NTNU – Trondheim
Norwegian University of
Science and Technology

DEPARTMENT OF CHEMICAL ENGINEERING

TKP4550 SPECIALIZATION PROJECT

Economic Plantwide Control using Commercial Process Simulation Software

Written by:
Adriana REYES LÚA

Supervisors:
Sigurd SKOGESTAD
Vladimiros MINASIDIS

December 10th, 2013

Abstract

It is desired to make the Economic Plantwide Control design procedure proposed by Skogestad (2000, 2004, 2012) available for engineers without deep knowledge of process control or optimization. The integration of the use of commercial process simulators to obtain the process model could be a useful tool for the automation of the economic plantwide design procedure. However, process simulators are set up in "design mode" and often work poorly in "operation mode". In this project, the use of commercial process simulators to generate process models suitable for an automated economic plantwide control procedure is explored. The analyzed process is methanol production, as it consists of: a reactor, a separator, and a recycle stream with purge. Simulations were made in `UniSim R400 Design Suite`. The optimization for nominal conditions and disturbed process was done using `Matlab` NLP `fmincon` algorithm.

Acknowledgments

I would like to thank Vladimiros Minasidis for his time, patience, guidance, feedback, and availability. In particular I would like to thank him for helping me writing the optimization code and using the COM interface.

I would also like to thank Sigurd Skogestad for allowing me to be part of the Process Systems Engineering group and letting me work on this project.

Contents

List of Figures

List of Tables

1 Introduction

Plantwide control refers to a control structure design for complete chemical plants. Each chemical plant is unique and there are multiple control layers and potentially an enormous number of variables involved. Moreover, it is of the interest of the industry to operate chemical plants with the best possible economical performance. Therefore, it is a challenge to develop a systematic procedure for designing control structures that achieve safe and close to optimal economic performance.

A systematic procedure for the design of an economic plantwide control system was proposed by Skogestad (2000). The main goal of this procedure is to design an optimal control structure for a complete chemical plant based on steady state plant economics. It is a stepwise procedure clearly separated into a top-down part, concerned with the steady-state economics, and a bottom-up part, mainly concerned with stabilization and pairing of loops (Skogestad, 2012).

Downs and Skogestad (2011) mention that techniques for plantwide process control require some characteristics in order to be accepted by process engineers in the industry. These techniques must result in processes with near optimal operation but at the same time should not employ complex control technology. Moreover, the control structure design procedure should not require "the care and feeding of control experts". Minasidis et al. (2013) observed that an important step would be to develop an automated procedure, hiding unnecessary complexities.

Commercial process simulators are a standard tool for process engineers in industry. They are convenient in terms of generating the process model because the engineer can set it in a rather intuitive manner, without the express mathematical model. Besides, this type of software includes a considerable amount of useful information such as kinetic and thermodynamic data. Previously, process simulators have been used to generate the steady state process model for the plantwide control procedure, obtaining good results and insight (Brandao de Araujo, 2007; Panahi, 2011; Jacobsen, 2011).

However, the use of process simulators to generate the process model still has some pending issues regarding the implementation of the plantwide process control procedure. One drawback is that process simulators are set up in "design mode" and often work poorly in "operation mode". Another disadvantage is that the model usually results in a large non-linear equation set with poor numerical properties for optimization (Skogestad, 2012). Furthermore, the way that the model is commonly set and optimized ends up being useful to analyze a specific case for which it was designed, and makes it difficult to extend the analysis to other models or other optimization methods.

This said, the implementation of a plantwide control procedure could be simpler with the integration of the automatic design in the major process simulators (Minasidis et al., 2013). Therefore, the objective of the present study is to explore the use of commercial process simulators to generate process models suitable for an automated economic plantwide control procedure. The application of this analysis is mainly on the first two steps of the top-down part of the plantwide control procedure.

As a typical chemical plant, we consider a process consisting of: a reactor, a separator, and a recycle stream with purge. The methanol plant is selected because it incorporates the basic structure of most of the chemical plants. `UniSim Design R400` was used to generate the process model. The optimization at nominal conditions and with disturbances was done using the `Matlab` gradient-based (NLP) algorithm `fmincon`.

2 Plantwide Control

This section describes briefly the plantwide control procedure proposed by Skogestad (2000), as presented in Skogestad (2004) and Skogestad (2012). As short overview of the complete procedure is included, but the description will be focused on the first two steps, where is the main application of the analysis in this report.

2.1 Basic concepts

Plantwide control refers to a control structure design applied to chemical plants; specifically, to the control philosophy for the whole plant. It might be thinkable to try to formulate the mathematical problem to describe and control the whole plant. However, it would be expensive and unpractical for normal-sized chemical plants, as an acceptable control can be achieved with simpler structures.

Figure 1 shows the typical control hierarchy in a chemical plant. It decomposes the overall control problem on a time scale basis. The upper layers are explicitly related to economic optimization. The presented procedure deals with the two lower layers.

Basically, the control system should: stabilize the plant and implement a near-optimal operation. Stabilization occurs in the regulatory control layer, in a fast time scale and is usually done with PID controllers. It does not use degrees of freedom because its setpoints come from the supervisory (upper) layer. Supervisory control, which sends the set points to the regulatory control can be achieved with PID controllers, but MPC is currently a widespread tool. The setpoints for the supervisory control come from the optimization layer.

In the end, after the plantwide control procedure has been followed, the following decisions will be made:

- **Decision 1:** selection of primary controlled variables (CV_1).
- **Decision 2:** selection of secondary controlled variables (CV_2).
- **Decision 3:** location of the throughput manipulator (TPM).
- **Decision 4:** pair valves to controlled variables (CV_2).

Primary controlled variables are also called economic variables; while secondary controlled variables are also called stabilizing variables. These are subsets or combinations of the measured variables. The selection or combination is done using matrices H and H_2.

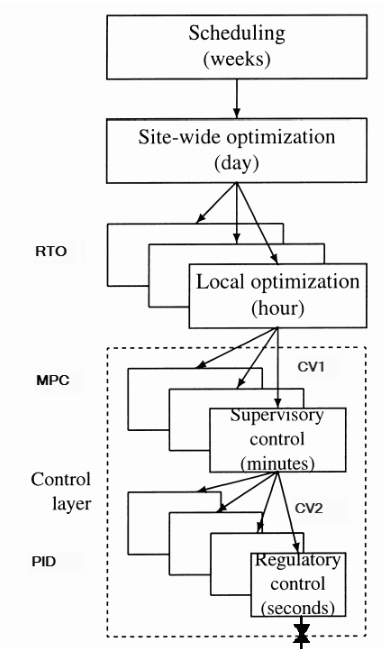


Figure 1: Typical control hierarchy in a chemical plant (Skogestad, 2004).

As Skogestad (2004) explains, to achieve a truly optimal operation, the model would need to be perfect and all the measurements should be available and reliable; which is unrealistic. Then, the concept of loss (L) is introduced as "the difference between the actual value of the cost function obtained with a specific control strategy, and the truly optimal value of the cost function". This concept brings the idea to find controlled variables such that, when keeping the setpoints constant we get an acceptable loss; even with disturbances. This way, we would not need to be constantly reoptimizing every time that disturbances occur. This is called *"self-optimizing control"*.

Figure 2 illustrates the concept of loss. It can be seen that when disturbances occur, the optimum cost is also modified and if we wanted to stay at the optimum we clearly would need to reoptimize. It can also be seen that by keeping the setpoints of the controlled variables constant, there is a loss. In the figure, by keeping the setpoints of CV_1 constant, it is possible to achieve a smaller loss than by keeping constant the setpoints of CV_2.

Another factor that affects the optimum is the measurement error. The most convenient variables to keep constant are those that are active constraints or those controlled variables for which the cost is insensitive; case (a) and (b) in Figure 3. If the cost is very sensitive to the value of the controlled variable, implementation is harder and it may not work with a large measurement error.

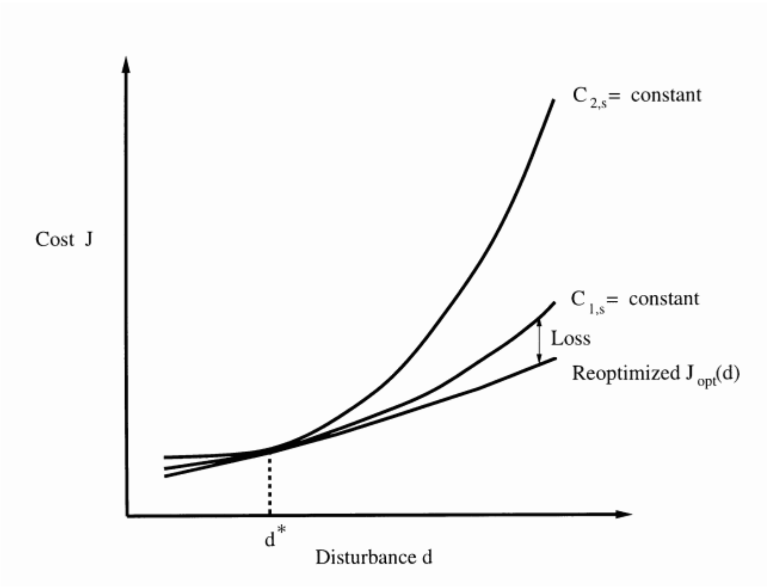
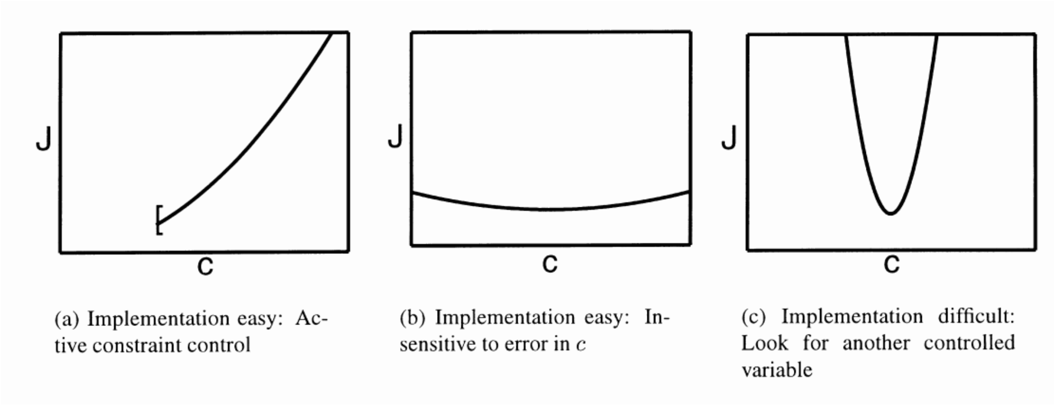


Figure 2: Loss imposed by keeping constant setpoint for the controlled variable (Skogestad, 2004).



(a) Implementation easy: Active constraint control

(b) Implementation easy: Insensitive to error in c

(c) Implementation difficult: Look for another controlled variable

Figure 3: Implementing the controlled variable (Skogestad, 2004).

2.2 Plantwide control procedure

The procedure is separated in two main parts: top-down and bottom-up. The top-down part focuses on steady-state optimal operation and economics. The bottom-up part focuses on the control layer structure and, while the steady-state considerations are still relevant, a dynamic model is required.

The procedure is as follows:

1. **Top-down**
    - **Step 1:** Define operational objectives (economics) and constraints.
    - **Step 2:** Identify steady state degrees of freedom and determine steady state operation conditions.
    - **Step 3:** Identify candidate measurements y and select CV_1 =H y .
    - **Step 4:** Select the location of the throughput manipulator.
2. **Bottom-up**
    - **Step 5:** Select the structure of the regulatory control layer: $CV_2 = H_2$y and pairings for CV_2.
    - **Step 6:** Select the structure of the supervisory control layer.
    - **Step 7:** Select the structure for the optimization layer (RTO) - if required.

2.2.1 Step 1: Define operational objectives

The operational objective is defined as a scalar cost function J (\$/s) that should be minimized. Typically:

$$J = \text{cost of feed} + \text{cost of utilities} - \text{value of products} \tag{1}$$

Constraints are operational constraints such as minimum and maximum flows, temperatures and pressures. Quality specifications, safety and environmental requirements should also be included here. Then, the optimization problem looks like:

$$\begin{aligned} &\min_{u} && J(u,x,d) \\ &\text{s.t.} && \\ &\text{model equations} && f(u,x,d) = 0 \\ &\text{operational constraints} && g(u,x,d) \leq 0 \end{aligned} \tag{2}$$

Where u are the degrees of freedom for operation; they are "for operation" because the equipment is fixed. It is the number of *u's* that is important because it does not really matter which variables we include in u as long as they make up an independent set. The disturbances d could be changes in feed rate and composition, changes in specifications, changes in prices, among others. The internal variables (states) are denoted by x.

2.2.2 Step 2: Determine steady-state optimal operation

The steady-state optimization problem was defined in Step 1. The process model can be developed explicitly or it could be indirectly provided by some process simulator. The challenge of using process simulators is that the resulting model is a non-linear equation set with poor numerical properties for optimization (Skogestad, 2012).

The operational mode is chosen in this step. It can be either:

- *Mode 1:* Maximize efficiency given throughput - this results in a trade-off between valuable product recovery and energy usage.
- *Mode 2:* Maximize production - when product prices are high compared to energy and raw material's prices. In this case, the feed rate is a degree of freedom.

In order to determine the steady-state optimal operation: the degrees of freedom must be identified, the important disturbances should be identified, and the operation should be optimized, also for the disturbances. In the end, constraint regions (regions of operation with the same active constraints) should be found. In summary, in Step 2 the following should be done:

- **Identify steady state degrees of freedom**
  Here is important to differentiate between the physical degrees of freedom and the steady state degrees of freedom u. Physical degrees of freedom correspond to valves while steady state degrees of freedom are those that affect the cost function J. The former are the ones that are required for the optimization. They can be identified either by valve counting or using the potential degrees of freedom method, as described in Skogestad (2012).
- **Identify important disturbances and their expected range**
  The "importance" of a disturbance is proportional to the sensitivity of the cost function to that disturbance. Common important disturbances are the feed rate and feed composition. Other disturbances could be changes in product specification and active constraints, changes in parameters (equilibrium constants, efficiencies), and price variations.
- **Identify active constraints regions**
  Once that the disturbances and their range is specified, the function is optimized along the disturbance space. Finally, the active constraints regions are found.

2.2.3 Step 3: Select primary (economic) controlled variables

Each steady state degree of freedom needs to be paired with a primary controlled variable. Active constraints could be seen as self-optimizing variables because at the optimum they are constant. They can either be inputs y or outputs u. Active input constraints would mean to fully close or open a valve. Active output constraints would require a controller. As the controller would require some time to adjust after a disturbance and there will be some measurement error, the setpoint should not be exactly at the constrained value; a *back-off* is required. After pairing the active constraints, self-optimizing controlled variables for the remaining degrees of freedom should be identified. First, candidate measurements and their expected error should be identified. Then, the primary controlled variables are selected. This is done using a selection or combination matrix $\mathbf{H}$. There are some qualitative requirements for the selection of variables, and there are quantitative approaches discussed elsewhere (Downs and Skogestad, 2011; Skogestad, 2012; Minasidis et al., 2013).

2.2.4 Step 4: Select the location of throughput manipulator

One degree of freedom is specified as the throughput manipulator (TPM); it defines the mass moved through the plant. It could be situated anywhere in the plant, but the location will affect the economics and the structure of the regulatory system. This is the decision that links the top-down and the bottom-up part of the procedure. The TPM is further discussed by Aske and Skogestad (2009) and Skogestad (2012).

2.2.5 Step 5: Select the structure of the regulatory control layer

In this step the regulatory control variables CV_2=H_2y are selected; then, inputs and pairings for the CV_2 are selected. No degrees of freedom are actually used here because the setpoints for CV_2 are actually the the manipulated variables of the supervisory control layer. The selection should be done taking into account that by controlling CV_2, the effect of disturbances on CV_1 should be small (local disturbance rejection) and that the effect of disturbances on internal variables x should also be small.

2.2.6 Step 6: Select the structure of the supervisory control layer

There are two alternatives for this layer: advanced single loop control (PID control with some additions) or MPC. This layer must not only control the primary controlled variables CV_1 but also supervise the performance of the regulatory layer and switch the controlled variables if necessary.

2.2.7 Step 7: Select the structure of the optimization layer

The real time optimization layer sends the setpoints for the primary controlled variables and updates them if there are changes of the active constraint region. If self-optimizing variables are chosen, it is probable that the benefit of RTO is not so high.

3 Methanol Process Description

As a typical chemical plant, we consider a process consisting of a reactor, a separator and a recycle stream with purge. The methanol plant was selected because it incorporates this basic structure.

Synthesis gas (*syngas*) composed by hydrogen and carbon monoxide is the raw material for methanol production. Typically, syngas is produced onsite from natural gas, as shown in Figure 4. Syngas with some carbon dioxide is fed to the methanol production section. Crude methanol (containing water) is sent to a purifying section to produce high purity methanol (≥ 99.5%) (Zhang et al., 2013). For the purpose of this analysis, syngas production and methanol purification are not included and only the delimited section in Figure 4 will be considered. Syngas is considered as the feed (and disturbance) to the process and crude methanol is considered to be the product. Typical plant capacities range from 150 to 6000 t/d (Moulijn et al., 2013).

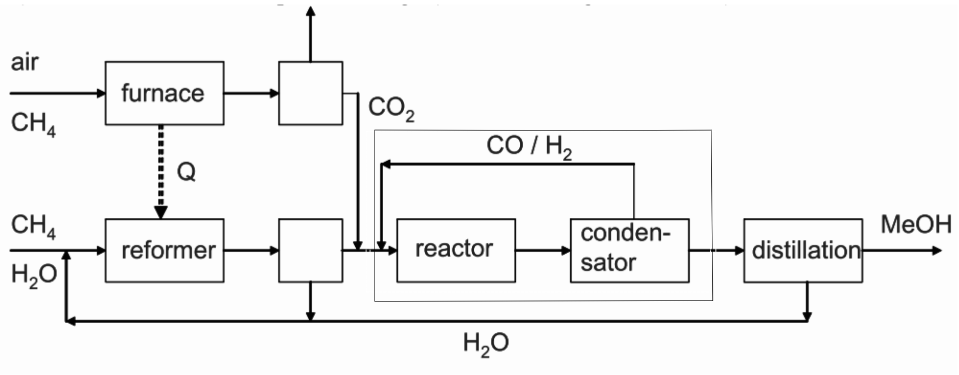


Figure 4: Conventional methanol process scheme, modified from Braunschweig et al. (2008).

From ca. 1830 - 1923, methanol was produced by dry distillation of wood. It was first synthesized industrially in 1923 from syngas. To reach acceptable conversions, high pressure (250-350 bar) and temperatures of 320-450 °C were required. In the 1960's, the ability to produce sulfur-free syngas and new catalysts (Cu/ZnO) allowed the production of methanol at milder conditions, especially regarding pressure. "Low-pressure plants" operate at 50-100 bar and 200-300 °C. The upper temperature bound is because at higher temperatures sintering occurs (Lange, 2001; Speight, 2002; Fiedler et al., 2005; Moulijn et al., 2013).

3.1 Reaction scheme

The main reactions for the formation of methanol are presented in equation set (3). Actually, only two of these three reactions are independent, and the reverse water-gas-shift reaction (3c) can be obtained by coupling reactions (3a) and (3b).

$$CO + 2H_2 \rightleftarrows CH_3OH \qquad \Delta H_{298\mathrm{K}} = -91\ \mathrm{kJ/mol} \tag{3a}$$

$$CO_2 + 3H_2 \rightleftarrows CH_3OH + H_2O \qquad \Delta H_{298\mathrm{K}} = -50\ \mathrm{kJ/mol} \tag{3b}$$

$$CO_2 + H_2 \rightleftarrows CO + H_2O \qquad \Delta H_{298\mathrm{K}} = +41\ \mathrm{kJ/mol} \tag{3c}$$

The ideal syngas for methanol production has a stochiometric number (SN) of 2.05. By looking only at reaction (3a), a required H_2/CO ratio of 2 mol/mol can be deduced. However, the concept of stochiometric number is required because methanol is also produced through reaction (3b). A lower stochiometric number increases the formation of side products such as higher boiling alcohols and dimethyl ether; a lower ratio implies unreacting hydrogen. Moreover, a small concentration of CO_2 (about 5%) increases catalyst activity (Løvik, 2001; Moulijn et al., 2013). Klier et al. (1982) found that at lower concentration of CO_2 the catalyst is deactivated by overreduction and at higher concentrations of CO_2, the synthesis is retarded by a strong adsorption of this gas. The low reaction temperatures possible with Cu/ZnO catalysts allow the high selectivity of current processes (Lange, 2001).

$$SN = \frac{H_2 - CO_2}{CO + CO_2} \tag{4}$$

3.1.1 Kinetics

A number of kinetic rate equations have been proposed in literature. Riaz et al. (2013) presented an updated summary of kinetic models for methanol synthesis. Most of the models are based on Langmuir-Hinshelwood kinetics or power law kinetics. Many models are also based on different variations of a $Cu/ZnO/Al_2O_3$ catalyst. A model that has been referred on several studies is the one proposed by Graaf et al. (1988) because it considered the three main reactions. Vanden Bussche and Froment (1996) presented a steady state kinetic model considering also the three reactions, and providing information about the catalyst.

A drawback of most published models is that the experiments on which they are based, in other words the range of validity, is up to about 50 bar (Graaf et al., 1988; Vanden Bussche and Froment, 1996; Lim et al., 2009; Riaz et al., 2013), while the operating pressure of industrial reactors is commonly around 80 bar.

3.2 Types of reactor

Commercial processes use a reactor with a circulation loop. Recirculation is standard because of the low single-pass conversion[1]. As the overall reaction is exothermic, quench reactors and cooled multitubular reactors are applied (Figure 5). The ICI process is the most representative for the quench scheme. An adiabatic reactor is used; the reactor is a single catalyst bed and cold reactant gas at different heights of the bed is used to quench the reactor. The Lurgi process is the most representative for the multitubular scheme. Catalyst particles are located in the tubes and boiler feed water (BFW) cools the reaction, which is nearly isothermal. Although most commercial processes are two-phase processes, recently a slurry process has been developed (Lange, 2001; Moulijn et al., 2013).

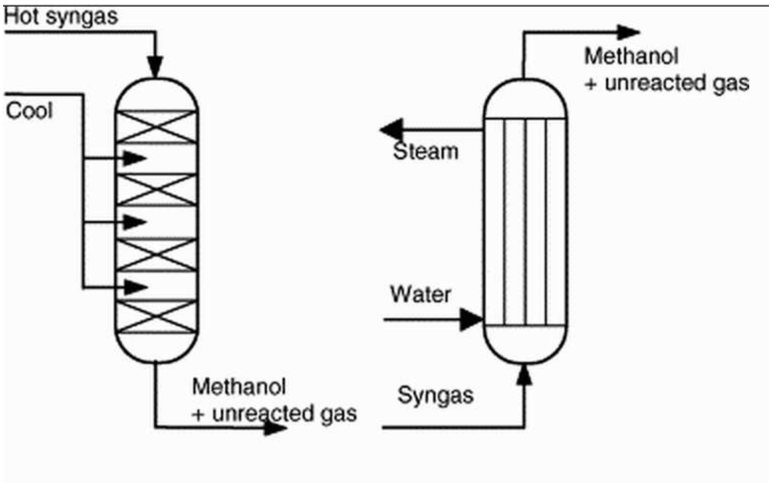


Figure 5: Methanol reactor types: quench (left) and multitubular (right) (Hamelinck and Faaij, 2002).

3.3 Process flow diagram

Process flow diagrams for the industrial production of methanol are similar and the most important difference is the reactor. A general scheme is presented in Figure 6. Fresh syngas is mixed with recycled syngas and the mixture is preheated before entering the reactor. There is some kind of temperature control in the reactor, either by quenching or water cooling. When quenching is used as a means of cooling, as in the ICI process, the quenching flows are not pre-heated.

[1] There are some papers that discuss single pass configurations (Hamelinck and Faaij, 2002; Pellegrini et al., 2011).

The reactor outlet is cooled by heat exchange with the feed and cooled further to separate methanol and water from the unreacted gas. The gas is recycled after purging a small part to keep the concentration of inert components within limits. The product of this section is called *crude methanol*. Crude methanol is purified in a distillation section, which is not considered in this analysis.

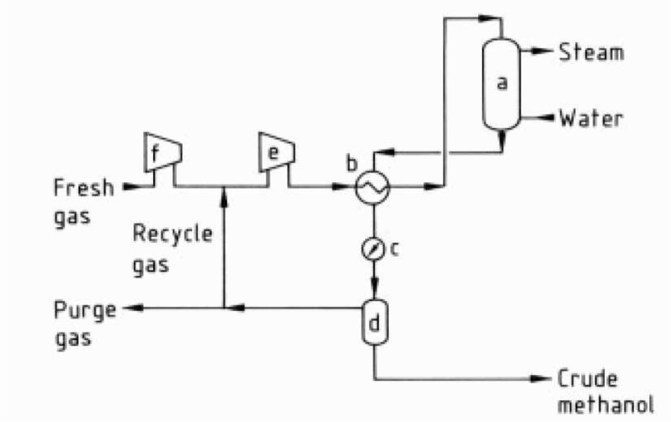


Figure 6: Methanol synthesis: a) reactor; b) heat exchanger; c) cooler; d) separator; e) recycle compressor; f) fresh gas compressor (Fiedler et al., 2005).

3.4 Thermodynamic model

Modifications of Soave-Redlich-Kwong (SRK) and Peng-Robinson (PR) are widely used to calculate thermodynamic properties of hydrocarbon mixtures. SRK equation of state is reported to have a good fit for the methanol-water-carbon monoxide system. SRK or some modified version has been used for several previous studies (Løvik, 2001; Arthur, 2010; Van-Dal and Bouallou, 2013).

While SRK is recommended for the methanol-water system, the rigorous SRK model does not predict phase equilibrium accurately enough due to the presence of the methanol-carbon dioxide system and due to the presence of hydrogen. The three compounds interact and the addition of water to methanol reduces the solubility of carbon dioxide. An extended SRK equation of state, such as the Mathias' polar correction factor, gives better results than the original SRK equation of state (Chang and Rousseau, 1985; Graaf et al., 1986; Løvik, 2001; Rostrup-Nielsen and Christiansen, 2011).

Peng-Robinson intends to improve SRK ability to predict hydrocarbons properties in the vicinity of the critical region (Ahmed, 1989). It has been proven to fit systems where low molecular weight alcohols, water and CO_2 are present (Breman and Beenackers, 1996; Joung et al., 2001; Shahrokhi and Baghmisheh, 2005;

Zhang et al., 2013; Kim et al., 2013). PRSV gives a good description of nonideal systems by both enhancing pure compound vapor pressure prediction, specifically water-alcohol systems (Pellegrini et al., 2011).

Due to the non-linear increase in the solubility of carbon dioxide, most models report some deviations when approaching the critical point, as reported by Yoon et al. (1993) and Chang et al. (1997). The critical point of methanol is very close to the process conditions[2]. Chang et al. (1998) and Joung et al. (2001) analyzed the performance of Peng-Robinson to model the methanol-CO_2 system in the vicinity of the critical region for CO_2-H_2O system and concluded that performed well.

[2] The critical point is at 239.45°C and 80.9 bar (Chang et al., 1997).

4 Process Simulation

4.1 Process flow diagram

As mentioned in Section 3, this analysis considers the methanol production section, with syngas as the raw material and crude methanol as the main product. After an initial analysis, the simplified process flowsheet proposed by Løvik (2001) and shown in Figure 7 was considered to best serve the purpose of the present study. This process considers a Lurgi reactor, in which the reaction heat is transferred to boiling water and the reactor temperature is actually controlled by the pressure of the boiling water to produce medium pressure steam. Fresh syngas is considered to have the required conditions to enter the loop.

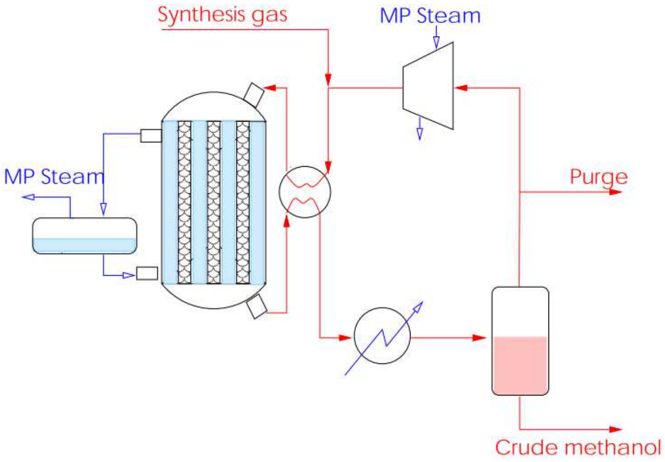


Figure 7: Methanol synthesis loop with Lurgi reactor (Løvik, 2001).

The simulation flow diagram is shown in Figure 8 and is based in Figure 7. The simulation was done in `UniSim R400`. Fresh syngas is fed to the loop at 140°C and the same pressure as the output of the recirculation compressor (Arthur, 2010). This stream is mixed with the outlet of the recirculation compressor and the mixed stream is pre-heated before entering the reactor. Pressure is set upstream the reactor. The outlet stream of the reactor serves as pre-heating medium for the inlet. The outlet is further cooled to allow the separation of unreacted gas and crude methanol (methanol and water). A fraction of the gaseous stream is purged. The recirculated gas is then compressed.

Typical operating temperature and pressure reported by Løvik (2001) and Arthur (2010) for Lurgi reactors are shown in Table 1. Fresh syngas make-up flow was set to 6000 t/d. The nominal composition of fresh syngas is shown in Table 2. Methane is considered to be inert.

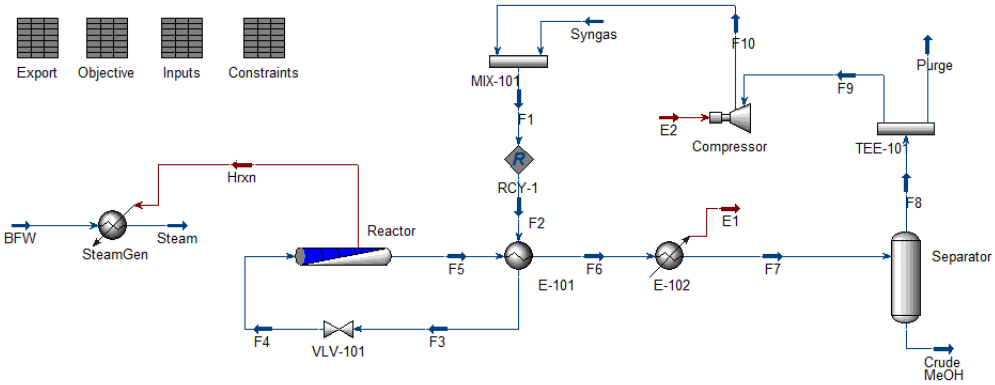


Figure 8: UniSim process flow diagram.

Table 1: Typical operating conditions.

Parameter	Value
Reactor temperature	250 °C
Reactor pressure	80 bar
Separator temperature	45 °C

4.2 Reactor

The reactor was initially modeled as an equilibrium reactor with cooling, which corresponds to a well-sized Lurgi reactor. The reactions that were added are the conversion of syngas to methanol and the reaction of carbon dioxide and hydrogen to produce methanol and water. The reverse water gas shift reaction also occurs, but it is not required to explicitly add it because it is not independent from the other two. The reaction equilibrium data that was used for these reactions is one of the pre-loaded tables in UniSim. It considers the equilibrium constant on a partial pressure basis, and approximates it as function of temperature. The effect of pressure is reflected in the partial pressure of the components.

To consider the effect of the size of the reactor on the operation a PFR model was introduced instead of the initial equilibrium model. Heterogeneous catalytic reactions for the hydrogenation of carbon dioxide and for the reverse water-gas-shift reaction were used to model the reactor. The kinetic model is the one proposed by Vanden Bussche and Froment (1996) as implemented by Arthur (2010). Besides the kinetic constants, Vanden Bussche and Froment (1996) gives the information for the $Cu/ZnO/Al_2O_3$ catalyst particle, while Arthur (2010) develops the model to introduce it to the process simulator and provides the sizing of the reactor. The pressure drop through the compressor is calculated by UniSim as per the Ergun equation.

Table 2: Nominal composition of syngas.

Component	mol fraction
Hydrogen	0.63
Carbon monoxide	0.31
Carbon dioxide	0.05
Methane	0.01

4.2.1 Steam generation

Temperature control is done by means of the production of steam, by controlling the steam pressure. For the purpose of the simulation, the temperature of the reactor is set for flow F_5 (Figure 7). The temperature of the boiler feed water (BFW) is set equal to the reactor temperature, while the outlet of the steam generator is set to be steam at the saturation point. The pressure and mass flow of steam are calculated given temperature, saturation (1), and the available energy, which corresponds to the heat that is removed from the reactor.

4.3 Heat exchangers

Disregarding the heat exchange in the reactor, there are two heat exchangers in the process: E-101 and E-102. E-101 uses part of the energy in the reactor outlet stream to pre-heat the inlet stream and, at the same time, as a first cooling of the outlet of the reactor. E-102 is a cooler that adjusts the temperature to allow the separation of crude methanol from the gases. E-102 uses cooling water (CWS) as cooling medium.

`UniSim` is thought as a process design software and the standard configuration recalculates the heat exchange area every time that the cooling or heating requirements are modified. The intention of this analysis is to design a control structure and not to design the plant; therefore, the heat exchange area should be fixed and the heat exchange capacity should be limited within a certain range. If the heat transfer coefficient is considered to be constant, *UA* should also be constant. The value of *UA* of E-101 at the nominal point was calculated and then put as an equality constraint in the optimization problem, varying the cold outlet temperature (Section 5 and Section 6).

E-102 is modeled as a simple cooler, without information of the cooling media in the simulation. However, the "real" heat exchanger has a maximum cooling capacity given by the maximum flow of cooling water and this should also be implemented as a constraint.

4.4 Circulation compressor

The circulation compressor in a methanol plant is a single-stage centrifugal compressor, with a low pressure ratio, commonly driven by a steam turbine (Lüdtke, 2004; Ohsaki et al., 2004), as depicted in the compressor in Figure 7. In the case of the simulation, the energy for the compressor is considered to be electrical energy.

A limitation on plant capacity (recirculated mass) is the compressor capacity. A minimum flow is required to avoid surge; a high flow would imply loss of compression capacity and probably undesired peaks in power consumption[3]. It is common that recirculation compressors in the methanol process have IGVs (inlet guide vanes), which allow a wider operating window and allow power savings (Bloch, 2006; Atlas Copco, 2011; Hitachi Plant Systems, 2012). For the sake of simplicity in the simulation, the recirculation compressor is considered to have a single operating curve (no IGVs and constant speed).

Compressor curves in Figure 9 were generated for the simulation, based on the typical behavior of centrifugal compressors. Having these curves enables to set an operating window for the compressor and for the recirculated flow. A minimum and maximum flow (or Δp or pressure ratio) should be implemented for the analysis (see Section 5). For given input conditions, the outlet pressure is given. The pressure in the loop is not controlled by the compressor, but by valve VLV-101.

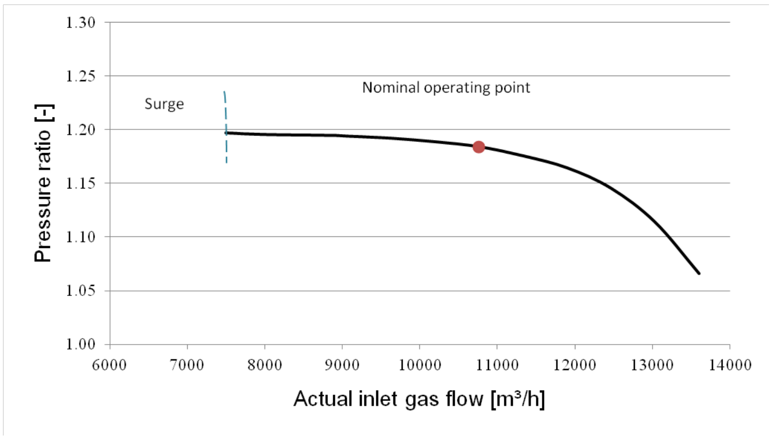


Figure 9: Compressor curve introduced in UniSim.

[3] A high load might cause the compressor to trip

4.5 Separator

The gas-liquid separator is modeled as a two-phase separator without pressure drop or change of temperature. The cooling is done in E-102. The separator has one inlet, a liquid outlet and a vapor outlet. The vapor phase carries most of the non-condensable gas, while the liquid phase is crude methanol and contains most of the methanol and the produced water.

4.6 Purge

The purge is separated through TEE-101. The recirculated flow rate is limited by the capacity of the compressor, as explained in section 4.4. Flow F_9 is the one that is set as an input; while the purge flow rate is calculated. As will be explained in Sections 5 and 6 , the capacity limits of the compressor (related to the compressor curve) are in gas flow units. Then, one of the decision variables is the inflow to the compressor, in gas flow units. In `UniSim` the default input is in mass flow units; so, the conversion from gas flow units to mass flow is done in a spreadsheet.

5 Optimization Problem

The problem can be seen as an optimization problem, formulated as[4]:

$$\begin{aligned} \min_{u} \quad & J(u,x,d) \\ \text{s.t.} \quad & l_b \leq u \leq u_b \\ & g(u,x,d) \leq 0 \\ & c(u,x,d) = 0 \end{aligned} \tag{5}$$

Where u refers to inputs or decision variables, d are the disturbances, and x refers to internal variables. Model equations are not required because the process model is represented indirectly by the simulator. Then, the constraints are operational constraints and aids for the simulation. The operational mode is *Mode I*; the throughput is given and variations of the feed rate are disturbances.

5.1 Objective function

The operational objective to be maximized is given by the profit. As the optimization problem is set as a minimization problem, the objective function considers as positive the cost of utilities and raw material, while the cost of the product and produced steam is set as negative.

$$J = C_{syn}\dot{m}_{syn} + C_{purge}\dot{m}_{purge} + C_{CWS}\dot{H}_{CWS} + C_e\dot{H}_e - C_{steam}\dot{m}_{steam} - C_{MeOH}\dot{m}_{MeOH} \tag{6}$$

5.1.1 Costs for the objective function

The cost of raw material and product price used for the optimization are shown in Table 3. Pellegrini et al. (2011) report the price of crude methanol, which is lower than the price of high purity methanol reported by Zhang et al. (2013) and Methanex (2013).

The purge is required to keep the inert concentration within acceptable limits, considering that the size of the equipment is given. The cost of the purge is set considering that it contains syngas, carbon dioxide, and some inert. The costs of syngas, depending on the H_2/CO ratio is given by Noureldin et al. (2013). Considering the approximate concentration of the purged gas, the cost was set as the cost of syngas (1:1).

[4] This formulation can be easily implemented in `fmincon` (MathWorks, 2013).

Table 3: Costs of methanol, syngas, and purge. Source: (Pellegrini et al., 2011; Noureldin et al., 2013)

Symbol	Variable	Cost	Unit
C_{MeOH}	Crude methanol	0.204	\$/kg
C_{syngas}	Syngas (2:1)	0.150	\$/kg
C_{purge}	Syngas purge (1:1)	0.080	\$/kg

The cost for utilities is shown in Table 4. The cost of electric energy is set to 0.032 \$/kWh as reported by Pellegrini et al. (2011)[5]. Cooling water is considered to be an inexpensive service [6]. Steam is produced using the heat of reaction. The price of steam is reported by Noureldin et al. (2013) as 0.008 \$/kg.

Table 4: Costs of utilities. Sources:(Pellegrini et al., 2011; Zhang et al., 2013; Noureldin et al., 2013)

Symbol	Variable	Cost	Unit
C_e	Electric Energy	0.032	\$/kWh
C_{CWS}	Cooling Water (CWS)	0.100	\$/GJ
C_{steam}	Steam	0.0008	\$/kg

5.2 Constraints

Constraints were set accordingly to a physical plant. UniSim is a design program. If physical constraints are not set, it will size the equipment accordingly to mass and energy flows. The idea of this work is not to design an optimal plant in terms of sizing, but to analyze the behavior of a given plant when designing a plantwide control structure. Therefore, there are two main sources for the constraints: those coming from operating values typical for the (methanol) process and those coming from the capacity and limitations of the specific plant.

To exemplify this, any methanol reactor has a similar maximum operating temperature (between 240-260°C (Aasberg Petersen et al., 2011)) because every methanol reactor uses the same type of catalyst that will sinter above a certain temperature (about 280-300 °C). However, the areas of the heat exchangers of each methanol plant are not necessarily the same and will limit the operating window of each specific plant.

[5] The cost reported by (Pellegrini et al., 2011) is in Saudi Arabia. However, the cost for the Norwegian industry is in the range of 0.05 \$/kWh (Statistics Norway, 2013)

[6] The cost of using demineralized water (DMW) instead of cooling water supply (CWS) would be in the range of 0.5 \$/GJ, considering 0.021 \$/treported by Pellegrini et al. (2011) and a ΔT of 10°C.

The sizing of the equipment as a source for constraints can be observed in three cases:

- **Compressor capacity:**

  Constraints have to be set to define lower and upper bounds on the actual gas flow at the inlet. These should be treated as hard constraints because they must be always satisfied in order to assure convergence on the process simulator. As a curve was introduced, if the inlet gas flow is not within the ranges of the curve, the simulation will stop.

- **E-101 heat exchanger:**

  Considering *UA* as constant, and given the heat capacities, inlet temperature, and the flows of both streams, there are no actual degrees of freedom. However, the process simulator "does not know" that *UA* is set and requires an outlet temperature as input. If the cold outlet temperature is used as input variable, there could be temperature crossings with the hot inlet temperature.

  For this reason, the minimum temperature approach (hot inlet-cold outlet) is included as decision variable to find the outlet temperature that would be obtained with the given heat exchanger. As this minimum temperature approach is required to avoid temperature crossings and assure convergence, it should be treated as a hard constraint. On the other hand, the simulation will converge even if *UA* is not constant. To set *UA* constant, a soft equality constraint is introduced. The value of *UA* is set to the value at the nominal operating point.

- **E-102 heat exchanger:**

  This is a similar case as E-101. However, in this case the second stream is cooling water. If the flow of water had been defined, this heat exchanger could have been handled a similar way as E-101. However, it was decided to vary directly the outlet temperature within a lower and an upper bound. The range of variation is small and should be possible with a real heat exchanger. Additionally, a range for the retired energy was set as a constraint. The result is physically feasible because given a ΔT in the cooling water and assuming a constant *UA*, there is a maximum and a minimum flow of water through the pipeline. Therefore, there is a maximum and minimum heat removal capacity.

Supposing that the design and sizing of a plant is given, the operating variables can be seen as inputs (decision variables) for the optimization. These can be varied within certain bounds, given in Table 5[7]. These are constraints in the form $l_b \leq u \leq u_b$.

[7] The tags refer to those in the process flow diagram shown in Figure 8.

Table 5: Lower and upper bounds for inputs.

Input	Unit	Lower bound	Upper bound
Reactor temperature	°C	240	260
Reactor pressure	bar	75	81
Separator temperature	°C	40	50
Compressor Inlet flow	$Act \mathrm{m}^3/\mathrm{h}$	7500	12000
E-101 min ΔT	°C	2	50
Slack for E-101 *UA*		-25	25

Among these six variables, only four are actual degrees of freedom. Two of them, E-101 ΔT and the slack for E-101 *UA*, are inputs for the simulation but it would not be possible to use them as manipulated variables in an actual control structure.

For some inputs, such as the temperature of the reactor, the lower and upper bounds can be easily defined based on process knowledge. However, as the behavior of the process is not linear, the bounds for some inputs are not so evident. If not well posed, some input combinations might violate physical principles.

The NLP solver will vary the inputs only within the lower and upper bounds (Section 6). Therefore, if there is a hard constraint, in the sense that it requires to be satisfied to assure the convergence of the process simulator, it must be included as an input along with those in Table 5.

It is important to select the input variables that will assure convergence of the process simulator in order to have a robust model. For example, for E-101 there are two possible decision variables to use to adjust the temperatures so that *UA* is kept constant. We could use the temperature of F_3 or use the temperature difference between reactor temperature (F_5) and and the cold outlet temperature (F_3) and then calculate the temperature of F_3. For some combinations of reactor temperature and E-101 cold outlet temperature there could be temperature crossings in the heat exchanger because the range within these two temperatures vary overlaps. Therefore, the temperature difference between these two streams was used as input variable.

Another example is given in the purge and compressor. As mentioned before, the curves for the compressor are given and if the inlet gas flow is not within the range of the curves, the simulation will fail to converge and will stop. To decide on how much to purge, the first option would be to use the purge split as a decision variable. However, this could lead to recycled gas flows out of the curve of the compressor. The recycled flow was used as a decision variable, and the bounds were set according to the compressor curve.

Once that combinations that would lead to an error in the process simulator have been handled as inputs, there are solutions that will not stop the process simulator from converging but would be unfeasible in a real plant. The "non-desired" solutions are discarded by introducing inequality constraints in the form $g(u,x,d) \leq 0$. From the point of view of the set-up, these are treated as soft constraints, because they are outputs and the NLP solver cannot avoid violating one at some point. Despite the solver would not consider it a solution, the process simulator would stop anyway.

For example, `UniSim` allows valves to increase pressure and coolers to increase temperature; a constraint on pressure drop can be set. In the same line, another source of soft constraints is related to the quality of the product. Having a low quality will not avoid the process simulator to converge. In this case, a lower limit on the concentration of methanol in the product (crude methanol) is set.

The inequality constraints ($g(u,x,d) \leq 0$) for the problem are:

$$0.90 - x_{MeOH} \leq 0 \tag{7a}$$
$$x_{CH_4} - 0.07 \leq 0 \tag{7b}$$
$$-LMDT_{E-101} \leq 0 \tag{7c}$$
$$0.80 * \dot{H}^*_{E-102} - \dot{H}_{E-102} \leq 0 \tag{7d}$$
$$-1.25 * \dot{H}^*_{E-102} + \dot{H}_{E-102} \leq 0 \tag{7e}$$
$$-\Delta p_{VLV-101} + 1 \leq 0 \tag{7f}$$

Where:

- x_{MeOH}: mol fraction of methanol in crude methanol stream.
- x_{CH_4}: mol fraction of methane in recycle stream.
- $LMDT_{E-101}$: log mean temperature difference in E-101. It should always be positive to avoid temperature crossing.
- $\dot{H}^*_{E-102}$: nominal heat flow removed from E-102.
- $\dot{H}_{E-102}$: actual heat flow removed from E-102.
- $\dot{Q}_{compressor}$: actual volumetric flow at the inlet of the compressor. The bounds are set to be within the compressor curve, in actual m^3/h.
- $\Delta p_{VLV-101}$: Pressure drop in VLV-101; pressure should decrease.

Actually, constraint (7c) should not be necessary because of the lower and upper bound on the minimum temperature approach in E-101. However, it serves as a reference. The same occurs with constraints (7d) and (7e), which should not be violated due to the bounds on E-102 outlet temperature.

The only equality constraint ($c(u,x,d) = 0$) for the problem is the one that keeps the exchange area of E-101 constant. As the NLP method does not assure that the constraint is satisfied, a slack variable was added as an input to have control over the error (Table 5).

$$UA_{E-101} - UA^*_{E-101} + s = 0 \tag{8}$$

6 Optimizing the Simulation

Once that the simulation was set in UniSim and the optimization problem was defined, the objective function and the constraints were introduced in spreadsheets so that the values could be read and modified by a Matlab code by means of the Component Object Model (COM) interface. This way, only Matlab handles the optimization while the process is completely simulated by UniSim.

The Component Object Model (COM) interface allows the use of a code in an automation client to interact with UniSim. The code can access exposed objects, making possible to send and receive information to and from UniSim (Oli Systems, 2007). It is important to outline that not every UniSim property is exposed through the COM interface. However, this was overcome using UniSim spreadsheets to access the values. This way, even properties that are not available through the COM interface become accessible - they become "values" in the spreadsheets. This gives valuable flexibility for the definition of a robust optimization problem and allows a "cleaner" communication.

It is important to stress that the way that the model and optimization were set, the NLP problem is formulated independently from the NLP solver. UniSim and Matlab communicate in a very clean way. For the NLP solver, the objective and variables are mere numeric values. On the other hand, if the process simulation is seen as the NLP problem, the inputs are only modified at specific pre-defined locations of the UniSim spreadsheets.

6.1 NLP Solver

NLP methods have been used previously for the methanol process with Lurgi reactors to analyze the effect of varying operating conditions on a process plant. Kralj and Glavič (2009) used a gradient method for a methanol process because, despite a global optimal solution is not guaranteed, it gives good results for complex processes.

In this study, the optimization was done using the non-linear gradient-based (programming) solver fmincon in Matlab. The interior-point algorithm was used and gradients are estimated using finite differences.

Process variables such as compositions, temperatures, and flows are in different orders of magnitude. To facilitate convergence, two scaling functions were written. One function scales from zero to one the input values read from UniSim and these scaled values are used in fmincom; the other function "descales" the variables to send them back to Unisim.

A feature of the Matlab code (Appendix C.1) is that it reads the number of bounded inputs, equality and inequality constraints. With this information, the NLP code (Appendix C.2) is generated. This way, there is no need to re-write the code for optimization.

6.2 Setting UniSim for optimization

The values for the optimization problem described in Section 5 were added to three dedicated spreadsheets. The values that are required for the optimization problem, such as the objective function, the inputs and the constraints are written in specific cells, which are defined in the Matlab code. The layout of the spreadsheets can be consulted in Appendix A.

The spreadsheets are:

- **Objective:** contains the objective function in cell A2, which is the only cell in this spreadsheet that interacts with Matlab. The rest of the spreadsheet is used to get the values from the simulation, introduce the costs and perform any other calculation that is required to to calculate the objective function.
- **Inputs:** columns A-E are read by Matlab. Decision variables set as constraints in the form $l_b \leq u \leq u_b$ are set here. Columns A through E contain: the actual value of the input, the lower bound, the upper bound, the initial value for the optimization, and the units of measurement. Column F is not read or modified by Matlab, but is used to describe the decision variable.
- **Constraints:** inequality constraints are specified in column A. Equality constraints are specified in column B. The rest of the columns are used to get the values and perform the required calculations.

Posing the optimization problem in a way that assures convergence is an important step before using Matlab to optimize the problem with and without disturbances. This has been described in Sections 4 and 5. In order to assure that the optimization result is consistent and constant among runs, the recycle sensitivities were increased. This was done by reducing the sensitivity value to 0.01 [8]. The maximum number of iterations was also increased to assure convergence and prevent the simulator from crashing.

[8] As explained in UniSim manual, the entered sensitivities values serve as a multiplier for UniSim internal convergence tolerances (Unisim, 2007).

7 Results

The goal of this analysis was to establish the basis to use a process simulator such as `UniSim` to systematically apply the plantwide control procedure. The first two steps of the plantwide procedure presented in Skogestad (2012) were applied: the definition of the operational objectives and the steady state optimization. In order to perform the steady state optimization in Step 2, a reliable steady state model must be available. The simulation described in Section 4 and the code and modifications described in Section 6 were used.

7.1 Step 1: Define operational objectives

Section 5 describes the definition of the optimization problem. Equation (6) defines the objective function, equation set (7) defines the inequality constraints for the outputs and Table 5 details the lower and upper bounds for the inputs. The goal is to minimize the cost satisfying operational constraints.

7.2 Step 2: Determine steady state optimal operation

This step is usually very time consuming (Skogestad, 2012), and this work is done with the intention of helping to automatize it.

7.2.1 Identification of steady-state degrees of freedom

It is important to outline that the simulation inputs do not correspond to the degrees of freedom for operation. Two of the simulation inputs were introduced to find the values that would make possible operation given that the equipment is fixed. There are five physical degrees of freedom (valves), as shown in Figure 10; but there are only four steady-state degrees of freedom.

In Figure 10 there is a valve to control the level in the separator. This adds a physical degree of freedom, but would not have any steady state effect. Therefore, there are four steady-state degrees of freedom. The valve to control the level in the separator was not included in the simulation (Figure 8). In this case, the "evident" physical degrees of freedom in the simulation correspond to the steady state degrees of freedom. This could be seen as a result of the fact that the simulation was set for a steady-state analysis.

Using the potential steady-state degrees of freedom method (Skogestad, 2012), we obtain the same result:

- splitter: 1 (TEE-101; *n=2; n-1=1*)
- heat exchanger: 2 (E-102 and SteamGen)
- pressure: 1 (VLV-101)

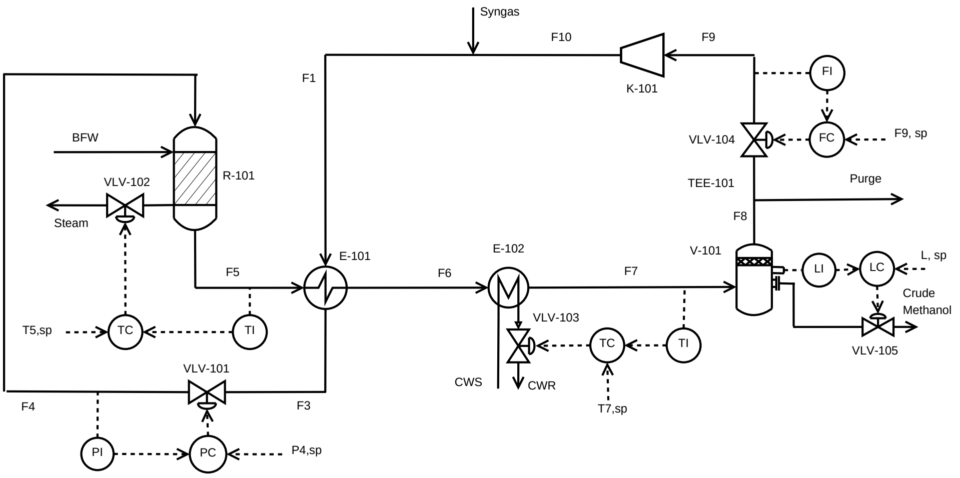


Figure 10: Process Flow Diagram.

Figure 10 shows how the actual manipulation of the process parameters would be done (there is a valve for each degree of freedom). Neither the work nor the speed can be varied in the compressor, so it cannot be considered as a degree of freedom. The reactor is in gas phase, and it is not a degree of freedom either. E-101 does not add a degree of freedom because both flows and the size are given and there is no bypass.

7.2.2 Identification of important disturbances

Two important disturbances are in the feed: the flow and the composition of fresh syngas. It was defined that the H_2:CO:CO_2 ratio in the syngas would be kept constant and that the mole fraction of the inert component (methane) would be the disturbance.

$$d = [\dot{m}_{syn}, x_{CH_4}] \tag{9}$$

7.2.3 Optimization

The process was optimized using the NLP solver and simulation as explained in Section 6. The nominal syngas make-up flow is 6000 t/d (250 000 kg/h) and the nominal composition is the one in Table 2. It was defined that syngas flow could vary ±15%. Keeping the H_2:CO:CO_2 ratio constant. The composition of the inert component in the syngas (methane) was varied ±20%.

7.2.4 Operating regions

Input constraints are identified in Table 6, while output constraints are identified in Table 7. Optimization was performed for seven levels for the flow (±15%, ±10%, ±5% and nominal) and five levels of composition (±10%, ±20% and nominal). It can be pinpointed that along the 35 optimization procedures, the simulation worked smoothly. It is clear that the time to find the solution depends on computing capabilities; but to give a sense of the velocity, convergence was reached within 5-15 minutes.

Table 6: Input inequality constraints identification

Constraint	Variable	Description
c1	F5 temperature	E-101 cold outlet
c2	F4 pressure	Reactor pressure
c3	F7 temperature	Separator temperature
c4	F9 flow	Flow to the compressor
c5a	UA slack	
c5b	E-101 ΔT	

Table 7: Output inequality constraints identification

Constraint	Variable	Definition
c6	MeOH mole concentration in crude methanol	$x_{MeOH} \geq 0.9$
c7	Methane mole concentration in recycle	$x_{CH_4} \leq 0.07$

Table 8 shows the active constraints in each of the identified regions depicted in Figure 11. For input constraints, that have a lower and upper bound, the subscript indicates whether it is the upper or the lower bound that is active. Numeric optimization results can be consulted in Appendix B.

Table 8: Identification of active constraint regions.

I	$c1_u$, $c3_u$, $c4_l$, c7	**IX**	$c2_u$, $c3_u$
II	$c1_u$, $c3_u$, $c4_l$, c6, c7	**X**	$c2_u$
III	$c3_u$, $c4_l$, c6, c7	**XI**	$c2_u$, $c4_u$, c6
IV	$c3_u$, $c4_l$, c6	**XII**	$c1_u$, $c2_u$, $c4_u$
V	$c2_u$, $c3_u$, c6	**XIII**	$c1_u$, $c2_u$, $c3_u$, $c4_u$
VI	$c2_u$, $c3_u$, c6, c7	**XIV**	$c1_u$, $c2_u$, $c3_u$
VII	c7	**XV**	$c1_u$, $c2_u$
VIII	$c2_u$, $c3_u$,c7		

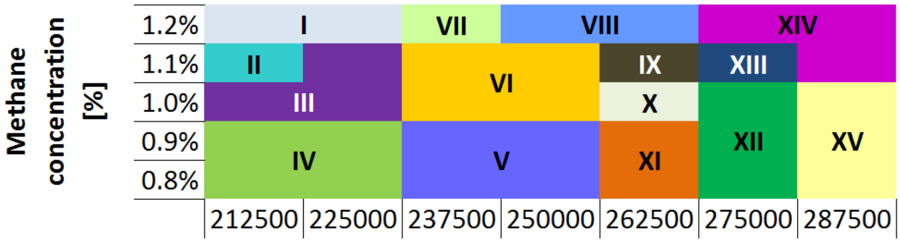


Figure 11: Active constraint regions.

The amount of constraint areas reflects the non-linearity of the process. Each optimization was run more than once to confirm the results. The results seem reasonable. Some observations are:

- Constraint c1, that limits reactor temperature becomes active at higher methane concentrations and also at higher flowrates. Given that reactor size is fixed, this could be because of the "need" to maintain the production rate when residence time is decreased.
- Constraint c2, that limits reactor pressure is active at middle to high flowrates. This could have a similar explanation as the trend for c1.
- Constraint c4, that limits the flow to the compressor is active in the lower bound at low flowrates and becomes active in the upper bound at middle-high flowrates. It is not active at every high flowrate; this could be explained by the fact that flowrate as a disturbance is mass flowrate and the bound for the compressor is gas flowrate, which is influenced by the composition, which is a product of reaction conditions and separation.
- Constraints that were added for the convergence of E-101 did not reach their bounds. This might be because they were set to "help" the simulation to converge and not to stop it. The minimum temperature approach in E-101 was of 3.8°C; the bound was set at 2°C.
- Constraint c6, that limits the minimum methanol concentration in crude methanol became active at low flowrates.
- Constraint c7, that limits the maximum concentration of methane in the recirculation becomes active when the concentration of methane in the syngas increases.

8 Discussion

Together with obtaining the model, the optimization step is often the most time consuming in the plantwide control procedure (Skogestad, 2012). The work presented in this report seeks to facilitate the integration of process simulators in the automatic design of the economic plantwide control procedure. Some insight about what is required from the process model for optimization, mainly applicable to the first two steps of the procedure, was obtained.

8.1 On the setting of the optimization model

Initially, it was intended to handle operating constraints such as the boundaries for the compressor curve by adding inequality constraints (equation set 7). However, the NLP solver evaluates and varies input parameters in the process simulator and then verifies whether the output constraints are satisfied or not. When some constraints are violated, the process simulator still converges. For example, when the equality constraint that assures that the heat exchange area does not change is not satisfied, the simulation still works; the "only" thing is that the solution does not serve the purpose of the analysis.

In the case of the recirculation compressor, if the actual inlet flow is not within the curve, it does not converge, causes an error and the optimization stops. Evidently, adding curves for more speeds or IGVs would widen the operating window, but would not solve the problem. In this case, it was initially solved by increasing the lower bound for the minimum purge, which initially was one of the input variables. However, choosing an arbitrary bound for split for the TEE could lead to an unnecessary sub-optimal solution. Then, it was decided to use the actual gas flow to the compressor as decision variable. As the actual gas flow of the product streams is not one of the standard inputs to the TEE in `UniSim`, additional calculations were required to obtain the equivalent mass flow and set it in the unit operation.

A similar case occurred with the temperature approach of E-101. In this case, a decision variable was required to adjust the temperatures so that the heat transfer area (*UA*) was kept constant. Despite setting a constraint for LMDT that would assure a feasible heat exchanger, the optimization procedure did not avoid the process simulator to "try" solutions with temperature crossings. In those cases, the process simulator failed to converge and stopped. This was solved by using the temperature difference between the hot inlet (F5) and the cold outlet (F3) instead of the temperature of the hot variable as a decision variable to find the temperatures in E-101. This was implemented by calculating the cold outlet temperature based on a given ΔT and the temperature of the hot inlet in an additional spreadsheet. Both of these values were linked from the Input spreadsheet.

Knowledge and insight of the process is a very powerful tool and well selected boundaries are very important to assure convergence of the simulator. From the point of view of the definition of the optimization problem, it is important to assure that the boundaries of the inputs give results that will converge. Therefore, it is recommended to set the hard constraints as inputs u or decision variables (Inputs spreadsheet), because they need to be satisfied at every moment to assure convergence in the simulator. Then, the soft constraints, which are not absolutely necessary for the convergence of the simulator should be set as equality and inequality constraints (Constraints spreadsheet). This way, these will help to discern between the solutions that satisfy requirements such as product quality or some equipment sizing.

From the procedure point of view, the number of degrees of freedom is what matters, and which variables we include in u is not really important Skogestad (2012). However, from the simulation point of view, an appropriate selection of variables, significantly improves robustness. In the end, the manipulated variables in control might be different from the inputs required in the process simulator. Using spreadsheets as means of communication with the optimization solver facilitates the clear distinction of inputs and outputs and eliminates possible errors due to mixing of these.

8.2 On the optimization method

By using the process simulator, there are no explicit equations that model the behavior of the process. This simplifies the modeling step of the optimization procedure. However, in a standard optimization problem, when the equations for the behavior of the parameters of the problem are given and before starting to actually solve the problem, it should be defined whether it is convex or not in the area of interest. Chemical processes are mostly non-linear and it is possible that operating conditions are in a non-convex area.

In this study, a NLP solver (`fmincon`) was used to optimize the problem. The results were good in the sense that convergence was reached and the optimization results were consistent. However, it has to be reminded that for each evaluation the gradient has to be calculated; to do this, the simulation needs to be run. This makes the algorithm not precisely fast. The time to converge varied between 5-15 minutes. It has to be mentioned that tolerances were small, to assure that the results were consistent given the high non-linearity of the process. The time to converge should not be a significant problem if a limited number of conditions is to be tested and if the analysis is to be done offline. However, there is an area of opportunity to improve this.

Another approach would be to use derivative-free optimization methods such as the Nelder-Mead method. These have the advantage that are designed to solve problems in which the derivatives are not available or costly to calculate, which is actually the case when using a process simulator. In the other hand, current algorithms are effective only for relatively small problems and the effective handling of constraints is still under investigation (Nocedal and Wright, 2006). As the process simulator convergence is affected by the appropriate handling of the constraints, especially the bounds for the inputs, this would be an issue to be solved. However, gradient-free methods could be a good option to reduce the time to solve the problem.

8.3 On the simulation results

The objective of this analysis was to explore the use of a commercial process simulator such as `UniSim` to design the control structure for a chemical plant. This procedure involves optimization and most of the identified challenges are related to the stability of the simulation when varying the input; in other words, when disturbing the process. In the end, the objective was reached because the simulation ran smoothly when disturbed and optimized.

The level of detail of the simulation was set trying to approximate the limitations of a real plant. If it was used to design a plant, most probably the simulation results would be different. For a deeper analysis, the level of detail might require to be increased. For example, the kinetic model that was implemented, proposed by Vanden Bussche and Froment (1996), was developed for pressures between 15 and 51 bar(as many other models), while the simulation is run at pressures in the range of 80 bar. Other aspects that could be modeled with more detail are: heat exchangers' pressure drops and heat transfer coefficients, as well as the compressor curves.

It has to be mentioned that there is no actual scan of the complete objective function. The process proved to be highly non-linear. Therefore, there is no obvious way to find out if the result is a global minimum; it can only be assured that it is a local minimum.

An effort was put on setting the costs for the objective function at real values. However, the objective function had a positive value, which would mean economical losses. This might be explained by the fact that the price for crude methanol was set at 0.204 $/kg, as done by Pellegrini et al. (2011). However, the commercial price for pure methanol (the actual final product) is in the range of 0.5$/kg (Methanex, 2013). If 0.5$/kghad been used instead of 0.204 $/kg, the value of the objective function would have been negative.

9 Conclusions

The use of commercial process simulators could be an important step to develop an automated procedure for designing control structures that achieve safe and close to optimal economic performance. Hiding unnecessary complexities would ease the acceptance of the plantwide control procedure by process engineers in the industry. However, the use of process simulators to generate the model still has unresolved issues. The plantwide control procedure intends to design the control structure of a given chemical plant, not to design the plant. Commercial simulators are set up in "design mode" and often work poorly in "operation mode". Also, there is no standardized way to set up the simulators for the optimization in Step 2 - and to be used for the rest of the steps.

In this project, the use of `UniSim`, one of the most popular commercial simulators, was explored. The analyzed process was a methanol plant, that includes the basic features of a typical chemical plant: a reactor, a separator, and a recycle stream with purge. This way, relevant issues of this configuration were analyzed. The model included sizing of equipment such as the reactor and the compressor. This way, the simulation could be run in "operation mode".

As capacity constraints were introduced, special attention was put on the convergence of an integrated heat exchanger and the recirculation compressor. Quality constraints such as minimum product concentration or maximum inert concentration were also introduced. In the end, a robust simulation that converged over a reasonable wide range of disturbances was obtained. With the simulation and optimization algorithm, active constraint regions were identified.

It became evident that a clear definition of input and output constraints is important to achieve a robust simulation. The simulation needs to be robust to avoid it from crashing during the optimization procedure. The optimization algorithm varies the inputs, and cannot "predict" the outputs before running the simulation. Therefore, the choice of input and output constraints becomes important for the convergence of the simulator.

The way that the simulation and the optimization were set up allows the NLP problem to be generated independently of the NLP solver and vice versa. The problem (model) is managed by the process simulator while the solver was managed by `Matlab`. In the future, the optimization could be done with a different algorithm or program, just by reading and modifying the input values in the simulation spreadsheets via COM interface. A gradient-based algorithm was used for this project, but the simulation could be used with a gradient-less algorithm, perhaps more apt for the type of model, in which no explicit equations are available.

9.1 Further Work

Some work that can be done after this analysis is:

- Apply the remaining steps of the plantwide control procedure.
- Generate a dynamic simulation and analyze the use of the process simulator to perform the controllability analysis for the bottom-up design. This should also be done in a consistent and systematic way.
- Explore the use of different types of optimization algorithms such as gradient-less algorithms in order to find out if results can be improved, considering the non-linearity of the process model.
- Make a more detailed analysis of the operating regions (finer mesh).
- Develop a more detailed simulation:
  - Compare effect on different types of reactor. The reactor model has been analyzed previously (Løvik, 2001; Kim et al., 2013).
  - Include cooling water in the system.
  - Add variable speed or IGVs to the compressor.
- Explore the use of "user variables" in `UniSim` to make more flexible the possible inputs for the simulation.

Bibliography

Aasberg Petersen, K., Nielsen, C. S., Dybkjær, I., and Perregaard, J. (2011). Large Scale Methanol Production from Natural Gas. Technical report, Haldor Topsoe.

Ahmed, T. H. (1989). *Hydrocarbon phase behavior.* Gulf Pub. Co.

Arthur, T. (2010). *Control Structure Design for Methanol Process.* Master thesis, Norwegian University of Science and Technology.

Aske, E. M. B. and Skogestad, S. (2009). Dynamic Degrees of Freedom for Tighter Bottleneck Control. In de Brito Alves, R. M., do Nascimento, C. A. O., and Biscaia, E. C., editors, *Computer Aided Chemical Engineering*, volume 27, pages 1275–1280. Elsevier.

Atlas Copco (2011). Driving Centrifugal Compressor Technology.

Bloch, H. P. (2006). Understanding Centrifugal Process Gas Compressors. In *Compressors and Modern Process Applications.* John Wiley & Sons, Inc.

Brandao de Araujo, A. C. (2007). *Studies on Plantwide Control.* Doctoral thesis for the degree of philosophiae doctor, Norwegian University of Science and Technology.

Braunschweig, B., Joulia, X., Herder, P. M., Stikkelman, R. M., Dijkema, G. P., and Correljé, A. F. (2008). Design of a syngas infrastructure. *Computer Aided Chemical Engineering*, 25:223–228.

Breman, B. B. and Beenackers, A. A. (1996). Thermodynamic Models To Predict Gas Liquid Solubilities in the Methanol Synthesis, the MethanolHigher Alcohol Synthesis, and the Fischer Tropsch Synthesis via Gas Slurry Processes. *Industrial & Engineering Chemistry Research*, 35(10):3763–3775.

Chang, C. J., Chiu, K.-L., and Day, C.-Y. (1998). A new apparatus for the determination of P xy diagrams and Henry's constants in high pressure alcohols with critical carbon dioxide. *The Journal of Supercritical Fluids*, 12(3):223–237.

Chang, C. J., Day, C.-Y., Ko, C.-M., and Chiu, K.-L. (1997). Densities and P-x-y diagrams for carbon dioxide dissolution in methanol, ethanol, and acetone mixtures. *Fluid Phase Equilibria*, 131(1):243–258.

Chang, T. and Rousseau, R. W. (1985). Solubilities of carbon dioxide in methanol and methanol-water at high pressures: experimental data and modeling. *Fluid Phase Equilibria*, 23(2):243–258.

Downs, J. J. and Skogestad, S. (2011). An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35(1):99–110.

Fiedler, E., Grossman, G., Kersebohm, B., Weiss, G., and Witte, C. (2005). Methanol. In *Ullmann's Encyclopedia of Industrial Chemistry*. Wiley-VCH Verlag GmbH & Co.

Graaf, G., Sijtsema, P., Stamhuis, E., and Joosten, G. (1986). Chemical equilibria in methanol synthesis. *Chemical Engineering Science*, 41(11):2883–2890.

Graaf, G., Stamhuis, E., and Beenackers, A. (1988). Kinetics of low-pressure methanol synthesis. *Chemical Engineering Science*, 43(12):3185–3195.

Hamelinck, C. N. and Faaij, A. P. (2002). Future prospects for production of methanol and hydrogen from biomass. *Journal of Power Sources*, 111(1):1–22.

Hitachi Plant Systems (2012). Accomplishment of Centrifugal Compressor with Inlet Guide Vane for Methanol Synthesis Plant.

Jacobsen, M. G. (2011). *Identifying active constrain regions for optimal operation of process plants*. PhD thesis, Norwegian University of Science and Technology.

Joung, S. N., Yoo, C. W., Shin, H. Y., Kim, S. Y., Yoo, K.-P., Lee, C. S., and Huh, W. S. (2001). Measurements and correlation of high-pressure VLE of binary CO2 alcohol systems (methanol, ethanol, 2-methoxyethanol and 2-ethoxyethanol). *Fluid Phase Equilibria*, 185(1):219–230.

Kim, W. S., Yang, D. R., Moon, D. J., and Ahn, B. S. (2013). The process design and simulation for the methanol production on the FPSO (floating production, storage and off-loading) system. *Chemical Engineering Research and Design*.

Klier, K., Chatikavanij, V., Herman, R., and Simmons, G. (1982). Catalytic synthesis of methanol from CO/H2 , The effects of carbon dioxide. *Journal of Catalysis*, 74(2):343–360.

Kralj, A. K. and Glavič, P. (2009). Multi-criteria optimization in a methanol process. *Applied Thermal Engineering*, 29(5):1043–1049.

Lange, J.-P. (2001). Methanol synthesis: a short review of technology improvements. *Catalysis Today*, 64(1):3–8.

Lim, H.-W., Park, M.-J., Kang, S.-H., Chae, H.-J., Bae, J. W., and Jun, K.-W. (2009). Modeling of the Kinetics for Methanol Synthesis using Cu/ZnO/Al 2 O 3 /ZrO 2 Catalyst: Influence of Carbon Dioxide during Hydrogenation. *Industrial & Engineering Chemistry Research*, 48(23):10448–10455.

Løvik, I. (2001). *Modelling, Estimation and Optimization of the Methanol Synthesis with Catalyst Deactivation*. Doctoral thesis for the degree of philosophiae doctor, Norwegian University of Science and Technology.

Lüdtke, K. H. (2004). *Process Centrifugal Compressors: Basics, Function, Operation, Design, Application (Google eBook)*. Springer.

MathWorks (2013). fmincon.

Methanex (2013). Methanex - Methanol Price.

Minasidis, V., Jäschke, J., and Skogestad, S. (2013). Economic plantwide control: Automated controlled variable selection for a reactor-separator-recycle process. In *10th International Symposium on Dynamics and Control of Process Systems (submitted)*, Mumbai, India.

Moulijn, J. A., Makkee, M., and van Diepen, A. E. (2013). *Chemical Process Technology*. Wiley, 2nd editio edition.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition edition.

Noureldin, M. M. B., Elbashir, N. O., and El-Halwagi, M. M. (2013). Optimization and Selection of Reforming Approaches for Syngas Generation from Natural/Shale Gas. *Industrial & Engineering Chemistry Research*.

Ohsaki, H., Horiba, J., Hashizume, K., and Masutani, J. (2004). High Efficiency Mitsubishi Centrifugal Compressors and Steam Turbines for Large Methanol and DME Plants. *Mitsubishi Heavy Industries, Ltd.*

Oli Systems (2007). Oli Pro Customization Guide.

Panahi, M. (2011). *Plantwide Control for Economically Optimal Operation of Chemical Plants*. Doctoral thesis for the degree of philosophiae doctor, Norwegian University of Science and Technology.

Pellegrini, L. A., Soave, G., Gamba, S., and Langè, S. (2011). Economic analysis of a combined energy methanol production plant. *Applied Energy*, 88(12):4891–4897.

Riaz, A., Zahedi, G., and Klemeš, J. J. (2013). A review of cleaner production methods for the manufacture of methanol. *Journal of Cleaner Production*, 57:19–37.

Rostrup-Nielsen, J. and Christiansen, L. J. (2011). *Concepts in Syngas Manufacture*. World Scientific.

Shahrokhi, M. and Baghmisheh, G. (2005). Modeling, simulation and control of a methanol synthesis fixed-bed reactor. *Chemical Engineering Science*, 60(15):4275–4286.

Skogestad, S. (2000). Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507.

Skogestad, S. (2004). Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234.

Skogestad, S. (2012). Economic Plantwide Control. In Kariwala, Vinay Rangaiah, G. P., editor, *Plantwide Control*, chapter 11, pages 229–251. John Wiley & Sons, Ltd.

Speight, J. G. (2002). *Chemical and Process Design Handbook*. McGraw Hill.

Statistics Norway (2013). Electricity prices, Q2 2013.

Unisim (2007). Unisim Operations Guide.

Van-Dal, E. S. o. and Bouallou, C. (2013). Design and simulation of a methanol production plant from CO2 hydrogenation. *Journal of Cleaner Production*, 57:38–45.

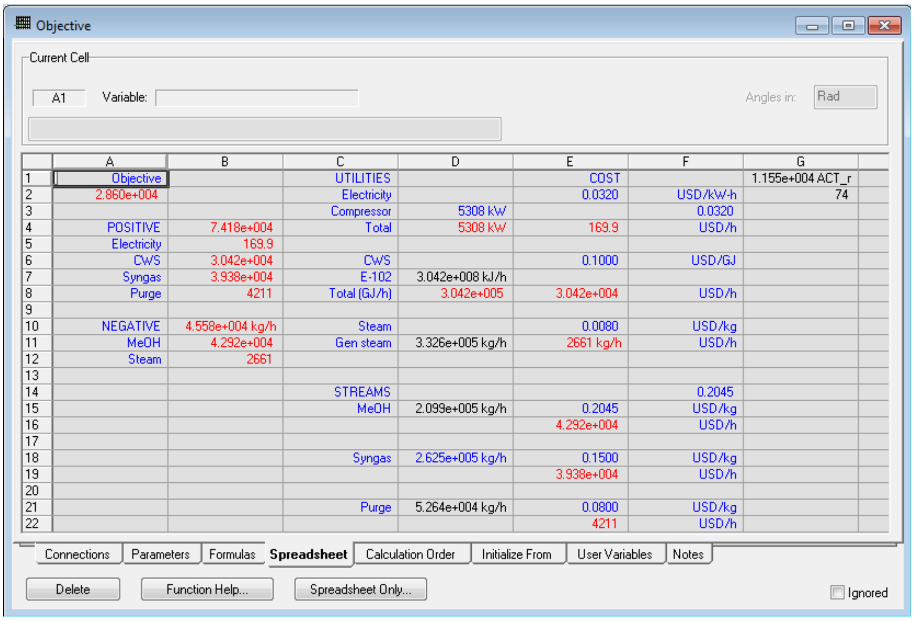
Vanden Bussche, K. and Froment, G. (1996). A Steady-State Kinetic Model for Methanol Synthesis and the Water Gas Shift Reaction on a Commercial Cu/ZnO/Al2O3Catalyst. *Journal of Catalysis*, 161(1):1–10.

Yoon, J. H., Lee, H. S., and Lee, H. (1993). High-pressure vapor-liquid equilibria for carbon dioxide + methanol, carbon dioxide + ethanol, and carbon dioxide + methanol + ethanol. *Journal of Chemical & Engineering Data*, 38(1):53–55.

Zhang, Y., Cruz, J., Zhang, S., Lou, H. H., and Benson, T. J. (2013). Process simulation and optimization of methanol production coupled to tri-reforming process. *International Journal of Hydrogen Energy*, 38(31):13617–13630.

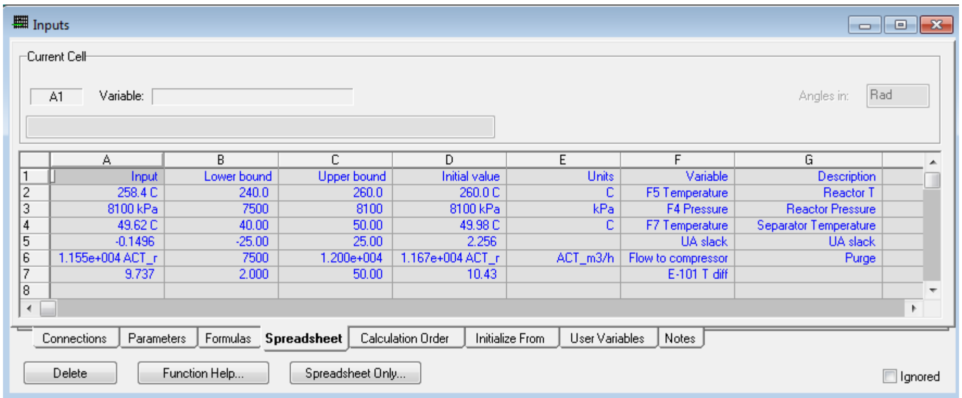
A UniSim setting

The layout of the spreadsheets used to modify values in the simulation is shown in this Appendix. COM works in the Objective, Input, and Constraints spreadsheets.



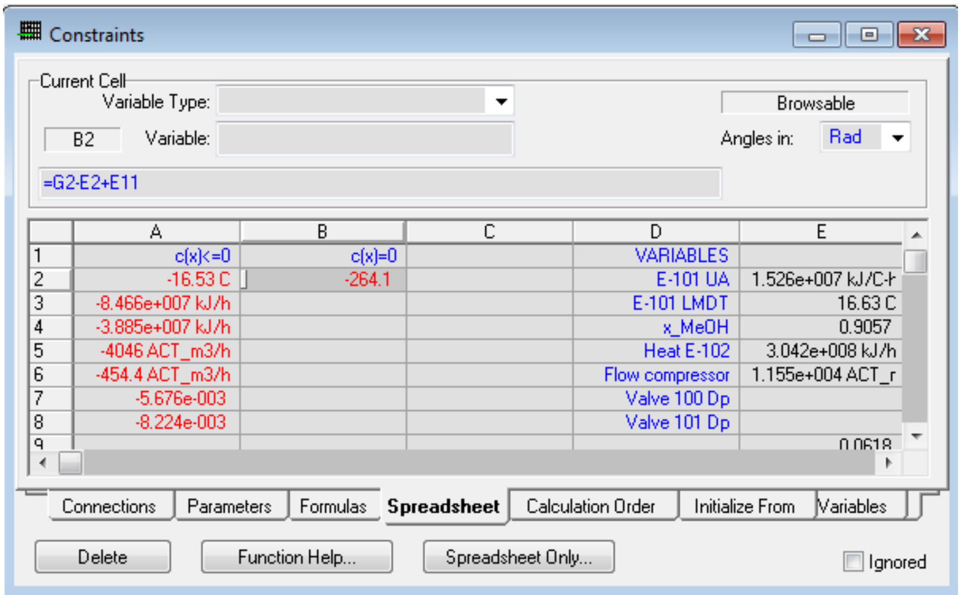
	A	B	C	D	E	F	G
1	Objective		UTILITIES		COST		1.155e+004 ACT_r
2	2.860e+004		Electricity		0.0320	USD/kW-h	74
3			Compressor	5308 kW		0.0320	
4	POSITIVE	7.418e+004	Total	5308 kW	169.9	USD/h	
5	Electricity	169.9					
6	CWS	3.042e+004	CWS		0.1000	USD/GJ	
7	Syngas	3.938e+004	E-102	3.042e+008 kJ/h			
8	Purge	4211	Total (GJ/h)	3.042e+005	3.042e+004	USD/h	
9							
10	NEGATIVE	4.558e+004 kg/h	Steam		0.0080	USD/kg	
11	MeOH	4.292e+004	Gen steam	3.326e+005 kg/h	2661 kg/h	USD/h	
12	Steam	2661					
13							
14			STREAMS			0.2045	
15			MeOH	2.099e+005 kg/h	0.2045	USD/kg	
16					4.292e+004	USD/h	
17							
18			Syngas	2.625e+005 kg/h	0.1500	USD/kg	
19					3.938e+004	USD/h	
20							
21			Purge	5.264e+004 kg/h	0.0800	USD/kg	
22					4211	USD/h	

Figure 12: Layout of Objective spreadsheet.



	A	B	C	D	E	F	G
1	Input	Lower bound	Upper bound	Initial value	Units	Variable	Description
2	258.4 C	240.0	260.0	260.0 C	C	F5 Temperature	Reactor T
3	8100 kPa	7500	8100	8100 kPa	kPa	F4 Pressure	Reactor Pressure
4	49.62 C	40.00	50.00	49.98 C	C	F7 Temperature	Separator Temperature
5	-0.1496	-25.00	25.00	2.256		UA slack	UA slack
6	1.155e+004 ACT_r	7500	1.200e+004	1.167e+004 ACT_r	ACT_m3/h	Flow to compressor	Purge
7	9.737	2.000	50.00	10.43		E-101 T diff	
8							

Figure 13: Layout of Inputs spreadsheet.

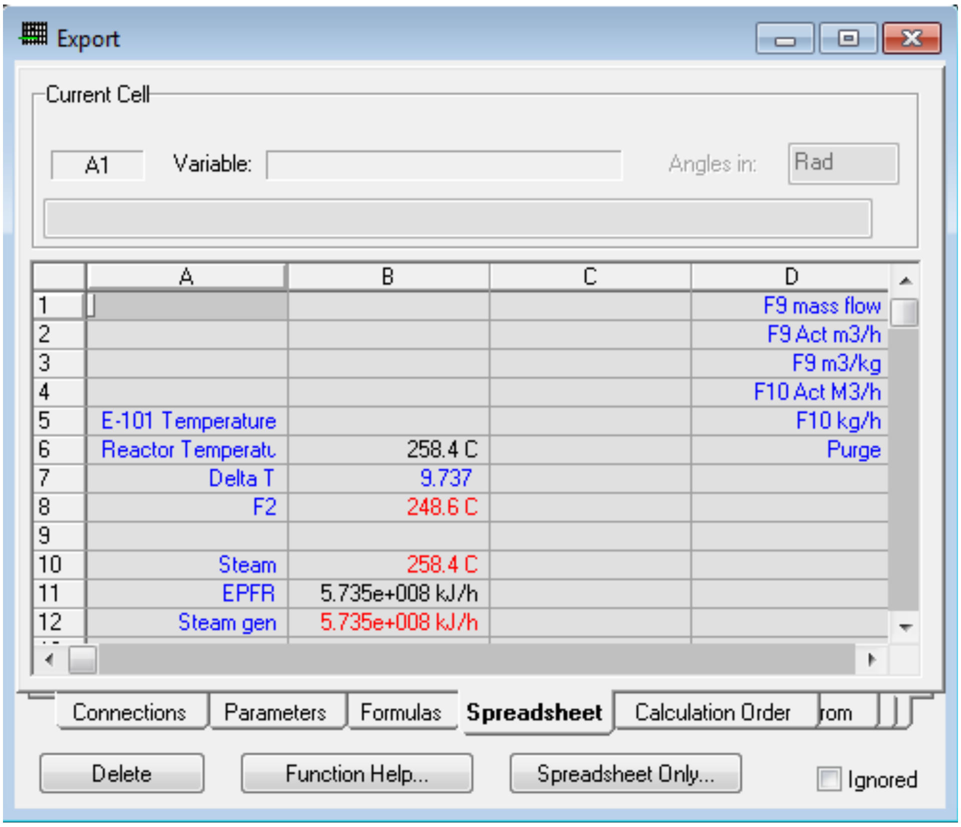


B2 Variable:

=G2-E2+E11

	A	B	C	D	E
1	c(x)<=0	c(x)=0		VARIABLES	
2	-16.53 C	-264.1		E-101 UA	1.526e+007 kJ/C-h
3	-8.466e+007 kJ/h			E-101 LMDT	16.63 C
4	-3.885e+007 kJ/h			x_MeOH	0.9057
5	-4046 ACT_m3/h			Heat E-102	3.042e+008 kJ/h
6	-454.4 ACT_m3/h			Flow compressor	1.155e+004 ACT_r
7	-5.676e-003			Valve 100 Dp	
8	-8.224e-003			Valve 101 Dp	
9					0.0618

Figure 14: Layout of Constraints spreadsheet.



A1 Variable:

	A	B	C	D
1				F9 mass flow
2				F9 Act m3/h
3				F9 m3/kg
4				F10 Act M3/h
5	E-101 Temperature			F10 kg/h
6	Reactor Temperatu	258.4 C		Purge
7	Delta T	9.737		
8	F2	248.6 C		
9				
10	Steam	258.4 C		
11	EPFR	5.735e+008 kJ/h		
12	Steam gen	5.735e+008 kJ/h		

Figure 15: Layout of spreadsheet for conversions - for transformation of Inputs.

B Optimization Results

For each table, syngas make-up flow is varying. Numbers in bold are at constraint value. Pressure in [kPa], temperature in [°C], concentrations in [mol fraction], mass flows in [kg/h], F9 flow in [actual m^3/h].

Table 9: Optimization at nominal syngas concentration.

Syngas flow [kg/h]	212500	225000	237500	250000	262500	275000	287500
H2	0.6300	0.6300	0.6300	0.6300	0.6300	0.6300	0.6300
CO	0.3100	0.3100	0.3100	0.3100	0.3100	0.3100	0.3100
CO2	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500
Methane	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
Costs [$/h]							
Objective	21007	22087	23771	25870	28604	31589	34973
Electricity	150	150	161	172	170	165	155
CWS	24184	25461	27216	29283	30425	31307	31455
Syngas	31875	33750	35625	37500	39375	41250	43125
Purge	2954	3116	3319	3549	4211	5011	6118
Steam	2252	2346	2466	2582	2661	2717	2727
Product	35904	38045	40083	42052	42916	43427	43152
Input constraints							
F5 Temperature	259	253	254	256	258	**260**	**260**
F4 Pressure	8038	8073	**8100**	**8100**	**8100**	**8100**	**8100**
F7 Temperature	**50.0**	**50.0**	**50.0**	**50.0**	49.6	49.5	49.2
F9 Flow	**7500**	**7500**	8809	10902	11546	**12000**	11410
UA slack	-0.35	-0.45	0.28	0.21	0.82	0.23	-0.05
E-101 delta T	4.61	4.23	5.87	8.63	9.74	10.67	10.43
Output constraints							
E-101 UA	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07
x_MeOH	**0.90**	**0.90**	**0.90**	**0.90**	0.91	0.91	0.92
Heat E-102	2.42E+08	2.55E+08	2.72E+08	2.93E+08	3.04E+08	3.13E+08	3.15E+08
CH4 concentration	**0.070**	**0.070**	**0.070**	**0.070**	0.062	0.054	0.045
LMDT	12.30	12.11	13.63	15.83	16.63	17.19	16.74
Input relative values							
F5 Temperature	0.95	0.66	0.70	0.81	0.92	1.00	1.00
F4 Pressure	0.90	0.96	1.00	1.00	1.00	1.00	1.00
F7 Temperature	1.00	1.00	1.00	1.00	0.96	0.95	0.92
UA slack	0.49	0.49	0.50	0.50	0.52	0.50	0.50
Flow to compressor	0.00	0.00	0.29	0.76	0.90	1.00	0.87
E-101 T diff	0.05	0.05	0.08	0.14	0.16	0.18	0.18

Table 10: Optimization at 80% of nominal methane concentration in syngas

Syngas flow [kg/h]	212500	225000	237500	250000	262500	275000	287500
H2	0.6313	0.6313	0.6313	0.6313	0.6313	0.6313	0.6313
CO	0.3106	0.3106	0.3106	0.3106	0.3106	0.3106	0.3106
CO2	0.0501	0.0501	0.0501	0.0501	0.0501	0.0501	0.0501
Methane	0.0080	0.0080	0.0080	0.0080	0.0080	0.0080	0.0080
Costs [$/h]							
Objective	20514	21789	23337	25275	27959	31009	34470
Electricity	147	150	158	171	175	167	158
CWS	24314	25566	27205	29305	31239	31868	31903
Syngas	31875	33750	35625	37500	39375	41250	43125
Purge	2772	3006	3206	3383	3811	4701	5860
Steam	2224	2355	2483	2606	2703	2757	2762
Product	36370	38327	40373	42477	43938	44220	43814
Input constraints							
F5 Temperature	251	252	254	256	258	**260**	**260**
F4 Pressure	7766	8022	**8100**	**8100**	**8100**	**8100**	**8100**
F7 Temperature	**50.0**	**50.0**	**50.0**	**50.0**	47.4	47.3	48.3
F9 Flow	**7500**	**7500**	8512	10596	**12000**	**12000**	11578
UA slack	0.57	-0.07	0.16	-0.58	-0.01	-0.01	0.01
E-101 delta T	3.79	4.05	5.38	8.08	9.97	10.38	10.38
Output constraints							
E-101 UA	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07
x_MeOH	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**	0.91	0.92
Heat E-102	2.43E+08	2.56E+08	2.72E+08	2.93E+08	3.12E+08	3.19E+08	3.19E+08
CH4 concentration	0.060	0.058	0.058	0.058	0.055	0.046	0.038
LMDT	11.42	11.96	13.30	15.52	17.21	17.40	16.95
Input relative values							
F5 Temperature	0.55	0.62	0.70	0.80	0.91	1.00	1.00
F4 Pressure	0.44	0.87	1.00	1.00	1.00	1.00	1.00
F7 Temperature	1.00	1.00	1.00	1.00	0.74	0.73	0.83
UA slack	0.51	0.50	0.50	0.49	0.50	0.50	0.50
Flow to compressor	0.00	0.00	0.22	0.69	1.00	1.00	0.91
E-101 T diff	0.04	0.04	0.07	0.13	0.17	0.17	0.17

Table 11: Optimization at 90% of nominal nominal methane concentration in syngas.

Syngas flow [kg/h]	212500	225000	237500	250000	262500	275000	287500
H2	0.6306	0.6306	0.6306	0.6306	0.6306	0.6306	0.6306
CO	0.3103	0.3103	0.3103	0.3103	0.3103	0.3103	0.3103
CO2	0.0501	0.0501	0.0501	0.0501	0.0501	0.0501	0.0501
Methane	0.0090	0.0090	0.0090	0.0090	0.0090	0.0090	0.0090
Costs [$/h]							
Objective	20640	21926	23538	25562	28279	31299	34714
Electricity	148	150	160	174	174	166	156
CWS	24267	25516	27230	29410	31111	31571	31650
Syngas	31875	33750	35625	37500	39375	41250	43125
Purge	2818	3056	3252	3430	3933	4860	5995
Steam	2218	2348	2475	2596	2687	2736	2744
Product	36251	38199	40255	42356	43627	43812	43468
Input constraints							
F5 Temperature	251	252	254	256	258	**260**	**260**
F4 Pressure	7793	8047	**8100**	**8100**	**8100**	**8100**	**8100**
F7 Temperature	**50.0**	**50.0**	**50.0**	**50.0**	47.5	48.5	48.9
F9 Flow	**7500**	**7501**	8691	10935	**11998**	**12000**	11494
UA slack	0.6408	0.1781	-0.1001	-0.6757	0.0407	-1.0700	-0.2052
E-101 delta T	3.85	4.11	5.65	8.58	10.05	10.53	10.41
Output constraints							
E-101 UA	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07
x_MeOH	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**	0.91	0.916
Heat E-102	2.43E+08	2.55E+08	2.72E+08	2.94E+08	3.11E+08	3.16E+08	3.16E+08
CH4 concentration	0.066	0.064	0.064	0.065	0.060	0.050	0.042
LMDT	11.46	12.01	13.49	15.84	17.21	17.29	16.83
Input relative values							
F5 Temperature	0.55	0.62	0.70	0.81	0.91	1.00	1.00
F4 Pressure	0.49	0.91	1.00	1.00	1.00	1.00	1.00
F7 Temperature	1.00	1.00	1.00	1.00	0.75	0.85	0.89
UA slack	0.51	0.50	0.50	0.49	0.50	0.48	0.50
Flow to compressor	0.00	0.00	0.26	0.76	1.00	1.00	0.89
E-101 T diff	0.04	0.04	0.08	0.14	0.17	0.18	0.18

Table 12: Optimization at methane concentration in syngas increased 10% relative to nominal value.

Syngas flow [kg/h]	212500	225000	237500	250000	262500	275000	287500
H2	0.6294	0.6294	0.6294	0.6294	0.6294	0.6294	0.6294
CO	0.3097	0.3097	0.3097	0.3097	0.3097	0.3097	0.3097
CO2	0.0499	0.0499	0.0499	0.0499	0.0499	0.0499	0.0499
Methane	0.0110	0.0110	0.0110	0.0110	0.0110	0.0110	0.0110
Costs [$/h]							
Objective	21803	22907	24317	26262	28892	31868	35223
Electricity	145	147	151	160	168	164	154
CWS	23872	25172	26635	28532	30214	31123	31205
Syngas	31875	33750	35625	37500	39375	41250	43125
Purge	3258	3428	3630	3866	4347	5137	6254
Steam	2222	2343	2435	2554	2644	2700	2710
Product	35127	37248	39288	41241	42567	43105	42805
Input constraints							
F5 Temperature	**260**	258	254	256	258	**260**	**260**
F4 Pressure	7950	8098	**8100**	**8100**	**8100**	**8100**	**8100**
F7 Temperature	**50.0**	**50.0**	**50.0**	**50.0**	**50.0**	**50.0**	**50.0**
F9 Flow	**7500**	**7500**	8135	9913	11456	**12000**	11359
UA slack	3.8392	0.0569	-0.0285	-0.2365	-0.5920	0.0723	-0.1435
E-101 delta T	4.74	4.72	5.15	7.48	9.73	10.79	10.50
Output constraints							
E-101 UA	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07
x_MeOH	**0.90**	**0.90**	**0.90**	**0.90**	0.91	0.91	0.917
Heat E-102	2.39E+08	2.52E+08	2.66E+08	2.85E+08	3.02E+08	3.11E+08	3.12E+08
CH4 concentration	**0.070**	**0.070**	**0.070**	**0.070**	0.066	0.058	0.049
LMDT	12.24	12.48	12.97	14.95	16.53	17.14	16.63
Input relative values							
F5 Temperature	1.00	0.92	0.70	0.81	0.92	1.00	1.00
F4 Pressure	0.75	1.00	1.00	1.00	1.00	1.00	1.00
F7 Temperature	1.00	1.00	1.00	1.00	1.00	1.00	1.00
UA slack	0.58	0.50	0.50	0.50	0.49	0.50	0.50
Flow to compressor	0.00	0.00	0.14	0.54	0.88	1.00	0.86
E-101 T diff	0.06	0.06	0.07	0.11	0.16	0.18	0.18

Table 13: Optimization at methane concentration in syngas increased 20% relative to nominal value.

Syngas flow [kg/h]	212500	225000	237500	250000	262500	275000	287500
H2	0.6287	0.6287	0.6287	0.6287	0.6287	0.6287	0.6287
CO	0.3094	0.3094	0.3094	0.3094	0.3094	0.3094	0.3094
CO2	0.0499	0.0499	0.0499	0.0499	0.0499	0.0499	0.0499
Methane	0.0120	0.0120	0.0120	0.0120	0.0120	0.0120	0.0120
Costs [$/h]							
Objective	22590	23731	24961	26772	29160	32142	35442
Electricity	140	144	144	152	167	161	154
CWS	23546	24854	26130	27886	30062	30766	31109
Syngas	31875	33750	35625	37500	39375	41250	43125
Purge	3562	3743	3947	4184	4462	5310	6339
Steam	2184	2314	2405	2521	2629	2682	2697
Product	34349	36445	38479	40428	42275	42663	42588
Input constraints							
F5 Temperature	**260**	**260**	255	256	258	**260**	**260**
F4 Pressure	7807	8041	8093	**8100**	**8100**	**8100**	**8100**
F7 Temperature	**50.0**	**50.0**	*49.9*	**50.0**	**50.0**	**50.0**	**50.0**
F9 Flow	**7500**	**7500**	7555	9074	11376	11540	11333
UA slack	0.1183	-0.7434	0.0743	-0.1315	-0.3125	-0.5040	-0.1055
E-101 delta T	4.78	4.94	4.67	6.58	9.72	10.38	10.56
Output constraints							
E-101 UA	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07	1.53E+07
x_MeOH	0.91	0.91	0.91	0.91	0.91	0.91	0.918
Heat E-102	2.35E+08	2.49E+08	2.61E+08	2.79E+08	3.01E+08	3.08E+08	3.11E+08
CH4 concentration	**0.070**	**0.070**	**0.070**	**0.070**	**0.070**	0.061	0.052
LMDT	12.05	12.50	12.41	14.15	16.47	16.77	16.62
Input relative values							
F5 Temperature	1.00	1.00	0.74	0.80	0.92	1.00	1.00
F4 Pressure	0.51	0.90	0.99	1.00	1.00	1.00	1.00
F7 Temperature	1.00	1.00	0.99	1.00	1.00	1.00	1.00
UA slack	0.50	0.49	0.50	0.50	0.49	0.49	0.50
Flow to compressor	0.00	0.00	0.01	0.35	0.86	0.90	0.85
E-101 T diff	0.06	0.06	0.06	0.10	0.16	0.17	0.18

C MATLAB Code

C.1 NLP-Generator

```
clc
clear all
close all

%Declare global variables
global h;
global hyCase;
global f;

h = actxserver('UnisimDesign.Application');
hyCase = h.Activedocument;
f = hyCase.Flowsheet;

%%
% Count the inputs
i=0;
while f.Operations.Item('Inputs').Cell(strcat(['A' int2str(i+2)])).CellVariable.IsKnown
    i=i+1;
end

numberOfInputs=i;

if numberOfInputs<1
    error('Warning, no inputs');
end

inputIndices= [ 2:1:numberOfInputs+1;
                2:1:numberOfInputs+1];
inputIndices2= [ 1:1:numberOfInputs;
                2:1:numberOfInputs+1];
inputIndices3=[ 2:1:numberOfInputs+1;
                1:1:numberOfInputs;
                2:1:numberOfInputs+1];
inputIndices4= [ 1:1:numberOfInputs;
                2:1:numberOfInputs+1;
                1:1:numberOfInputs];

% Read the upper and lower bounds plus the initial values
inputLB=zeros(numberOfInputs,1);
inputUB=zeros(numberOfInputs,1);
initialUs=zeros(numberOfInputs,1);

parameter.inputUnits=cell(numberOfInputs,1);

for i=1:1:numberOfInputs
    inputLB(i)=f.Operations.Item('Inputs').Cell(strcat(['B' int2str(i+1)])).CellVariable.Value;
    inputUB(i)=f.Operations.Item('Inputs').Cell(strcat(['C' int2str(i+1)])).CellVariable.Value;
    initialUs(i)=f.Operations.Item('Inputs').Cell(strcat(['D' int2str(i+1)])).CellVariable.Value;
    parameter.inputUnits(i,1)=cellstr(f.Operations.Item('Inputs').Cell(strcat(['E' int2str(i+1)])).CellText);
end

lb=strcat('lb = [',num2str(inputLB'),']'';\n');
ub=strcat('ub = [',num2str(inputUB'),']'';\n');
u0=strcat('u0 = [',num2str(initialUs'),']'';\n');
%%
fName = 'NLP4MATLAB.m';          %# A file name
fid = fopen(fName,'w');          %# Open the file

%Create the opt
if fid ~= -1
    fprintf(fid,...
        strcat(...
        'function [u_opt,fval,exitflag] = NLP4MATLAB()\n\n',...
        'clc\n',...
        'clear all\n',...
        'close all\n\n',...
        'global h;\n',...
        'global f;\n',...
        'global hyCase;\n',...
        'h = actxserver(''UnisimDesign.Application'');\n',...
        'hyCase = h.Activedocument;\n',...
        'f = hyCase.Flowsheet;\n\n',...
        'par=[];\n'));
%      fprintf(fid,'lb(%d) = f.Operations.Item(''Inputs'').Cell(''B%d'').CellVariable.Value;\n',inputIndices2);
%      fprintf(fid,'ub(%d) = f.Operations.Item(''Inputs'').Cell(''C%d'').CellVariable.Value;\n',inputIndices2);
    fprintf(fid,'lb=zeros(1,%d);\nub=ones(1,%d);\n',numberOfInputs,numberOfInputs);
    fprintf(fid,'u0(%d) = scaleInputs(f.Operations.Item(''Inputs'').Cell(''D%d'').CellVariable.Value,lb,ub,1);\n',inputIndices2);
    fprintf(fid,...
        strcat(...
        'options = optimset(''TolFun'',10e-8,''TolCon'',1e-4,''Display'',''iter'',''Algorithm'',''interior-point'',''Diagnostics'',''on'', ''FinDiffType'',''central'',''ScaleProblem'',''obj-and-constr'',''FinDiffRelStep'',1e-2);\n',...
        'tic\n',...
        '[u_opt,fval,exitflag]=fmincon(@(u)objFun(u,par),u0,[],[],[],[],lb,ub,@(u)nonLinConFun(u,par),options);\n',...
        'toc\n',...
        'end\n\n'...
        ));
end
```

```matlab
%%
if fid ~= -1
    fprintf(fid,...
        strcat(...
        'function y = objFun(u,par)\n',...
        'global f;\n',...
        'global hyCase;\n',...
        'hyCase.Solver.CanSolve=0;\n'...
        ));
    fprintf(fid,'lb(%d) = f.Operations.Item(''Inputs'').Cell(''B%d'').CellVariable.Value;\n',inputIndices2);
    fprintf(fid,'ub(%d) = f.Operations.Item(''Inputs'').Cell(''C%d'').CellVariable.Value;\n',inputIndices2);
    fprintf(fid,'input%dUnits = f.Operations.Item(''Inputs'').Cell(''E%d'').CellText;\n',inputIndices);
    fprintf(fid,'f.Operations.Item(''Inputs'').Cell(''A%d'').CellVariable.SetValue(deScaleInputs(u,lb,ub,%d),input%dUnits);\n',inputIndices3);

    fprintf(fid,...
        strcat(...
        'hyCase.Solver.CanSolve=1;\n',...
        'y = f.Operations.Item(''Objective'').Cell(''A2'').CellValue;\n',...
        'end\n\n'...
        ));
end
%%
% Count the constraints
i=0;
while f.Operations.Item('Constraints').Cell(strcat(['A' int2str(i+2)])).CellVariable.IsKnown
    i=i+1;
end
numberOfInequalityConstraints=i;

i=0;
while f.Operations.Item('Constraints').Cell(strcat(['B' int2str(i+2)])).CellVariable.IsKnown
    i=i+1;
end
numberOfEqualityConstraints=i;

ineqConIndices=[2:1:numberOfInequalityConstraints+1;2:1:numberOfInequalityConstraints+1];
eqConIndices=[2:1:numberOfEqualityConstraints+1;2:1:numberOfEqualityConstraints+1];

if fid ~= -1
    fprintf(fid,...
        strcat(...
        'function [c,ceq] = nonLinConFun(u,par)\n',...
        'global f;\n',...
        'global hyCase;\n',...
        'hyCase.Solver.CanSolve=0;\n'...
        ));
    fprintf(fid,'lb(%d) = f.Operations.Item(''Inputs'').Cell(''B%d'').CellVariable.Value;\n',inputIndices2);
    fprintf(fid,'ub(%d) = f.Operations.Item(''Inputs'').Cell(''C%d'').CellVariable.Value;\n',inputIndices2);
    fprintf(fid,'input%dUnits = f.Operations.Item(''Inputs'').Cell(''E%d'').CellText;\n',inputIndices);
    fprintf(fid,'f.Operations.Item(''Inputs'').Cell(''A%d'').CellVariable.SetValue(deScaleInputs(u,lb,ub,%d),input%dUnits);\n',inputIndices3);

    fprintf(fid,'hyCase.Solver.CanSolve=1;\n');

    if numberOfInequalityConstraints>0
        fprintf(fid,'c(%d)=f.Operations.Item(''Constraints'').Cell(''A%d'').CellValue;\n',ineqConIndices);
    else
        fprintf(fid,'c=[];\n');
    end
    if numberOfEqualityConstraints>0
        fprintf(fid,'ceq(%d)=f.Operations.Item(''Constraints'').Cell(''B%d'').CellValue;\n',eqConIndices);
    else
        fprintf(fid,'ceq=[];\n');
    end
    fprintf(fid,'end\n\n');
end
%%
if fid ~= -1
    fprintf(fid,...
        strcat(...
        'function y = scaleInputs(u,lb,ub,index)\n',...
        'y=(u(index)-lb(index))/(ub(index)-lb(index));\n',...
        'end\n\n'...
        ));
end

%%
if fid ~= -1
    fprintf(fid,...
        strcat(...
        'function y = deScaleInputs(u,lb,ub,index)\n',...
        'y=lb(index)+u(index)*(ub(index)-lb(index));\n',...
        'end\n\n'...
        ));
end

fclose(fid);
```

C.2 NLP Example code

```matlab
function [u_opt,fval,exitflag] = NLP4MATLAB()

clc
clear all
close all

global h;
global f;
global hyCase;
h = actxserver('UnisimDesign.Application');
hyCase = h.Activedocument;
f = hyCase.Flowsheet;

par=[];
lb=zeros(1,6);
ub=ones(1,6);
u0(1) = scaleInputs(f.Operations.Item('Inputs').Cell('D2').CellVariable.Value,lb,ub,1);
u0(2) = scaleInputs(f.Operations.Item('Inputs').Cell('D3').CellVariable.Value,lb,ub,1);
u0(3) = scaleInputs(f.Operations.Item('Inputs').Cell('D4').CellVariable.Value,lb,ub,1);
u0(4) = scaleInputs(f.Operations.Item('Inputs').Cell('D5').CellVariable.Value,lb,ub,1);
u0(5) = scaleInputs(f.Operations.Item('Inputs').Cell('D6').CellVariable.Value,lb,ub,1);
u0(6) = scaleInputs(f.Operations.Item('Inputs').Cell('D7').CellVariable.Value,lb,ub,1);
options = optimset('TolFun',10e-8,'TolCon',1e-4,'Display','iter','Algorithm','interior-point','Diagnostics','on', 'FinDiffType','central','ScaleProblem','obj-and-constr','FinDiffRelStep',1e-2);
tic
[u_opt,fval,exitflag]=fmincon(@(u)objFun(u,par),u0,[],[],[],[],lb,ub,@(u)nonLinConFun(u,par),options);
toc
end

function y = objFun(u,par)
global f;
global hyCase;
hyCase.Solver.CanSolve=0;
lb(1) = f.Operations.Item('Inputs').Cell('B2').CellVariable.Value;
lb(2) = f.Operations.Item('Inputs').Cell('B3').CellVariable.Value;
lb(3) = f.Operations.Item('Inputs').Cell('B4').CellVariable.Value;
lb(4) = f.Operations.Item('Inputs').Cell('B5').CellVariable.Value;
lb(5) = f.Operations.Item('Inputs').Cell('B6').CellVariable.Value;
lb(6) = f.Operations.Item(’Inputs’).Cell(’B7’).CellVariable.
    Value;
ub(1) = f.Operations.Item(’Inputs’).Cell(’C2’).CellVariable.
    Value;
ub(2) = f.Operations.Item(’Inputs’).Cell(’C3’).CellVariable.
    Value;
ub(3) = f.Operations.Item(’Inputs’).Cell(’C4’).CellVariable.
    Value;
ub(4) = f.Operations.Item(’Inputs’).Cell(’C5’).CellVariable.
    Value;
ub(5) = f.Operations.Item(’Inputs’).Cell(’C6’).CellVariable.
    Value;
ub(6) = f.Operations.Item(’Inputs’).Cell(’C7’).CellVariable.
    Value;
input2Units = f.Operations.Item(’Inputs’).Cell(’E2’).CellText;
input3Units = f.Operations.Item(’Inputs’).Cell(’E3’).CellText;
input4Units = f.Operations.Item(’Inputs’).Cell(’E4’).CellText;
input5Units = f.Operations.Item(’Inputs’).Cell(’E5’).CellText;
input6Units = f.Operations.Item(’Inputs’).Cell(’E6’).CellText;
input7Units = f.Operations.Item(’Inputs’).Cell(’E7’).CellText;
f.Operations.Item(’Inputs’).Cell(’A2’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,1),input2Units);
f.Operations.Item(’Inputs’).Cell(’A3’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,2),input3Units);
f.Operations.Item(’Inputs’).Cell(’A4’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,3),input4Units);
f.Operations.Item(’Inputs’).Cell(’A5’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,4),input5Units);
f.Operations.Item(’Inputs’).Cell(’A6’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,5),input6Units);
f.Operations.Item(’Inputs’).Cell(’A7’).CellVariable.SetValue(
    deScaleInputs(u,lb,ub,6),input7Units);
hyCase.Solver.CanSolve=1;
y = f.Operations.Item(’Objective’).Cell(’A2’).CellValue;
end

function [c,ceq] = nonLinConFun(u,par)
global f;
global hyCase;
hyCase.Solver.CanSolve=0;
lb(1) = f.Operations.Item(’Inputs’).Cell(’B2’).CellVariable.
    Value;
lb(2) = f.Operations.Item(’Inputs’).Cell(’B3’).CellVariable.
    Value;
lb(3) = f.Operations.Item(’Inputs’).Cell(’B4’).CellVariable.
    Value;
lb(4) = f.Operations.Item(’Inputs’).Cell(’B5’).CellVariable.
    Value;
lb(5) = f.Operations.Item(’Inputs’).Cell(’B6’).CellVariable.
    Value;
lb(6) = f.Operations.Item(’Inputs’).Cell(’B7’).CellVariable.
    Value;
ub(1) = f.Operations.Item(’Inputs’).Cell(’C2’).CellVariable.
    Value;
```

```
ub(2) = f.Operations.Item('Inputs').Cell('C3').CellVariable.Value;
ub(3) = f.Operations.Item('Inputs').Cell('C4').CellVariable.Value;
ub(4) = f.Operations.Item('Inputs').Cell('C5').CellVariable.Value;
ub(5) = f.Operations.Item('Inputs').Cell('C6').CellVariable.Value;
ub(6) = f.Operations.Item('Inputs').Cell('C7').CellVariable.Value;
input2Units = f.Operations.Item('Inputs').Cell('E2').CellText;
input3Units = f.Operations.Item('Inputs').Cell('E3').CellText;
input4Units = f.Operations.Item('Inputs').Cell('E4').CellText;
input5Units = f.Operations.Item('Inputs').Cell('E5').CellText;
input6Units = f.Operations.Item('Inputs').Cell('E6').CellText;
input7Units = f.Operations.Item('Inputs').Cell('E7').CellText;
f.Operations.Item('Inputs').Cell('A2').CellVariable.SetValue(deScaleInputs(u,lb,ub,1),input2Units);
f.Operations.Item('Inputs').Cell('A3').CellVariable.SetValue(deScaleInputs(u,lb,ub,2),input3Units);
f.Operations.Item('Inputs').Cell('A4').CellVariable.SetValue(deScaleInputs(u,lb,ub,3),input4Units);
f.Operations.Item('Inputs').Cell('A5').CellVariable.SetValue(deScaleInputs(u,lb,ub,4),input5Units);
f.Operations.Item('Inputs').Cell('A6').CellVariable.SetValue(deScaleInputs(u,lb,ub,5),input6Units);
f.Operations.Item('Inputs').Cell('A7').CellVariable.SetValue(deScaleInputs(u,lb,ub,6),input7Units);
hyCase.Solver.CanSolve=1;
c(2)=f.Operations.Item('Constraints').Cell('A2').CellValue;
c(3)=f.Operations.Item('Constraints').Cell('A3').CellValue;
c(4)=f.Operations.Item('Constraints').Cell('A4').CellValue;
c(5)=f.Operations.Item('Constraints').Cell('A5').CellValue;
c(6)=f.Operations.Item('Constraints').Cell('A6').CellValue;
c(7)=f.Operations.Item('Constraints').Cell('A7').CellValue;
c(8)=f.Operations.Item('Constraints').Cell('A8').CellValue;
ceq(2)=f.Operations.Item('Constraints').Cell('B2').CellValue;
end

function y = scaleInputs(u,lb,ub,index)
y=(u(index)-lb(index))/(ub(index)-lb(index));
end

function y = deScaleInputs(u,lb,ub,index)
y=lb(index)+u(index)*(ub(index)-lb(index));
end
```