NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for naturvitenskap og teknologi
Institutt for kjemisk prosessteknologi

SPECIALIZATION PROJECT 2011

TKP 4550

PROJECT TITLE:
Modelling and optimization of a PRICO LNG plant

By
Roald Bræck Leer

Supervisor for the project: Professor Sigurd Skogestad
Co-supervisor: Vladimiros Minasidis
Date: 16.12.2011

Abstract

This project was done as a preparatory part of the MSc in Chemical engineering at NTNU, Trondheim autumn 2011.

A Poly refrigerant integral cycle operation (PRICO) process has been modelled in Unisim Design. For the optimization, an interface was created between Matlab and Unisim Design, optimizing the model from Matlab.

A potential degree of freedom analysis was done for the process, indentifying the potential degrees of freedom. The actual degrees of freedom were found by subtracting the fixed variables and disturbances, giving a total of 4 degrees of freedom for the process.

Problems with optimizing the model when not converging were solved by adding penalties to the objective function and constraints.

A self-optimizing control structure for this process was not determined in this project. However, key aspects and possible procedures were addressed.

Acknowledgement

First of all I would like to thank my co-supervisor Vladimiros Minasidis for his guidance and valuable help throughout the project. His approach to solving problems through logical dissection is impressive, especially since he always seems to find the right answer.

Second, I would like to thank my supervisor Professor Sigurd Skogestad. Even though he has an immensely busy schedule, he always has time to answer questions.

Thank you both!

-Roald Bræck Leer

Table of Contents

1. Introduction

As the world population surpasses 7 billion people the need for energy is ever present. The global energy demand is expected to rise from 12 300 million tons of oil equivalent (Mtoe) in 2008 to 16 000 Mtoe in 2035 [1], an increase of 35%. Political instabilities, costly recovery methods and depleting reserves all contribute to a high oil price [2], which again stresses the need for other energy sources in the future.

Natural gas, as a reliable and clean energy source, is a good alternative to funnel this rising energy demand. Technological innovations over the last years have established natural gas as a major and competitive energy fuel [1]. Opposed to oil reserves, natural gas is still in abundance and also more evenly spread out geographically. Natural gas is normally transported through pipelines. However, this is not always economically or physically feasible. Pipelines over long distances and large depths are expensive and sometime dangerous [3].

When cooled to -162 ^{o}C natural gas becomes liquefied natural gas, decreasing its volume to make transportation easier. The liquefaction of natural gas is a well-proven technology with numerous of different designs used today. However, the construction and operation of an LNG plant is costly. Liquefaction trains accounts for almost half the operational costs for an LNG plant [4]. The Poly refrigerant integral cycle operation (PRICO) is the LNG process that is the basis for this project.

This specialization project was motivated by previous difficulties in optimizing LNG-processes. Especially the built-in optimizers in simulation programs have a tendency not being able to converge due to tight integration and small temperature difference between streams in the heat exchangers. Lifting the optimization outside of the simulation program by using Matlab is done in this project.

The goal for this specialization project was to model a PRICO LNG-train in Honeywell's Unisim Design, optimize the model using Matlab. Afterwards a "self-optimizing" control structure can be looked at.

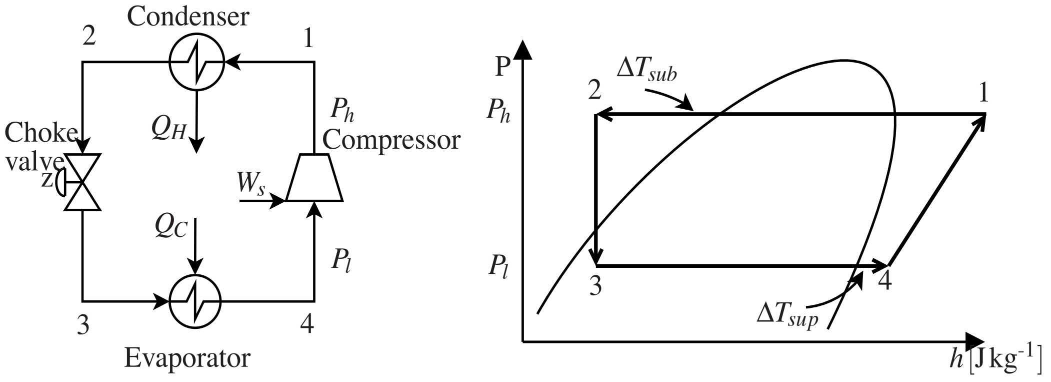
2. Background

This section introduces the theory that has been used in this project. The vapor-compression cycle shows the basics behind refrigeration. Liquefaction of natural gas is explained with emphasis on the PRICO process and the C3MR process for comparison. The basics of optimization and self-optimizing control are also covered. Note that this is not a complete theoretical study and should rather be viewed as more of an introduction to the theory.

2.1 Refrigeration

The use of refrigeration has expanded rapidly since a series of technological improvements in the first half of the 20th century [5]. Today, very few homes in the western world do not have any type of refrigeration. Just imagine the problems with food perseverance if not for refrigerators. Refrigeration is common industrially with one of the uses being liquefaction of natural gas.

The vapor-compression cycle is a simple refrigeration cycle where thermal energy is transferred from an area with low temperature to an area with higher temperature. This applies for everyday items such as household refrigerators and air-conditioners. A simple flowsheet of such a cycle is shown in Figure 2.1, along with its corresponding pressure–enthalpy diagram.



(a) Flowsheet

(b) Pressure enthalpy diagram

Figure 2.1: Vapor-compression cycle for a simple refrigeration process, with (a) flowsheet and (b) pressure-enthalpy diagram [6].

The cycle consists of an evaporator and a condenser, a compressor and a choke valve. The working fluid is evaporated in step $3 \rightarrow 4$ to superheated vapor by absorbing heat from the surroundings. The working fluid is then compressed from step $4 \rightarrow 1$ increasing its energy. After compression the high pressure vapor is condensed to subcooled liquid, step $1 \rightarrow 2$, by releasing heat to the surroundings. The liquid is expanded through a choke valve in step $2 \rightarrow 3$ to give a two-phase flow before entering the evaporator.

Following the consecutive steps for the cycle in the pressure-enthalpy diagram, the pressure P_l remains constant as the working fluid is evaporated from step $3 \rightarrow 4$. The enthalpy increases. At 4, the vapor is superheated which is normal for real refrigeration processes [5]. Vapor inside the evaporator continues to absorb heat from the surroundings even though it is saturated, thus being superheated before entering the compressor. Both the pressure and enthalpy increase when the vapor is compressed. The high-pressure vapor is condensed and subcooled. The same argument for superheating is valid for subcooling. The working fluid will continue to remove heat after it has been condensed, lowering its temperature below saturation temperature.

The refrigeration cycle's efficiency is often measured by its Coefficient of performance (COP) given in Equation 2.1

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

where Q_c is the heat removed from system and W_s is the compressor work. The maximum efficiency is limited to the Carnot efficiency, which is the maximum efficiency for a working body between a hot and cold reservoir. A schematic representation is given in Figure 2.2.

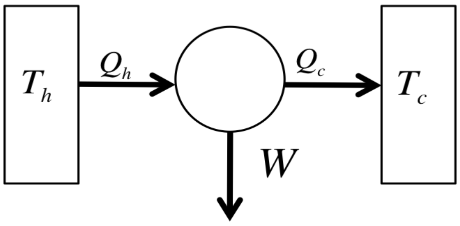


Figure 2.2: Schematic representation of a Carnot Engine.

The Carnot efficiency can be derived from the Second law of thermodynamics, $\Delta S \geq 0$. For a cyclic engine as the Carnot engine the net change in entropy is zero. From this the Carnot efficiency can be derived. For a reversible refrigeration process

$$\frac{Q_h}{T_h} = \frac{Q_c}{T_c} \tag{2.2}$$

and from the energy balance for the Carnot Engine

$$W_s = Q_h - Q_c = \frac{Q_c}{T_c}T_h - Q_c \tag{2.3}$$

Substituting Equation 2.3 into 2.1 gives the Carnot efficiency, Equation 2.4, which is the maximum efficiency for a refrigeration cycle.

$$COP_{Carnot} = \frac{Q_c}{\left(\frac{T_h}{T_c} - 1\right)Q_c} = \frac{T_c}{T_h - T_c} \tag{2.4}$$

From Equation 2.4 it is easily seen that a refrigeration process is favoured by a small difference between the hot and cold side.

2.2 Liquefied Natural gas

Transportation of natural gas is, when both economically and physically feasible, done through pipelines. However, this is not always the case. Due to long distances and harsh terrain natural gas is liquefied to decrease its volume, thus making easier to transport and store [3].

Natural gas is, for atmospheric pressures, liquefied at -162 ^{o}, decreasing its volume by a factor of 600. The liquefaction process follows the basic principle of the vapor-compression of refrigeration cycle described above. To cool the natural gas below its dew point a refrigerant removes the heat energy from the gas. The refrigerant is cooled in either an air cooler or seawater cooler and expanded before cooling the natural gas.

The refrigerant may either be pure or mixed. As seen from Figure 2.3 the cooling curve for mixed refrigerant is closer to that of natural gas than a pure refrigerant and therefore more efficient. For a pure refrigerant the cooling curve takes on staircase form. To make up for a large temperature difference between the hot and the cold side, multiple cascades are added to obtain the same results for a pure refrigerant as for a mixed refrigerant.

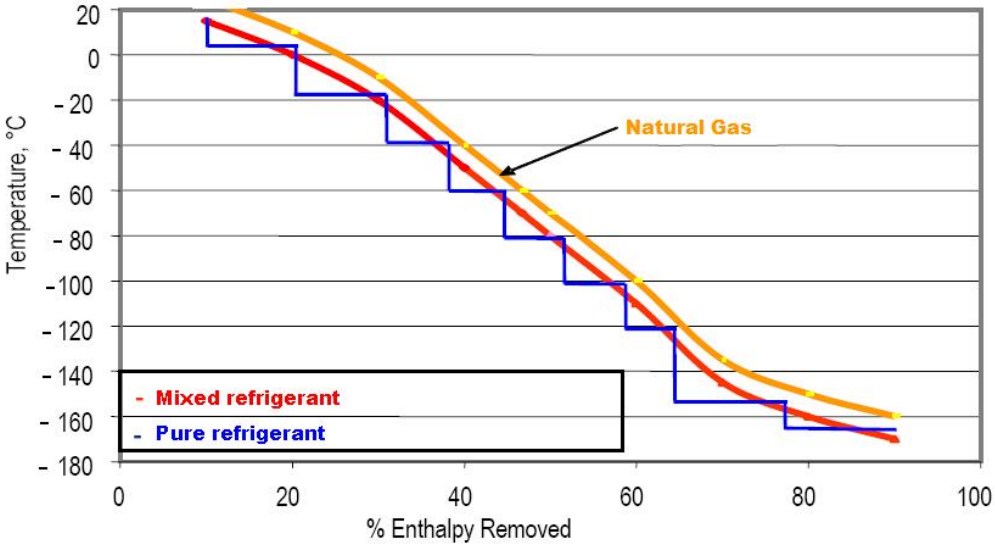


Figure 2.3: Cooling curve for pure and mixed refrigerant [].

There a wide range of liquefaction processes used world wide. All are based on the same refrigeration priciples, but they vary in both set-up and complexity.

Air Products and Chemicals Inc. have developed the C3MR process, which is widely used for liquefaction of natural gas. Around 80% of the liquefaction plants worldwide use this technology [7]. Train capacities are up to 5 Mtpa [15]. There are two main refrigerant cycles in the C3MR process. A precooling cycle that uses propane and a liquefaction cycle that uses a mixed refrigerant. The precooling cycle cools the natural gas and also cools and partially liquefy the mixed refrigerant. For the mixed refrigerant cycle the refrigerant is separated into vapor and liquid for cooling the natural gas in the main cryogenic heat exchanger. A flowsheet of the process is given in Figure 2.4.

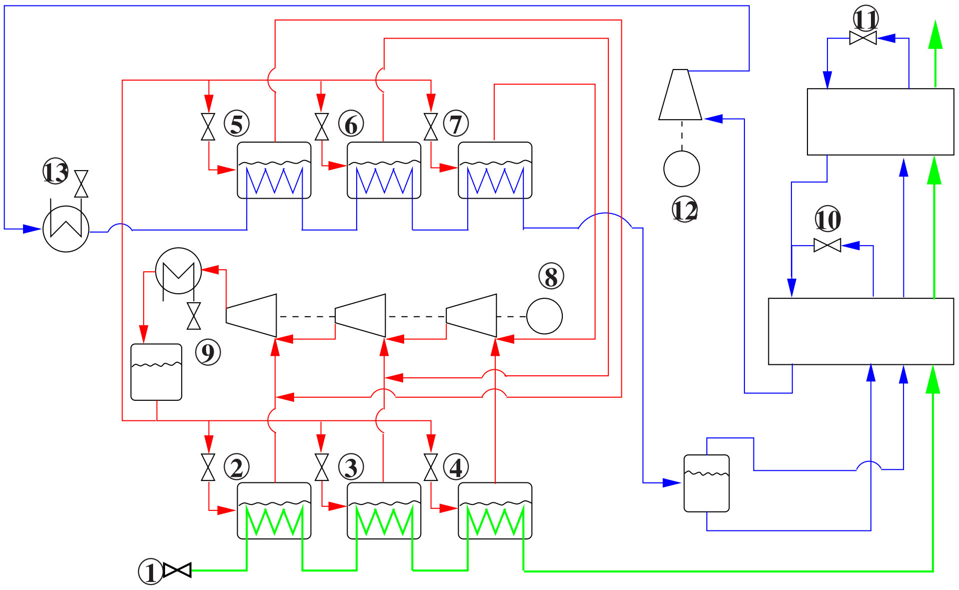


Figure 2.4: Flowsheet of C3MR cycle. The blue cycle is the propane precooling, the green is natural gas and the red is the mixed refrigerant [6]

Poly Refrigerant Integral Cycle Operation

The LNG process studied for this project is the Poly Refrigerant Integral Cycle Operation, abbreviated PRICO for the reminder of the report. The technology dates back to the 1960s, with the first baseload LNG plant in Algeria in 1965. A single mixed refrigerant loop is employed to liquefy the gas. The refrigerant is a mixture of hydrocarbons from methane to pentane and nitrogen. Normally this is drawn from the feed gas and added nitrogen [8].

Because of its simplicity with only one large heat exchanger network and a single refrigerant loop, it has a capacity of up to 1.3 Mtpa [15]. Therefore, it is often preferred for small-scale plants such as peak-shaving plants. A flowsheet of the PRICO process is shown in Figure 2.5.

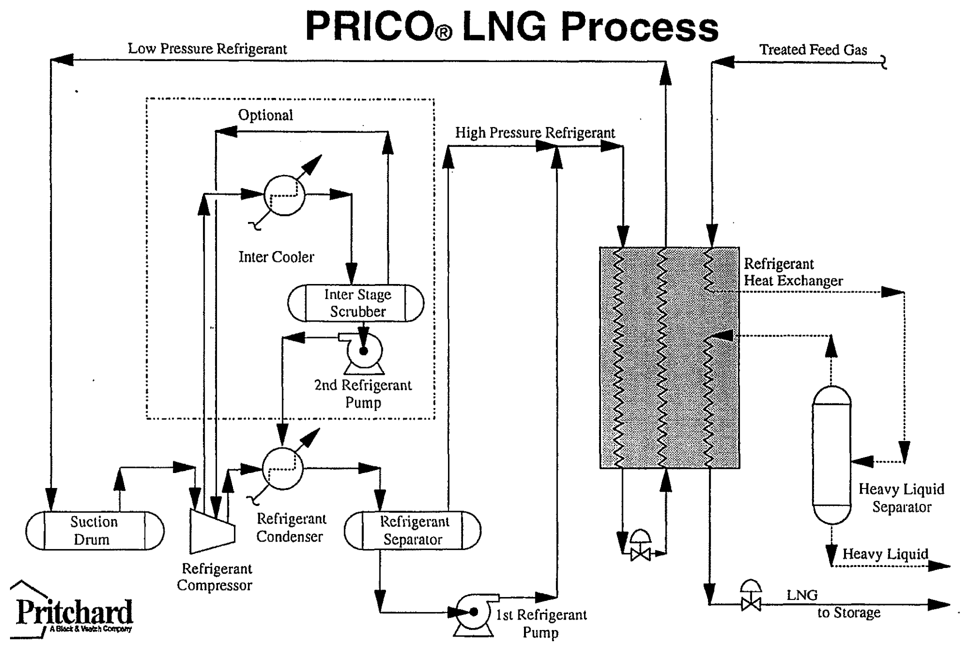


Figure 2.5: Flowsheet of a typical PRICO process [8].

2.3 Optimization

In everyday life we optimize. Seeking to find the best solution to any of the numerous little problems encountered, e.g. finding the shortest route home from work to save time.

A nonlinear minimization problem is defined

$$\begin{aligned} \min_{u} \quad & J(x,u,d) \\ \text{subject to} \quad & f(c,u,d) = 0 \\ & c(x,u,d) \leq 0 \end{aligned} \qquad (2.5)$$

J is the cost function, $f(x,u,d)$ is the process model equations and $c(x,u,d)$ is the process constraints. x are the internal variables, u are the manipulated variables and d are the disturbances.

For a minimization problem as given in Equation 2.5 a cost function is minimized over the expected disturbances while satisfying the process constraints [9]. After formulating a model an optimization algorithm is used to find a solution. When the solution is found it may be checked by using Optimality conditions.

2.4 Self-optimizing control

We always want a system to operate as close to optimum as possible. However, a system will always be affected by changes. Usually on-line optimization is done for every disturbance. A simpler strategy to solve this kind of problem is to somehow get the system to "self-optimize" for different disturbances, instead of doing the normal on-line optimization. This is the basic idea behind the self-optimizing control.

Skogestad gives the following explanation for self-optimizing control [10]:

> *"Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to reoptimize when disturbances occur)."*

By defining the loss from the previously mentioned cost function as the difference between actual operating costs for the controlled system and the optimal operational costs we get

$$L(u,d) = J(u,d) - J_{opt}(d) \tag{2.6}$$

where L is the loss, J is the actual costs and J_{opt} is the actual costs at the optimum. Obviously, we want a small difference between J and J_{opt}. This is done in practice by using the degrees of freedom for the system to control the optimal active constraints and using the remaning degrees of freedom to keep the ”self-optimizing" variables at a constant setpoint.

Different strategies are employed to select good controlled variables. Intuitively, a controlled variable need to be insensitive to disturbances at its optimal point so that the optimum does not shift for disturbances. Also, the optimum should be flat and therefore avoid problems with implementation errors.

3 Model

This chapter deals with the model used for the simulation and optimization. The model is described in detail along with the building of the model.

3.1 Process description

Originally developed by the Pritchard Group the PRICO process is patented by Black and Veatch. They deliver a wide range of liquefaction soultion for baseload, peak shaving and LNG storage facilities to floating liquefaction solutions [8].

The setup of the PRICO process is simple and easily comparable to the vapor-compression cycle. Cooled, high pressure refrigerant is supplied to a heat exchanger flowing downward through the whole exchanger being condensed and sub-cooled. A hydro turbine expands the refrigerant removing some of its energy. The choke valve acts only as a dummy valve. The refrigerant flows back through the heat exchanger from the bottom to the top cooling both the high pressure refrigerant and feed while being vaporized and superheated. The low pressure refrigerant is the compressed. This can be done in either a two-stage compression configuration as shown in previously in Figure 2.2 or a single stage. The flowsheet is given in Figure 3.1.

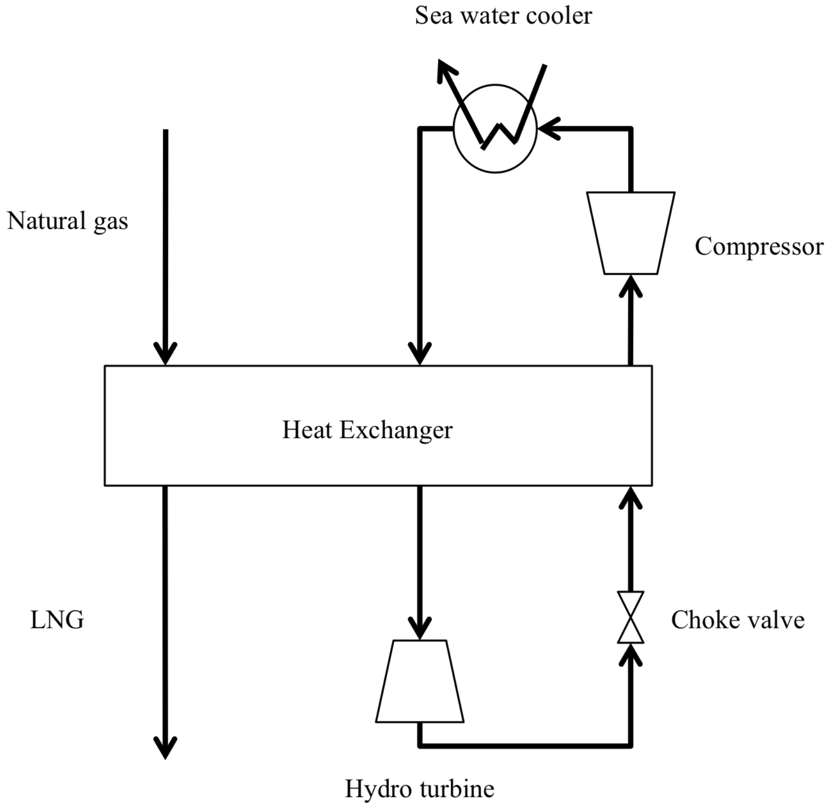


Figure 3.1: Flowsheet of the PRICO process used in the model.

The model varies from the previous explained PRICO process by having a hydro turbine. The hydro turbine was included to utilize the energy when decreasing the pressure of the refrigerant.

The use of a hydro turbine instead of a choke valve improves the efficiency of the refrigerant cycle [6]. This is easily shown in the pressure enthalpy diagram in Figure 3.2. i) represents the expansion path for a choke valve while ii) represents that of a hydro turbine. iii) represents the expansion path for a combination of a liquid turbine and a choke valve which is not considered here.

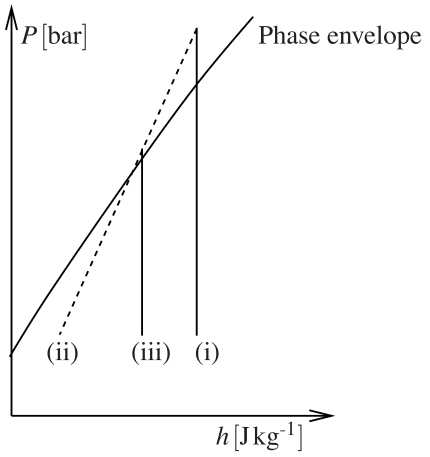


Figure 3.2: Pressure enthalpy diagram of expansion paths

Removal of contaminants and heavy hydrocarbons from the natural gas before liquefaction is not considered in the model. Also, the pressure reduction of the LNG to atmospheric pressure is not modelled.

3.2 Modelling in Unisim Design

The PRICO refrigeration cycle was modelled in the simulation software Unisim Design R400 from Honeywell. Steady-state design mode was used for the model. The Soave-Redlich-Kwong (SRK) Equation of state was used as the thermodynamic fluid package.

The basis for the model is the thesis of Jensen [6] and of Jacobsen [9].

The model consists of a mixed refrigerant loop in contact with the feed stream of natural gas through an LNG exchanger. The refrigerant, MR1, enters the LNG exchanger (HEX) where it is cooled by the cold side of the LNG exchanger. Energy is taken from the refrigerant by sending it through a hydro turbine to decrease the pressure. The refrigerant goes back into the LNG exchanger cooling the natural gas and the high pressure refrigerant. After compression the temperature is lowered using a seawater cooler. The model is given in Figure 3.3.

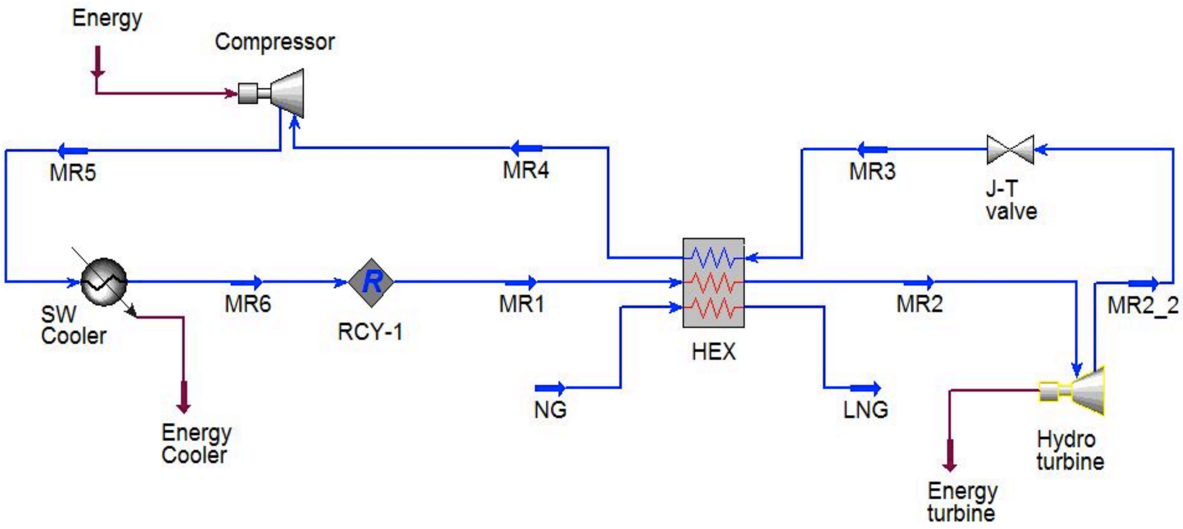


Figure 3.3: Model of PRICO process in Unisim Design

The components used for the refrigerant and natural gas cycles are given in Table 3.1.

Table 3.1: Components in refrigerant and natural gas.

	Natural gas - LNG	Mixed refrigerant cycle
Methane	0.327	0.897
Ethane	0.343	0.055
Propane	-	0.001
n-butane	0.233	0.028
Nitrogen	0.097	0.019

The process parameters specified by Jensen and Skogestad and Jacobsen is given in Table 3.2.

Table 3.2: Process parameters for model.

Parameter	Value
Flow natural gas	4.214 [kgmole/s]
Temperature natural gas	30 [^{o}C]
Temperature LNG	-157 [^{o}C]
Pressure natural gas	40 [bar]
Cooling of refrigerant in SW cooler	30 [^{o}C]
Pressure drop natural gas stream	5 [bar]
Pressure drop hot refrigerant stream	4 [bar]
Pressure drop cold refrigerant stream	1 [bar]
Pressure drop SW cooler	0.1 [bar]

LNG Exchanger

Unisim design uses a plate-fin heat exhanger as a model for an LNG exhanger [11]. The LNG exchanger (HEX) allow multiple inlet and outlet streams which is required in the PRICO process. To solve the energy balance Unisim Design employs an iterative approach. The tolerance for the LNG exchanger was set to 10^{-4} to make to model converge. It is also possible to add constraints for the energy balance.

Compressor

For the compressor in the model the adiabatic efficiency along with the outlet pressure is calculated from the compressor characteristic curves [6]. The pressure ratio is given in Equation 3.1.

$$Pr = Pr_0 + H\left(1 + \frac{3}{2}\left(\frac{mr}{W} - 1\right) - \frac{1}{2}\left(\frac{mr}{W} - 1\right)^3\right) \tag{3.1}$$

where $Pr = P_2 / P_1$ is the pressure ratio over the compressor, Pr_0 is the pressure ratio delivered at zero flow, *H* is the semi-height and *W* is the semi-width of the characteristic curve, and *mr* is the reduced mass flow rate. The equations for *H*, *W*, *mr*, are given Equation 3.2, 3.3 and 3.4 respectively. The equation for the reduced speed Nr is given in Equation 3.5.

$$H = H_0 - 1.2\left(H_0 + \frac{Pr_0}{2} - 1\right)\cdot(1 - Nr) \tag{3.2}$$

$$W = W_0(Nr)^{1/3} \tag{3.3}$$

$$mr = \frac{m\sqrt{\frac{R}{MW}T_1}}{D^2 P_1} \tag{3.4}$$

$$Nr = \frac{ND}{N_0}\frac{1}{\sqrt{\frac{R}{MW}T_1}} \tag{3.5}$$

where H_0 and W_0 are the nominal values for the semi-height and the semi-width, *m* is the mass flow rate, *R* is the gas constant, *MW* is the molecular weight, *D* is the compressor wheel diameter, *Nr* is the reduced compressor speed, N_0 is the nominal percentage of the compressor speed, *N* is the compressor speed percentage and P_1 and T_1 are the inlet pressure and temperature.

Jensen [6] proposed the following equation for the isentropic efficiency of the compressor.

$$\eta = \eta_0\left(\left(1 - \left(\frac{H - H_0}{H_0}\right)\right)^2 - 100 + (mr - 2W)^2\right) \tag{3.6}$$

η and η_0 being the isentropic efficiceny and the nominal isentropic efficiency.

The nominal values used for calculating the characteristic compressor curves are given in Table 3.3.

Table 3.3: Parameters for the characteristic curves

Parameter	**Value**
N_0	0.8
H_0	18.125
W_0	0.0698
T_1	284 [^{o}C]
MW	0.032 [kg/kgmole]
P_1	450 000 [Pa]
Pr_0	-29
η_0	0.828
D	1.7 [m]

Notice that T_1 and P_1 are set as nominal values. This is to simplificy the calculations done by Matlab for Unisim Design. Preliminary tests done without these nominal values shows that Unisim having troubles converging. This will be discussed later in this report.

The compressor characteristic curves also makes a model for the compressor surge limit, defining that the reduced mass flow must be at least $2 \cdot W$, for not having surge in the compressor.

4. Optimization

This chapter describes the objective function and constraints for the process. The degrees of freedom are determined and the optimization in Matlab is described.

4.1 Objective function and constraints

The first step in the top-down plantwide control procedure [10], is defining the objective function and the constraints. For optimal operation the capital costs are not considered, and only the operation costs are taken into account. The objective function for the modelled PRICO process takes the form

$$J_{operation} = p_{W_s} W_s + p_{SW} W_{SW} - p_{turbine} W_{turbine} + p_{feed} \dot{m}_{feed} - p_{prod} \dot{m}_{prod} \tag{4.1}$$

By making the assumptions that there are no costs regarding the seawater cooling, that the energy from the hydro turbine is negligible small compared to the compressor work and that the rate of production equals the rate of the feed, the objective function reduces to only a function of the compressor work.

$$J_{operation} = p_{W_s} W_s \tag{4.2}$$

The objective function is minimized with respect to the process variables in the Unisim model.

Different constraints are defined for the objective function to make sure the operation is both feasible and realistic. The heat exchanger is constrained by a minimum temperature approach $\Delta T_{\min}$, which is a minimum temperature difference between the hot or cold passes and the opposite composite curve. To avoid surge in the compressor the reduced mass flowrate, *mr*, was constrained to a minimum of 2 times the semi-width, *W* of the compressor. The *UA* used for the heat exchanger is taken from Jensen [6] and used as an equality constraint. The equality and the inequality constraints are given in Table 4.1.

Table 4.1: Constraints for the optimization

Variable	Constraint
$\Delta T_{\min}$	$> 1.2\ ^{o}\mathrm{C}$
mr	$> 2 \cdot W$
UA	$= 5.5 \cdot 10^{4}\ \mathrm{kW/^{o}C}$

4.2 Degrees of freedom analysis

The second step in the plantwide procedure [10] is identifying the degrees of freedom. For a given process the degrees of freedom must be specified to completely describe it. N_{SS} is the number of steady state degrees of freedom. The number is obtained by subtracting the number of degrees of freedom with no steady state effect with the number of dynamic manipulated variables.

$$N_{SS} = N_{MV} - N_0 \tag{4.3}$$

N_{MV} is the number of manipulated variables and N_0 represent the degrees of freedom with no steady state effect. N_0 can be manipulated variables with no steady state effect on operation and variables with no steady state effect that need to be controlled [12].

N_{SS} is also important because it determines the number of steady state controlled variables that need to be selected. The potential steady state degrees for the PRICO process based on Jensen and Skogestad table [12] is given in Table 4.2.

Table 4.2: Potential degrees of freedom for the PRICO process

Process unit	**Potential degree of freedom**
Feed	1
"Active charge"	1
Compressor	1
Hydro turbine	1
SW cooler	1
Composition of refrigerant	4
Heat exchanger	1
Total potential degrees of freedom	**10**

As the choke valve is just a dummy valve in the model it will not have a degree of freedom. For the optimization the process feed is given and is eliminated as a degree of freedom. The SW cooling was set to give an inlet refrigerant stream to the heat exchanger of 30 ^{o}C and considered a disturbance. The 4 degrees of freedom from the refrigerant composition will also be eliminated since we assume that the composition cannot be changed during operation. This gives a total of 4 degrees left for the optimization.

The 4 process variables for the optimization is

- Mass flow of refrigerant
- Temperature at MR2
- Pressure at MR3
- Rotational speed of compressor

4.3 Optimization in Matlab

The optimization of the PRICO process was lifted from Unisim Design and done in Matlab due to previous problems using built-in optimizers in simulation programs. Interfacing Unsim Design and Matlab was done creating an ActiveX server to run the simulation from Matlab.

The minimization function *fmincon* in Matlab was used for the optimization. *fmincon* is used for finding a minimum to a nonlinear constraind multivariable function. Both linear and non-linear constraints can be used. Different algorithms can be employed for the *fmincon* function. This project uses the *active-set* algorithm mainly due to good initial results. It uses a sequential quadratic programming (SQP) method and estimates the Hessian of the Lagrangian at each iteration.

The *fmincon* function is represented in Equation 4.4.

$$x = fmincon(@myfun, x0, A, b, Aeq, beq, lb, ub, @mycon) \tag{4.4}$$

Here the function *mynfun* is the objective function, *x0* is the initial estimate of *x,* $A \cdot x \leq b$ and $Aeq \cdot x = beq$ is the linear equality and inequality constraints. The lower and upper bounds are *lb* and *ub,* respectively. The function *mycon* calculates the non-linear constraints.

Scaling

All the variables, the bounds, and the objective function were scaled. The SQP method is said to be scaling invariant. However, scaling will result in smaller condition number for the Hessian and an improved performance of the algorithm [13]. Also, not using scaling may result in large rounding errors, decreasing the performance of the algorithm.

Penalties

The optimization in Matlab is dependant on the model in Unisim Design being able to converge. The objective function and constraints were given penalties when the model was not able to converge or the solution area became infeasible. Penalties were also given to the isentropic efficiency and the pressure from the compressor for negative values.

4.4 Optimization Results

The model in Unisim Design was optimized in Matlab using optimization routine in Appendix C. The results are shown in Table 4.3 along with the calculated values for the compressor efficiency and the pressure out of the compressor. The values from the optimal nominal cases of Jensen [6] and Jacobsen [9] are also shown for comparison. The values for the pressure after compression and the isentropic efficiency for the compressor are calculated from the rotational speed percentage, *N*. For detailed information from the optimization, see Appendix D.

Table 4.3: Results for the optimization in Matlab.

	Optimized value	Jacobsen values	Jensen values	Units
Objective function	113096	116000	120000	kW
Manipulated variable				
Mass flow, MR1	623.89	539.1	595	kg/s
Rotational speed compressor	0.651			-
Temperature before expansion, MR2	-156	-163.9	-157	^oC
Pressure low side	5.338	5.552	10.29	bar
Compressor efficiency	82.12	81.8	82.8	-
Pressure after compression	27.33	30.2	30	bar

The optimal values for the constraints are given in Table 4.4. Both inequality constraints are active.

Table 4.4: Constraint values at optimum

Constraint	Value
$\Delta T_{\min}$	= 1.2
mr	= 0.1303 = 2W
UA	$= 5.5 \cdot 10^4$

The characteristic compressor curves for the optimized model are given in Figure 4.1. The ten curves represent the different rotational speed percentage, ranging from 0 – 100%. The window of operation is to the right for curve peaks that has been highlighted in red. These are the surge limits.

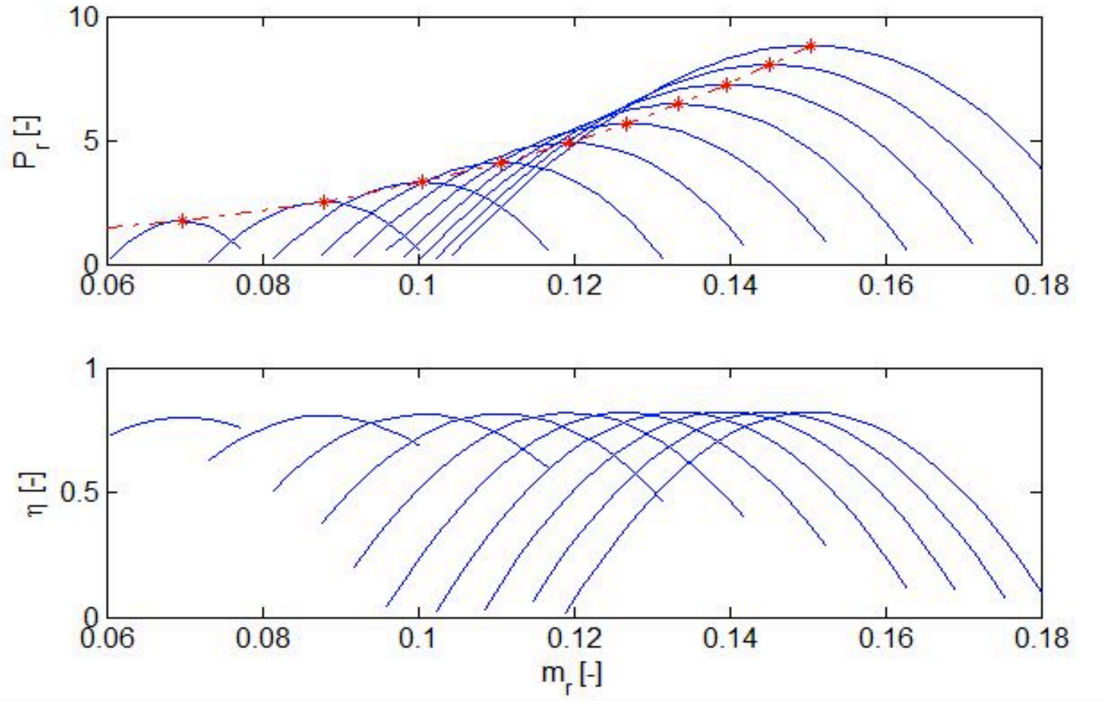


Figure 4.1: Characteristic compressor curves for the optimized model

4.4 Self-optimizing control

For operations mode both the active equality constraint for the *UA* value and the $\Delta T_{\min}$ constraint are not valid. The area of the heat exchanger is set. Also the $\Delta T_{\min}$ is not set in operations mode and is free to vary.

This results in one active constraint from the optimization to be controlled, the compressor surge, $\Delta M_{surge} = 0$. However, we also need to control the LNG temperature. In the optimization this was set to -157 ^{o}C but during operation this becomes an active constraint. This leaves two degrees of freedom left. To find good candidates for self-optimizing variables one may use the maximum gain method [14]. Typical measurements for a refrigeration cycle to be investigated are given in Table 4.5 [16].

Table 4.5: List of possible controlled variables for selecting controlled outputs [16]

Measurement
Condenser pressure
Evaporator pressure
Temperature at condenser exit
Degree of subcooling
Choke valve opening/hydro turbine power
Liquid level evaporator
Liquid level condenser

The gains of the potential controlled variables should be calculated numerically by applying small perturbations in the manipulated variables, while keeping the active constraints constant by manipulation of the other inputs. The process should be optimized for each of the disturbances to find the optimal ranges for the controlled variables. As mentioned earlier this will be the cooling of the refrigerant, the process flow rate and the composition of the feed. One should also consider the active constraints as disturbances. The scaled gains are obtained by dividing the nominal optimal gains with the span of each controlled variable + implementation error. The controlled variables with the largest scaled gain will preferably be the chosen “self-optimizing” variables that should be controlled at its nominal optimum.

5. Discussion

This section discusses the problems encountered during this project. Alternatives and possible solutions will be proposed.

The PRICO process model was built in Unisim Design based on previous models. For design purposes the software has proven itself as a good and user-friendly software. The model used is a simplified version of the PRICO process. The expansion of LNG to atmospheric pressures was not considered for the model. To make the model more realistic the compression could be modelled as a two-stage compression with intercooling to increase the efficiency. However, this study aimed at optimizing a simple refrigeration process due to previous difficulties, and complicating it with additional sections will just make the optimization harder.

The process values used for the model were taken from previous studies and have not been checked against industrial standards today. More effort may have been used to find better values for the model making more realistic. As mentioned by Jensen in his thesis [6], the compressor characteristics is not fitted to a real compressor. Thus, it is unknown if a real compressor would follow the behaviour of the model.

A hydro turbine was used for the expansion of the refrigerant. This was done to improve the efficiency of the cycle. A problem associated with using a turbine instead of a choke valve is wear of the turbine due to some of the liquid becoming vapor when expanded. In the model the choke valve was handled as a dummy valve. A possibility for the expansion is to use both the hydro turbine and the choke valve. The turbine expands the liquid to the saturation pressure and the choke valve expands the liquid to a two phase flow. This addition will not have any effect on the compressor power usage, as the pressure of the refrigerant after the evaporation will not change.

The tolerance of the LNG exchanger was loosened to make sure the model converged. The *UA* value may be specified in the Specifications section for the LNG exchanger in Unisim Design. However, the optimization algorithm has troubles finding any solutions for this case. Therefore the *UA* were used as a nonlinear equality constraint for the optimization, which proved a better strategy.

The compressor characteristics links the mass flow rate, pressure ratio, efficiency and rotational speed. Matlab calculates the efficiency and pressure out of the compressor and assign these values to the model for every iteration step. By doing calculations like these outside the simulation software the user has a better overview of the calculations when a problem is encountered, as it normally is. Having this transparency helps problem solving.

Problems with the optimization algorithm being stuck at a single point have also been encountered. To counteract these problems a solution is to have the initial starting points closer to the optimum. This requires knowledge of the optimization problem, which is not always the case. E.g “black box” optimization as in this project.

For the compressor characteristics, nominal values for both the pressure and temperature into the compressor were used instead of the calculated values from the simulation. The efficiency and pressure is then only a function of the refrigerant flow rate and the rotational speed percentage, all other parameters being fixed. This lightens the optimization due less interaction between the variables.

The optimization was carried out in Matlab by interfacing it with Unisim Design through an ActiveX server. This implementation is fairly easy and gives rise to numerous usages for connecting Matlab with simulation software. The *fmincon* function was used to minimize the objective function. Other optimization functions were considered but due to implementation problems these were not pursued further.

The main problem doing the optimization from Matlab was the Unisim Design model not converging. Scaling and the use of penalties for the objective function have been used extensively in the optimization to overcome Unisim not returning variables to Matlab. This happens when the optimization algorithm assigns values to the process variables in Unisim for a non-feasible region. By identifying these regions; large penalties may be assigned to the objective function or the constraints, continuing the optimization. Scaling was done for the optimization. There are uncertainties regarding the effect of it as the SQP method is said to be scaling invariant even though the optimization went more smoothly when scaling. Scaling and the use of penalties are recommended, as it will produce a more robust optimization procedure.

The active-set algorithm was used for the minimization. It recognises which of the process variables that produces the largest change in the objective function and uses that variable for closing in on the minimum. A problem with the active-set algorithm is that it may break the bounds trying to minimize objective function. This has been a problem especially when the initial points were far from the solution or when using large bounds. As the *fmincon* function can use different algorithms more time may have been used testing the algorithms with respect to robustness.

The maximum gain rule would probably give good estimates for the controlled variables. A more tedious method would be the "brute-force" evaluation. The objective function value is checked systematically by keeping a candidate set of controlled variable constant for given disturbances. Even with a couple of potential controlled variables this would be time consuming.

To simplify the problem one can argue that the temperature of the refrigerant before expansion in the turbine is an active constraint. For the nominal optimization the optimal value is -156 ^{o}C. If the temperature rises the model will not converge and therefore needing to be controlled. Only 1 degree of freedom will be left to determine a self-optimizing control structure, making it easier.

The results from the nominal optimization gave better values for the compressor power than previous case studies of the PRICO process [6][9]. As seen in Table 4.3 the process variables differ significantly, thus giving rise to the assumption that the objective function may have multiple local minimums. This was not investigated further as the optimization result was found to be the same for different initial points.

6. Conclusion and further work

Conclusion

The PRICO liquefaction process was modelled in Unisim Design. The optimization was lifted out of Unisim and done in Matlab through an ActiveX server. By doing the optimization in Matlab a solution was found for the nominal case.

A considerably amount of problems were encountered when doing the optimization. By using scales, and penalties for the non-feasible solutions the optimization algorithm becomes more robust.

A self-optimizing control structure has not been proposed for the process. However, central issues and possible approaches to finding the self-optimizing variables have been presented and discussed.

Previous problem in optimizing LNG processes prior to this project is in agreement with this project, concluding that Unisim Design is not an accurate tool for LNG simulations and optimizations.

Further work

The optimization algorithm from this project should be used to find a self-optimizing control structure for the PRICO process modelled. The proposed approach given in this report may be used. Also, the use of the Exact local method or Null space method may be considered.

An alternative approach may be to use dynamic simulation for finding the steady state solution.

More work may be done in making the model more accurate and robust. Other simulation software may be used that give a better transparency of the model.

7. References

[1] *Are we entering a golden age of gas?* Special Report, Interantional Energy Agency, 2011

[2] *World Energy Outlook*. International Energy Agency, 2010

[3] Lom, W. L., *Liquefied natural gas*, Applied Science Publishers LTD, 1974

[4] The Global Liquefied Natural Gas Market: Status and Outlook http://www.eia.gov/oiaf/analysispaper/global/lngindustry.html date:15.12.2011

[5] Dossat, R. J., Horan, T. J., *Principles of Refrigeration*, 5th Ed. Prentice Hall, 2002.

[6] Jensen, J. B., *Optimal operation of cyclic processes – Application to LNG processes.* Ph.D thesis , NTNU, 2008

[7] Chieu, C., *History of the development of LNG Technology*, AiCHe Annual Conference, Philadelphia, 2008

[8] Price, B., Mortko, R., *PRICO – a simple, flexible proven approach to natural gas liquefaction.* In: GASTECH, LNG, Natural Gas, LPG international conference, Vienna, 1998

[9] Jacobsen, M. G., *Identifying active constraint regions for optimal operation of process plants,* Ph.D thesis, NTNU, 2011

[10] Skogestad, S., *Plantwide control: The search for a self-optimizing structure,* J. Proc. Control, 10, 487-507, 2000

[11] Unisim Operations Guide, Honewell 2007

[12] Jensen, J. B., Skogestad, S., *``Steady-State Operational Degrees of Freedom with Application to Refrigeration Cycles",* Ind.Eng.Chem.Res, 48 (14), 6652-6659, 2009

[13] Lid, T., *Data reconciliation and optimal operation –With applications to refinery processes,* Ph.D. thesis, NTNU, 2007

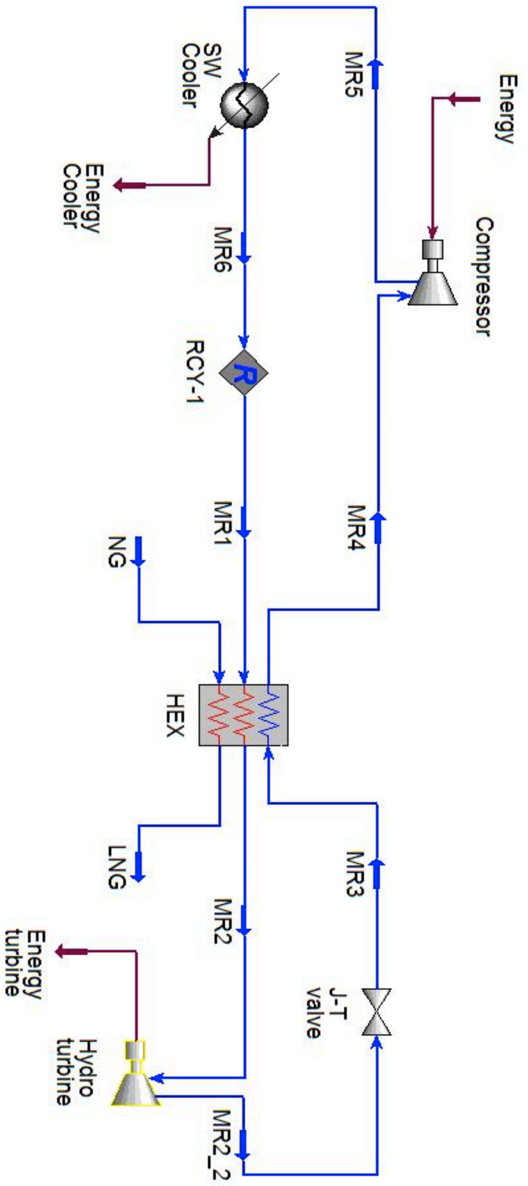
[14] Halvorsen, I. J., Skogestad, S., Morud, J. C., Alstad, V., *Optimal selection of controlled variables',* Ind. Eng. Chem. Res., **42** (14), 3273-3284 (2003)

[15] Shukuri, T., *LNG technology selection.* Foster Wheeler, 2008 http://www.fwc.com/publications/tech_papers/files/TariqLNG.pdf date: 15.12.2011

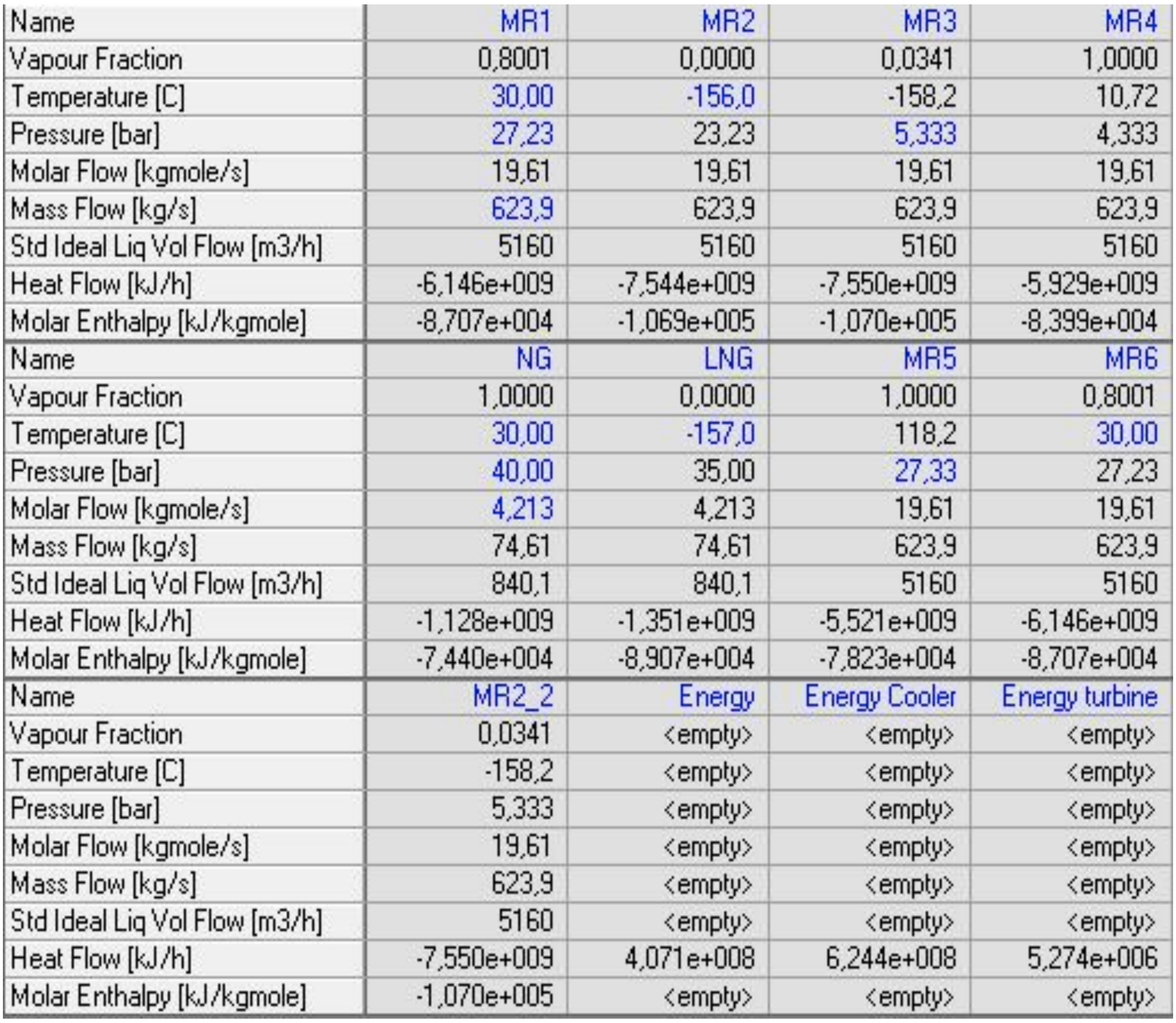
[16] Skogestad, S., Postlethwaite, I., *Multivariable feedback control*, Second edition, John Wiley and Sons Ltd., 2005

APPENDIX

A. Unisim Design Flowsheet



B. Unisim Design Workbook



Name	MR1	MR2	MR3	MR4
Vapour Fraction	0,8001	0,0000	0,0341	1,0000
Temperature [C]	30,00	-156,0	-158,2	10,72
Pressure [bar]	27,23	23,23	5,333	4,333
Molar Flow [kgmole/s]	19,61	19,61	19,61	19,61
Mass Flow [kg/s]	623,9	623,9	623,9	623,9
Std Ideal Liq Vol Flow [m3/h]	5160	5160	5160	5160
Heat Flow [kJ/h]	-6,146e+009	-7,544e+009	-7,550e+009	-5,929e+009
Molar Enthalpy [kJ/kgmole]	-8,707e+004	-1,069e+005	-1,070e+005	-8,399e+004
Name	NG	LNG	MR5	MR6
Vapour Fraction	1,0000	0,0000	1,0000	0,8001
Temperature [C]	30,00	-157,0	118,2	30,00
Pressure [bar]	40,00	35,00	27,33	27,23
Molar Flow [kgmole/s]	4,213	4,213	19,61	19,61
Mass Flow [kg/s]	74,61	74,61	623,9	623,9
Std Ideal Liq Vol Flow [m3/h]	840,1	840,1	5160	5160
Heat Flow [kJ/h]	-1,128e+009	-1,351e+009	-5,521e+009	-6,146e+009
Molar Enthalpy [kJ/kgmole]	-7,440e+004	-8,907e+004	-7,823e+004	-8,707e+004
Name	MR2_2	Energy	Energy Cooler	Energy turbine
Vapour Fraction	0,0341	<empty>	<empty>	<empty>
Temperature [C]	-158,2	<empty>	<empty>	<empty>
Pressure [bar]	5,333	<empty>	<empty>	<empty>
Molar Flow [kgmole/s]	19,61	<empty>	<empty>	<empty>
Mass Flow [kg/s]	623,9	<empty>	<empty>	<empty>
Std Ideal Liq Vol Flow [m3/h]	5160	<empty>	<empty>	<empty>
Heat Flow [kJ/h]	-7,550e+009	4,071e+008	6,244e+008	5,274e+006
Molar Enthalpy [kJ/kgmole]	-1,070e+005	<empty>	<empty>	<empty>

C. Matlab Optimization

Main script – prico_3.m

```
clear all
clc

h = actxserver('UnisimDesign.Application'); %creating active x
server, a link between matlab and unisim
uniCase = h.ActiveDocument;     %entering document in unisim
f = uniCase.Flowsheet;         %entering flowsheet in unisim
sol = uniCase.Solver;        %entering solver in unisim

%Make unisim converge before optimization

sol.CanSolve = 0;
f.MaterialStreams.Item('MR1').MassFlowValue = 631.7;
f.MaterialStreams.Item('MR2').TemperatureValue = -156.3;
f.MaterialStreams.Item('MR3').PressureValue = 569.9;
f.MaterialStreams.Item('MR5').PressureValue = 2886;
f.MaterialStreams.Item('MR1').PressureValue = 2886;

sol.CanSolve = 1;

global LB UB

LB = [400 -165  300 0.4];         %lower and upper bounds
UB = [1000 -155 900 1.0];

x0_nons = zeros(4,1);
x0 = zeros(4,1);
X = zeros(4,1);

x0_nons(1) = 620.7;          %initial mass flow MR before HEX
x0_nons(2) = -156.3;         %initial temperature in MR before
valve
x0_nons(3) = 569.9;          %initial pressure in MR after valve
x0_nons(4) = 0.679;

scaled = nonscaled_3(x0_nons);      %scaled variables
x0 = scaled;

lb = [0  0  0  0];                  % scaled lower and upper bounds
scaled
ub = [1  1  1  1];


%Optimization with nonlinear constraints

options = optimset('Algorithm','active-set','TolCon',1e-
4,'TolFun',1e-
4,'Display','iter','MaxIter',125,'MaxFunEvals',1e4,'FinDiffType','ce
ntral'); % Options for fmincon
```

```
[x,fval,exitflag,lambda,grad,hessian] =
fmincon(@objective_3,x0,[],[],[],[],lb,ub,@mycon_3,options);

for i = 1:length(x)
    X(i) = (x(i)*(UB(i)-LB(i)))+LB(i);  %scale back variables and
function value
end

FVAL = fval*1e6;
```

Objective function – objective_3.m

```
function sJ = objective_3(x)

h = actxserver('UnisimDesign.Application');     %creating active x
server, a link between matlab and unisim
uniCase = h.ActiveDocument;      %entering document in unisim
f = uniCase.Flowsheet;        %entering flowsheet in unisim
sol = uniCase.Solver;    %entering solver in unisim

sol.CanSolve = 0;

xscaled = x;

notscaled = scaleback_3(xscaled);

f.MaterialStreams.Item('MR1').MassFlowValue = notscaled(1); %Mass
flow, M1
f.MaterialStreams.Item('MR2').TemperatureValue = notscaled(2);
%Temperature after before turbine, MR2
f.MaterialStreams.Item('MR3').PressureValue = notscaled(3); %
Pressure after choke valve
N = notscaled(4);   % compressor speed

%Calculating P2 and compressor efficiency

m = notscaled(1);
N0 = 0.8;
P1 = 450000; %Pa
T1 = 284; %K
W0 = 0.0698;
MW = 0.032; %kg/mol
Rhat = 8.314/MW;
D = 1.7; %meters
H0 = 18.125;
Pr0 = -29;
eta0 = 0.822;

W = W0*(N/N0)^(1/3);
mr = m*sqrt(Rhat*T1)/(D^2*P1);
H = H0-1.2*(H0+Pr0/2-1)*(1-N/N0);
Pr = Pr0+H*(1+(3/2)*(mr/W-1)-0.5*(mr/W-1)^3);
P2 = Pr*P1;
eta = eta0*((1-((H-H0)/H0)^2)-1000*(mr-2*W)^2);

if P2<5.5e+5 || eta<0.01

    m = 550 ; N = 0.5;
    W = W0*(N/N0)^(1/3);

    mr = m*sqrt(Rhat*T1)/(D^2*P1);
    H = H0-1.2*(H0+Pr0/2-1)*(1-N/N0);
    Pr = Pr0+H*(1+(3/2)*(mr/W-1)-0.5*(mr/W-1)^3);
    P2 = Pr*P1;
    eta2 = eta0*((1-((H-H0)/H0)^2)-1000*(mr-2*W)^2);
    f.MaterialStreams.Item('MR5').PressureValue = P2*1e-3;
    f.Operations.Item('Compressor').CompAdiabaticEffValue =
eta2*100;

    sol.CanSolve = 1;

    sJ = 100;

else
    f.MaterialStreams.Item('MR5').PressureValue = P2*1e-3;
    f.Operations.Item('Compressor').CompAdiabaticEffValue = eta*100;

    sol.CanSolve = 1;

    if f.EnergyStreams.Item('Energy').Power.IsKnown == 1 &&
f.MaterialStreams.Item('MR4').VapourFractionValue == 1

            J = f.Operations.Item('SPRDSHT-1').Cell('A2').CellValue;
            sJ = J*1e-6;
        else
            sJ = 100;
    end
end
```

Constraints function – mycon_3.m

```
function [c ceq] = mycon_3(x)


h = actxserver('UnisimDesign.Application'); %creating active x
server, a link between matlab and unisim
uniCase = h.ActiveDocument;     %entering document in unisim
f = uniCase.Flowsheet;       %entering flowsheet in unisim
sol = uniCase.Solver;   %entering solver in unisim


sol.CanSolve = 0;                           %stops solver

xscaled = x;

notscaled = scaleback_3(xscaled);

f.MaterialStreams.Item('MR1').MassFlowValue = notscaled(1); %Mass
flow, MR1
f.MaterialStreams.Item('MR2').TemperatureValue = notscaled(2);
%Temp, MR2
f.MaterialStreams.Item('MR3').PressureValue = notscaled(3);
%Pressure after choke valve
N = notscaled(4);   %compressor speed

m = notscaled(1);
N0 = 0.8;
P1 = 450000; %Pa
T1 = 284; %K
W0 = 0.0698;
MW = 0.032; %kg/mol
Rhat = 8.314/MW;
D = 1.7; %meters
H0 = 18.125;
Pr0 = -29;
eta0 = 0.822;

W = W0*(N/N0)^(1/3);
 mr = m*sqrt(Rhat*T1)/(D^2*P1);
H = H0-1.2*(H0+Pr0/2-1)*(1-N/N0);
Pr = Pr0+H*(1+(3/2)*(mr/W-1)-0.5*(mr/W-1)^3);
P2 = Pr*P1;
eta = eta0*((1-((H-H0)/H0)^2)-1000*(mr-2*W)^2);

if P2<5.5e+5 || eta<0.01
    m = 550; N = 0.5;
    W = W0*(N/N0)^(1/3);

    mr = m*sqrt(Rhat*T1)/(D^2*P1);
    H = H0-1.2*(H0+Pr0/2-1)*(1-N/N0);
    Pr = Pr0+H*(1+(3/2)*(mr/W-1)-0.5*(mr/W-1)^3);
    P2 = Pr*P1;
    eta = eta0*((1-((H-H0)/H0)^2)-1000*(mr-2*W)^2);
    f.MaterialStreams.Item('MR5').PressureValue = P2*1e-3;
    f.Operations.Item('Compressor').CompAdiabaticEffValue = eta*100;
else
    f.MaterialStreams.Item('MR5').PressureValue = P2*1e-3;
    f.Operations.Item('Compressor').CompAdiabaticEffValue = eta*100;
end

sol.CanSolve = 1;

minapp = f.Operations.Item('HEX').MinApproach.Value;
UA = 5.5e+4;


if f.Operations.Item('Compressor').Energy.IsKnown == 1 &&
f.Operations.Item('HEX').UA.IsKnown == 1

res =
f.Operations.Item('HEX').Specifications.Item('UA_global').CurrentVal
ue;


 ress = (UA-res)*1e-4;

c1 = 1.2-minapp;
c2 = -mr+2*W;
c=[c1 c2];
ceq = ress;

else
    c=ones(2,1);
    ceq = 1;

end
```

Scaling fuctions – scaleback_3.m

```
function notscaled = scaleback_3(xscaled)

global LB UB

notscaled = zeros(4,1);

for i = 1:length(xscaled)
    notscaled(i) = xscaled(i)*(UB(i)-LB(i))+LB(i);
end
return
```

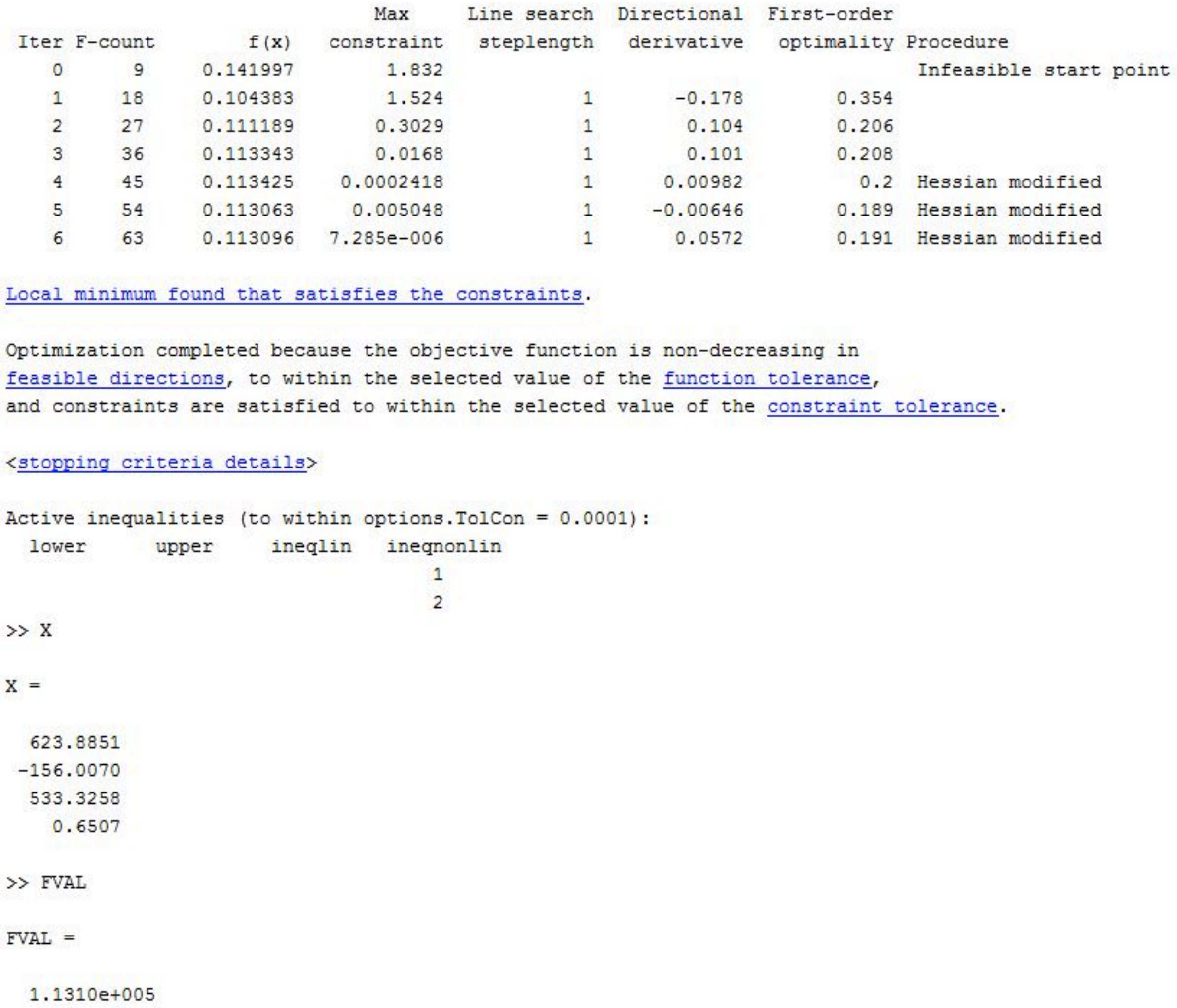
Scaling functions – nonscaled_3.m

```
function scaled = nonscaled_3(x0_nons)

global LB UB

scaled = zeros(4,1);

for i = 1:length(x0_nons)
    scaled(i) = (x0_nons(i)-LB(i))/(UB(i)-LB(i));
end
return
```

Characteristic compressor curves – Compressor_maps-2.m

```
%Compressor characteristic maps based on Jensen thesis
clear all;
clc;

%
%Variables
%
%Rotational speed [sec^-1]
N=0.6507;
N0=0.8;
%Mass flowrate [kg/s]
m=623.89;
%Molecular weight [kg/mol]
MW=0.032;
%Inlet temperature [K]
T1=284;
%Inlet pressure [kg/(m s^2)]
P1= 450000;
%Charactestic length of the compressor (wheel diameter) [m]
D=1.7;
%Semi-height
H0=18.125;
%Semi-width
W0=0.0698;
%Efficiency
eta0=0.822;
%Pressure ratio
Pr0=-29;
%Specific gas constant
Rhat=8.314472/MW;

X=linspace(0,1,11);
M=linspace(0,1000,101);
Pr=zeros(11,101);
eta=zeros(11,101);
Mr=zeros(1,101);
SurgeX=zeros(1,10);
SurgeY=zeros(1,10);



for i=1:1:11
    %rotation speed [%]
    N=X(i);
    for j=1:101
        %mass flowrate
        m=M(j);
        %Mass flow dimensionless
        Mr(j)=m*sqrt(Rhat*T1)/(D^2*P1);
        %Nr
        Nr=(N*D)/sqrt(Rhat*T1);
        Nr0=(N0*D)/sqrt(Rhat*T1);
        %Semi-height
        H = H0 - 1.2*(H0+Pr0/2-1)*(1-Nr/Nr0);
        %Semi-width
        W=W0*(Nr/Nr0)^(1/3);
        %Pressure ratio
        Pr(i,j)=Pr0+H*(1+3/2*(Mr(j)/W-1)-1/2*(Mr(j)/W-1)^3);
        %efficiency
        eta(i,j)=eta0*((1-((H-H0)/H0)^2)-1000*(Mr(j)-2*W)^2);
    end
    if(i~=1)
        %SurgeY(i)=Pr0+2*H;
        SurgeY(i)=Pr0+H*(1+3/2*(2*W/W-1)-1/2*((2*W/W-1)^3));
        SurgeX(i)=2*W;
    end

end

for i=1:11


end

%Prtemp=Pr(2,:);
figure(1)

%plot(Mr(Prtemp>0),Prtemp(Prtemp>0))

for i=2:size(Pr+1,1)
    subplot(211)
    Prtemp=Pr(i,:);
    plot(Mr(Prtemp>0),Prtemp(Prtemp>0))
    if i==2
        hold on
    end

    subplot(212)
    etaTemp=eta(i,:);

%plot(Mr(etaTemp>min(Prtemp)&etaTemp<max(Prtemp)&etaTemp>0),etaTemp(
etaTemp>min(Prtemp)&etaTemp<max(Prtemp)&etaTemp>0));
    plot(Mr(Prtemp>0&etaTemp>0),etaTemp(Prtemp>0&etaTemp>0))
    if i==2
        hold on
    end

end

hold off
subplot(212)
set(gca,'Xlim',[0.06 0.18]);
set(gca,'FontSize',12)
xlabel('m_r [-]')
ylabel('\eta [-]')

subplot(211)
plot(SurgeX,SurgeY,'-.r*')
set(gca,'Xlim',[0.06 0.18]);
set(gca,'FontSize',12)
ylabel('P_r [-]')
```

D Optimization results



```
                                  Max     Line search  Directional  First-order
 Iter F-count        f(x)   constraint   steplength   derivative   optimality Procedure
    0      9     0.141997        1.832                                         Infeasible start point
    1     18     0.104383        1.524            1       -0.178        0.354
    2     27     0.111189       0.3029            1        0.104        0.206
    3     36     0.113343       0.0168            1        0.101        0.208
    4     45     0.113425    0.0002418            1      0.00982          0.2  Hessian modified
    5     54     0.113063     0.005048            1     -0.00646        0.189  Hessian modified
    6     63     0.113096    7.285e-006           1       0.0572        0.191  Hessian modified

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the selected value of the function tolerance,
and constraints are satisfied to within the selected value of the constraint tolerance.

<stopping criteria details>

Active inequalities (to within options.TolCon = 0.0001):
  lower      upper     ineqlin   ineqnonlin
                                     1
                                     2
>> X

X =

  623.8851
 -156.0070
  533.3258
    0.6507

>> FVAL

FVAL =

  1.1310e+005
```

E Risk Assessment

No risks were assumed relevant for this project.