


f.MaterialStreams.Item('C3-11-NG').PressureValue = P1_C3;
f.MaterialStreams.Item('C3-7-NG').PressureValue = P2_C3;
f.MaterialStreams.Item('C3-3-NG').PressureValue = P3_C3;
f.MaterialStreams.Item('4').PressureValue = P3_MR;
f.MaterialStreams.Item('MR-8-V').PressureValue = Pc_MR;
f.MaterialStreams.Item('C3-2-NG').MolarFlowValue = flow1_C3;
f.MaterialStreams.Item('C3-2-MR').MolarFlowValue = flow2_C3;
f.MaterialStreams.Item('MR-14').MolarFlowValue = flow_MR;
f.MaterialStreams.Item('MR-15').PressureValue = P1_MR;
f.MaterialStreams.Item('2').PressureValue = P2_MR;

sol.CanSolve = 1 ;

T_minapp = 0.5;

Ceq = [] ; %No equality constraints
C=zeros(12,1);
C(1)  =  T_minapp - f.Operations.Item('E-100').MinimumApproachValue ;
C(2)  =  T_minapp - f.Operations.Item('E-101').MinimumApproachValue ;
C(3)  =  T_minapp - f.Operations.Item('E-102').MinimumApproachValue ;
C(4)  =  T_minapp - f.Operations.Item('E-103').MinimumApproachValue ;
C(5)  =  T_minapp - f.Operations.Item('E-104').MinimumApproachValue ;
C(6)  =  T_minapp - f.Operations.Item('E-105').MinimumApproachValue ;
C(7)  =  T_minapp - f.Operations.Item('LNG-1').MinApproachValue ;
C(8)  =  T_minapp - f.Operations.Item('LNG-2').MinApproachValue ;
C(9)  = 10 - f.Operations.Item('SPRDSHT-2').Imports.Item('A2: C7:').CellValue ;
C(10) = 10 - f.Operations.Item('SPRDSHT-2').Imports.Item('A3: C8:').CellValue ;
C(11) = 10 - f.Operations.Item('SPRDSHT-2').Imports.Item('A4: C9:').CellValue ;
C(12) = f.MaterialStreams.Item('NG-6').TemperatureValue + 156.9;

end
```

D MATLAB optimization results

```
Warning: Trust-region-reflective method does not currently solve this type of
problem,
 using active-set (line search) instead.
> In fmincon at 439
  In main at 58

                                Max     Line search  Directional  First-order
 Iter F-count        f(x)   constraint   steplength   derivative   optimality Procedure
    0      11      328482            0
    1      27      328144            0       0.0313         -360    3.74e+003
    2      40      327506    -0.003976         0.25         -177    5.47e+003  Hessian modifi
    3      52      326790     -0.02621          0.5         -165    6.72e+003
    4      68      326664   -0.0006657       0.0313        -35.8     3.1e+003

Local minimum possible. Constraints satisfied.

fmincon stopped because the predicted change in the objective function
is less than the selected value of the function tolerance and constraints
were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

No active inequalities.

x =

  1.0e+003 *

    0.1263
    0.2568
    0.4832
    4.8001
    0.5402
    0.0038
    0.0190
    0.0325
    2.2962
    3.3830


fval =

  3.2666e+005
```

```
exitflag =

     5


output =

         iterations: 5
          funcCount: 99
       lssteplength: 4.6566e-010
           stepsize: 1.9327e-008
          algorithm: 'medium-scale: SQP, Quasi-Newton, line-search'
      firstorderopt: 3.0976e+003
    constrviolation: -6.6575e-004
            message: [1x847 char]


lambda =

         lower: [10x1 double]
         upper: [10x1 double]
         eqlin: [0x1 double]
      eqnonlin: [0x1 double]
       ineqlin: [0x1 double]
    ineqnonlin: [12x1 double]


grad =

  1.0e+003 *

   -0.4483
    0.2500
    0.0436
    0.1138
    0.1031
    1.8950
    4.2961
    7.1287
    0.1316
    0.1280
```

```
hessian =

  1.0e+005 *

  Columns 1 through 8

    0.0074    0.0100   -0.0310    0.0063    0.0033    0.0347   -0.0482   -0.0501
    0.0100    0.0558    0.0038    0.0179    0.0165    0.0454    0.2112   -0.0706
   -0.0310    0.0038    0.1877   -0.0168   -0.0004   -0.1549    0.5143    0.2242
    0.0063    0.0179   -0.0168    0.0076    0.0055    0.0300    0.0185   -0.0435
    0.0033    0.0165   -0.0004    0.0055    0.0050    0.0151    0.0575   -0.0219
    0.0347    0.0454   -0.1549    0.0300    0.0151    0.1724   -0.2522   -0.2468
   -0.0482    0.2112    0.5143    0.0185    0.0575   -0.2522    2.1563    0.3244
   -0.0501   -0.0706    0.2242   -0.0435   -0.0219   -0.2468    0.3244    0.3862
    0.0147    0.0987    0.0249    0.0301    0.0293    0.0661    0.4196   -0.0998
    0.0148    0.0995    0.0252    0.0304    0.0295    0.0666    0.4232   -0.1006

  Columns 9 through 10

    0.0147    0.0148
    0.0987    0.0995
    0.0249    0.0252
    0.0301    0.0304
    0.0293    0.0295
    0.0661    0.0666
    0.4196    0.4232
   -0.0998   -0.1006
    0.1770    0.1784
    0.1784    0.1799


Optimization stopped because the predicted change in the objective function,
9.689413e-007, is less than the selected value of options.TolFun = 1.000000e-006,
and the maximum constraint violation, -8.463119e-003, is less than the default
value of options.TolCon = 1.000000e-006.

Optimization Metric                                           User Options
abs(steplength*directional derivative) =  9.69e-007       TolFun =  1e-006 (selected)
max(constraint violation) = -8.46e-003                    TolCon =  1e-006 (default)
```