NTNU

Department of Engineering Cybernetics

Final Year Project

A simulated debutanizer case study for teaching advanced process control

Author:
Morten Bakke

Supervisors:
Professor Sigurd Skogestad
MSc Bjørn Einar Bjartnes

Project description

In terms of control theory, a gap between academica and industry will always be present. Different abbrevations and naming conventions are maybe the main differences between the two, while the control theory remains the same. Whenever a newly qualified student enters the industry, chances are that the student have no idea what his or hers coworkers are talking about in terms of control because of these differences. One thing that academia and the industry do have in common, is that they both uses simulators in order to test different control configurations and control performance.

This project is motivated by Honeywell, who as of the summer of 2008 provided NTNU with licenses to their simulation tool *UniSim.* During the fall of 2008, Honeywell also provided NTNU with licenses to the software *Profit Design Studio*, which is the same software Honeywell uses for identification and implementation of model predictive controllers. The intention of this project is to develop a case study meant for teaching advanced process control, where the students gets familiar with the ”industry language” and uses a simulation environment that are closer to a ”real” process. This is intended to be a supplement to the classical lectures and assignments, not as a substitute for these. The main objectives of this project are:

- Develop a case study for teaching purposes by using Honeywells simulation tool UniSim. Theory from basic and advanced process control should be used, while the students should get some insight into the differences between industry and academia.

- In collaboration with Honeywell, develop an MPC controller by using *Profit Design Studio* to be used in the case study

- Draft a solution to the case study

- If possible, test the case study in a process control course at NTNU during the fall of 2008

Abstract

The concrete results of this project are two assignments with solutions, that are intended to be used for academic teaching purposes. The two assignments together make up a case study, in which control of a distillation column (debutanizer) using the LV-configuration is the main objective. For dynamic simulations, Honeywells simulation tool UniSim has been utilized, and for building a multivariable controller (MPC) Honeywells Profit Design Studio was used. The emphasis has been put on the use of control theory in a somewhat more practical setting than standard academic assignments provide. All files needed for these exercises can be found on Sigurd Skogestads homepage (http://www.nt.ntnu.no/users/skoge/vgprosessregulering/ - exercises).

The assignments have been given to students who attended prof. Sigurd Skogestads course "Advanced Process Control" at NTNU during the fall of 2008, which is a course for graduating master students and PhD students. The results of this project shows that it could be useful with assignment that give some practical insight into process control, judging from survey results from students who completed the exercises. One major benefit the students gets from these assignments, is that they get familiar with abbrevations used in the industry, and the gap between theory and practice will hopefully be a bit smaller for those who accomplish the assignments.

Contents

List of Figures

List of Abbreviations

ARX	Auto-Regressive with eXogenous inputs
CV	Controlled Variable
DOF	Degree Of Freedom
DV	Disturbance Variable
IC	Indicator Controller
ISE	Integrated Squared Error
MPC	Model Predictive Control
MV	Manipulated Variable
PFD	Process Flow Diagram
PID Controller	Proportional Integral Derivative Controller
PV	Process Variable
RGA	Relative Gain Array
RMPCT	Robust Multivariable Predictive Control Technology
SIMC	Skogestad/Simple Internal Model Control
SVD	Singular Value Decomposition

Introduction

In this project, a distillation column, or more specifically, a *debutanizer* is used for the purpose of teaching advanced process control. The debutanizer is implemented in the dynamic simulation tool UniSim, which is developed by Honeywell. The idea is that this model should be used for teaching purposes, and in this project a case study regarding this model is proposed. The case study is divided in two parts, or assignments; the first being tuning of basic control loops (levels) and temperature control, and the second part is concerned with more advanced control (i.e. decentralized PID control and multivariable control).

Some comments regarding the structure of this report: First, the most important theory used in the case study is discussed. Then each of the two assignments are presented, succeeded by their respective solution proposals. After this, some comments on the specific contents of the assignments with respect to pedagogics are given. This includes some comments from students that have actually done the assignments, on which topics they gained most new knowledge in. The final chapter includes conclusions to the project, both a technical conclusion to the assignments and a pedagogic conclusion which concludes the project. The appendices at the end includes some useful information regarding the two assignments, as well as a note on the identification experiment used to find a model for the MPC controller in the second assignment.

It is stressed that the focus in this report is on control theory, and not the tools used for simulation or identification. Nevertheless, some insight into the simulation tool is required, and the most important issues are discussed in the assignment texts.

The assignments texts and tasks are formulated solely by the author, as well as the solutions. Professor Skogestad has been very helpful with the theory, which is greatly appreciated. Bjørn Einar Bjartnes from Honeywell has been very helpful with respect to UniSim and design of the MPC controller, and this help is also greatly appreciated.

CHAPTER 1

THEORY

In this chapter, the control theory applied in the case studies will be discussed. The emphasis is put on topics regarding tuning of PID-controllers, simple model reduction, the maximum gain principle and Honeywells MPC implementation ”RMPCT”.

1.1 Regulatory level control

In this section, control of ”basic” control loops is discussed. The word ”basic” is put in quotation marks because which loops included in the basic control layer are not well defined for a general plant. These loops may include the loops that stabilizes the plant in a mathematical sence, i.e stabilizes the unstable modes of the plant. It may also include the loops that stabilizes the plant in a more practical sense, that is, avoiding drifting, large overshoots and oscillations on the controlled variables (CVs). The discussion here includes reduction of plant models, tuning rules and pairing of inputs and ouputs.

1.1.1 The SIMC tuning rules

Skogestad [SP05] has derived simple rules for model reduction and PID tuning. In process control, it is common to approximate a process with a first order plus time delay model:

$$G(s) = \frac{k}{\tau s + 1} e^{-\theta s} \tag{1.1}$$

Specifying a first order reference tracking response and using a first-order approximation of the time delay gives

$$K_c = \frac{1}{k} \frac{\tau}{\tau_c + \theta} \tag{1.2}$$

$$\tau_I = \tau \tag{1.3}$$

where τ_c is the desired closed loop time constant, and the only tuning parameter. However, for nearly integrating processes with large τ, step disturbances entering the plant input will affect the output in a ramp-like fashion. To counteract this, one may modify (increase) the integral action by decreasing τ_I. To avoid undesired closed loop oscillations, τ_I cannot be decreased too much. Skogestad [SP05] recommends the following general SIMC PI settings for the plant model (1.1):

$$K_c = \frac{1}{k}\frac{\tau}{\tau_c + \theta} \tag{1.4}$$

$$\tau_I = min(\tau, 4(\tau_c + \theta)) \tag{1.5}$$

The use of derivative action (PID control) is uncommon in process applications, where most plants are stable with overdamped responses. One exeption is for a ”dominant” second-order process

$$G(s) = k\frac{e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{1.6}$$

where $\tau_1 \geq \tau_2$ and ”dominant” means that $\tau_2 > \theta$. For the plant model (1.6), the recommended tunings are those in (1.4) and (1.5), in addition to

$$\tau_D = \tau_2 \tag{1.7}$$

To achieve ”fast” control, Skogestad [SP05] recommends

$$\tau_c = \theta \tag{1.8}$$

which for the model (1.6) gives a sensitivity peak $M_s \approx 1.7$, gain margin $GM \approx 3$ and crossover frequency $\omega_c = \frac{0.5}{\theta}$.

Sigurds rule

Skogestad [Sko03] proposes a method for tuning controllers used for integrating processes. Consider a PI-controller with settings K_{c0} and τ_{I0} which results in 'slow' oscillations with period P_0 (larger than 3 times τ_{I0}, approximately). The equation for the PI controller is

$$c(s) = K_c(1 + \frac{1}{\tau_I}s) \tag{1.9}$$

and the process model is, approximated as a pure integration

$$g(s) = \frac{k'}{s} \tag{1.10}$$

The closed loop polynomial $1 + g(s)c(s)$ hence is

$$1 + g(s)c(s) = \frac{\tau_I}{k'K_c}s^2 + \tau_I s + 1 \tag{1.11}$$

The standard second-order form reads

$$\text{denominator} = \tau_0^2 s^2 + 2\tau_0 \zeta s + 1 \tag{1.12}$$

Comparing (1.11) and (1.12) gives

$$\begin{aligned} \tau_0 &= \sqrt{\frac{\tau_I}{k'K_c}} \\ \zeta &= \frac{1}{2}\sqrt{k'K_c\tau_I} \end{aligned} \tag{1.13}$$

Oscillations occur for $\zeta < 1$. By using standard analysis of second-order systems (e.g [DS89] p. 112), the period of oscillations may be expressed as

$$P_0 = \frac{2\pi}{\sqrt{1-\zeta^2}}\tau_0 = \frac{2\pi}{\sqrt{1-\zeta^2}}\sqrt{\frac{\tau_{I0}}{k'K_{c0}}} \approx 2\pi\sqrt{\frac{\tau_{I0}}{k'K_{c0}}} \tag{1.14}$$

where it is assumed that $\zeta^2 << 1$ (significant oscillations). Rearranging (1.14) gives:

$$K_{c0}\tau_{I0} = (2\pi)^2\frac{1}{k'}\left(\frac{\tau_{I0}}{P_0}\right)^2 \tag{1.15}$$

In order to get $\zeta \geq 1$, we must from (1.13) require that

$$K_c\tau_I \geq \frac{4}{k'} \tag{1.16}$$

Dividing inequality (1.16) by equation 1.15, we get the requirement

$$\frac{K_c\tau_I}{K_{c0}\tau_{I0}} \geq \frac{1}{\pi^2}\left(\frac{P_0}{\tau_{I0}}\right)^2 \tag{1.17}$$

By approximating $\frac{1}{\pi^2} \approx 0.1$, the definition of Sigurds rule is

Definition 1. *To avoid slow oscillations of period P_0, the product of the controller gain and integral time should be increased by a factor $f \approx 0.1\left(\frac{P_0}{\tau_{I0}}\right)^2$*

1.1.2 A systematic approach to reduce plant models

This section is extracted from [Sko03]. In order to use simple tuning rules (the SIMC tuning rules), a simplified plant model is needed. A general transfer function from input to output is noted:

$$G(s) = k\frac{(\tau_{z,k}s+1)(\tau_{z,k-1}s+1)\dots(\tau_{z,0}s+1)e^{-\theta s}}{(\tau_{p,n}s+1)(\tau_{p,n-1}s+1)\dots(\tau_{p,0}s+1)} \tag{1.18}$$

where z and p indicates that the time constants are associated with a zero or a pole respectively, and k and n is the number of zeros and poles respectively. The terms are sorted so that the largest values of the indices k and n equal the largest time constants, and the time constants are descending in magnitude with lower index values.

Skogestad [SP05] recommends using the *half rule* when approximating a high-order transfer function as a first-order plus delay transfer function for PI-control, or as a second order plus delay transfer function for PID control.

Definition 2. *The largest neglected lag (denominator) time constant should be distributed equally to the effective delay and the smallest retained time constant. This is defined as the **half rule**.*

Approximation of right half plane zeros: Right half plane zeros of a transfer function leads to *inverse responses* in the time domain. An approach to approximate right half plane zeros is to use the approximation $e^{-\theta s} \approx 1 - \theta s$. By performing this approximation, the right half plane zeros are treated as effective time delays.

Approximation of left half plane zeros: By constructing transfer functions with one zero and one pole paired by the indices k and n, we may express (1.18) as

$$G(s) = \tilde{G}(s) \cdot \prod_{i=1}^{j} G_i(s) \tag{1.19}$$

where G(s) has j left half plane zeros, and

$$G_i(s) = \frac{\tau_{k-i,z} s + 1}{\tau_{n-i,p} s + 1} \tag{1.20}$$

and $\tilde{G}(s)$ is a transfer function containing right half plane zeros, pure time delay and $n - j$ denominator terms. Skogestad proposes the following approximations for the functions $G_i(s)$, $\forall i \in [1, j]$:

$$G_i(s) = \frac{\tau_{k-i,z} s + 1}{\tau_{n-i,p} s + 1} \approx \begin{cases} \frac{\tau_{k-i,z}}{\tau_{n-i,p}} & \text{for } \tau_{k-i,z} \geq \tau_{n-i,p} \geq \theta \\ \frac{\tau_{k-i,z}}{\theta} & \text{for } \tau_{k-i,z} \geq \theta \geq \tau_{n-i,p} \\ 1 & \text{for } \theta \geq \tau_{k-i,z} \geq \tau_{n-i,p} \\ \frac{\tau_{k-i,z}}{\tau_{n-i,p}} & \text{for } \tau_{n-i,p} \geq \tau_{k-i,z} \geq 5\theta \\ \frac{\tilde{\tau}_{n-i,p}/\tau_{n-i,p}}{(\tilde{\tau}_{n-i,p} - \tau_{k-i,z})s+1} & \text{for } \tilde{\tau}_{n-i,p} \triangleq min(\tau_{n-i,p}, 5\theta) \geq \tau_{k-i,z} \end{cases} \tag{1.21}$$

where θ is the time delay. Then, after all right and left half plane zeros are approximated, we have a transfer function on the form:

$$G_{\text{approx}}(s) = \frac{k \cdot e^{-\theta s}}{\prod_{i=1}^{m} (\tau_i s + 1)} \tag{1.22}$$

with m poles. This transfer function may in turn be reduced by applying the half rule, and adding the smaller lag terms to the effective time delay.

1.1.3 The maximum gain rule

The idea behind the max gain rule, is that an approximation of the solution to the optimization problem of minimizing a *loss function* $J(u, d)$ (u is manipulated inputs, d is disturbances)is derived through local analysis. When we have large gain from our input variable u to our controlled variable c, the effect of implementation error is 'minimized'. In order to define the local loss, we consider a Taylor series expansion of the loss function around the optimal point $(u_{\text{opt}}(d), d)$, when u differs from $u_{\text{opt}}(d)$

$$J(u, d) = J_{\text{opt}}(d) + \underbrace{\left(\frac{\partial J}{\partial u}\right)_{\text{opt}}^{\top}}_{=0} (u - u_{\text{opt}}(d)) + \frac{1}{2}(u - u_{\text{opt}}(d))^{\top} \left(\frac{\partial^2 J}{\partial u^2}\right)_{\text{opt}} (u - u_{\text{opt}}(d)) + \ldots \tag{1.23}$$

By ignoring higher order terms, we get

$$\text{LOSS} = J(u, d) - J_{\text{opt}}(d) \approx \frac{1}{2}(u - u_{\text{opt}}(d))^{\top} J_{uu}(u - u) \tag{1.24}$$

Now let G be the unscaled steady-state gain matrix of the system at hand:

$$(c - c_{\text{opt}}(d)) = G(u - u_{\text{opt}}(d)) \tag{1.25}$$

The loss function may now be expressed as

$$\begin{aligned} LOSS &= \frac{1}{2}(c - c_{\mathrm{opt}}(d))^{\top} G^{-\top} J_{uu} G^{-1}(c - c_{\mathrm{opt}}(d)) \\ &= \frac{1}{2}\left|\left|J_{uu}^{\frac{1}{2}} G^{-1}(c - c_{\mathrm{opt}}(d))\right|\right|_2^2 \\ &\leq \frac{1}{2\big(\underline{\sigma}(S_1 G J_{uu}^{-\frac{1}{2}})\big)^2} \end{aligned} \tag{1.26}$$

We are now ready to define the max gain rule:

Definition 3. *To minimize loss, we want to maximize* $\underline{\sigma}(S_1 G J_{uu}^{-\frac{1}{2}})$ *of equation* (1.26)*, where*

$$G_s = S_1 G J_{uu}^{-\frac{1}{2}} \tag{1.27}$$

$$S_1 = diag\left\{\frac{1}{span(c_i)}\right\} \tag{1.28}$$

$$span(c_i) = n_i^c + \Delta c_{i,opt}(d) \tag{1.29}$$

$$n_i^c = implementation\ error \tag{1.30}$$

The controlled variables (CVs) that maximize $\underline{\sigma}(G_s)$ *should be selected. This is defined as the* ***maximum gain rule****.*

In words, the maximum gain rule states that controlled variables for which the gain G (controllable range) is large compared to its span (sum of optimal variation and control error) should be selected.

1.2 Supervisory Control

Skogestad and Postlethwaithe [SP05] defines the supervisory control in a hierarchic manner, being on top of the regulatory control layer. In terms of control, this involves more advanced control structures, like MPC, decouplers, decentralized PID etc. According to Skogestad and Postlethwaithe [SP05], the hierarchic way of decompose the control system involves that controllers are designed sequentially. The fast layers are designed first, and then cascaded in a hierarchical manner with the above layers. For supervisory control in this project, Honeywells RMPCT will be utilized. An introduction to RMPCT will be given here, from [Mac02]. In addition, the decentralized PID approach will be used for 'comparison', but no in-depth discussion will be given about this approach here.

1.2.1 Introduction to RMPCT

The name RMPCT is an acronym for Robust Multivariable Predictive Control Technology, a product which has been developed by Honeywell. A distinctive feature of RMPCT is that it implements the 'zone' or 'funnel' concept, in which the objective of control is to keep plant outputs within defined regions, rather than at specific setpoints. Each controlled output has an allowed range or zone, defined by a lower limit and a higher limit. As long as its value is within this range, it is not penalized in any way. If an output is outside the range, a straight-line trajectory is constructed from its current value to the nearest edge of the allowed range; the output variable is supposed to follow this trajectory to the target zone. The time taken by this trajectory to hit the allowed range is pre-determined for each control output, and it varies between 0 and twice the open-loop settling time of that output. Whenever the output is predicted to lie outside the 'funnel' formed by the target zone and this trajectory, its distance from the edge of the funnel is defined to be an error. A quadratic cost function is formed as a weighted sum-of-squares of these errors. Future inputs are chosen so as to minimize this cost function, subject to constraints on input levels and moves. The optimization problem to be solved is a QP-problem, and an active set solution strategy is used. It can be seen that the 'agressiveness' with which an output is returned to its target zone is determined by the slope of the edge of the funnel which is constructed to bring it back into the allowed range. Thus this edge serves as a straight-line reference trajectory, combining this concept with the funnel concept. See figure 1.1 for an example on how the funnel concept works when there are range limits on the controlled variables (not set-point tracking). Conventional setpoint control of a variable is obtained by making the low and high limits the same for that variable, see figure 1.2. See figure 1.3 for an example where we have a change in set-point.

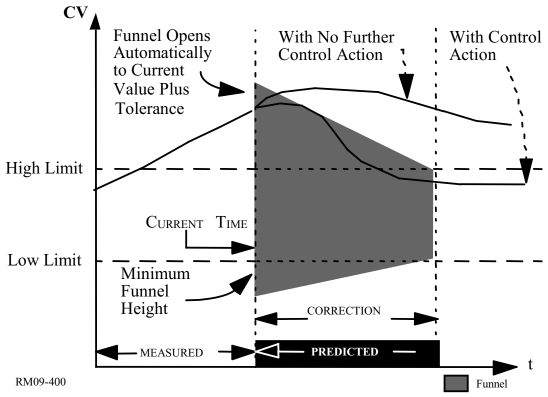


Figure 1.1: Funnel with range limits on the controlled variables

Feedforward to combat the effects of measured disturbances is handled in a similar way. The effect of such disturbances on the controlled outputs are predicted, and any output driven outside its allowed range is again driven back by reference to a funnel. Typically, the funnels for feedforward slope more steeply than those for recovery from unmeasured disturbances - namely those for correction by feedback action - because feedforward action, being outside any feedback loop, can be more aggresive without risk of being unstable. The RMPCT controller is organized in two levels, one for optimization and one for control. A steady-state model is used to find the optimal steady-state for the process, on the assumption that the 'current-steady-state' is the currently predicted steady state. The cost function used for this purpose is a combination of the traditional linear cost function which captures the economic benefits of producing particular quantities of various products and the economic cost of using control inputs, with a quadratic function

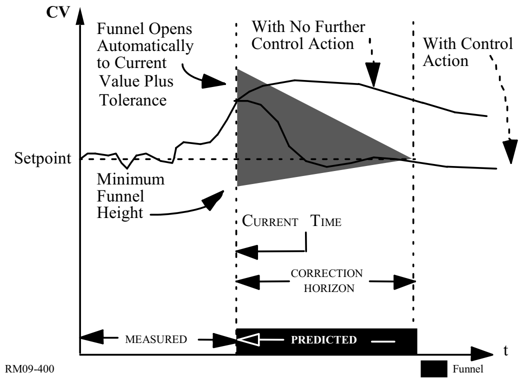


Figure 1.2: Funnel with constant set-point on the controlled variables

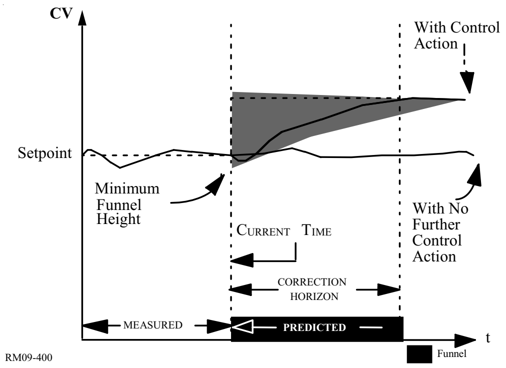


Figure 1.3: Funnel with change in set-point on the controlled variables

which represents the costs of moving away from the current steady-state inputs and outputs. The general form is, from [Hon07b],

$$\text{minimize}\, J = \sum_i b_i \cdot CV_i + \sum_i a_i^2 (CV_i - CV_{0,i})^2 + \sum_j b_j (MV_j) + \sum_j a_j^2 (MV_j - MV_{0,j})^2 \tag{1.31}$$

where

- b_i are the linear coefficients on the CVs
- b_j are the linear coefficients on the MVs
- a_i are the quadratic coefficients on the CVs
- a_j are the quadratic coefficients on the MVs
- $CV_{0,i}$ are the desired resting values of the CVs
- $MV_{0,j}$ are the desired resting values of the MVs

To maximize rather than minimize the objective function, the sign of the linear coefficients must be set negative. The quadratic coefficients will be squared in the objective function, so the terms with these coefficients will always be positive.

Process variables can appear in this cost function which do not appear in the cost function used for dynamic control, and vice versa. The steady-state optimization can be performed for several process units simultaneously, and the results passed down to the individual dynamic controllers. The interaction between the steady-state and dynamic optimization is not the conventional one, of simply passing steady-state objectives to the dynamic controller. Rather, the new steady-state values are included in the dynamic optimizations, but with a variable weighting which allows the 'strength' with which the process is pushed to the new steady-state to be varied. RMPCT allows 'blocking' of the predicted control moves, so that the moves computed by the optimization are not assumed to be spaced at equal intervals in the control horizon. The dynamic model used by RMPCT is a multivariable ARX model. RMPCT attempts to increase the robustness of its controller to modelling errors by using a number of strategies. The first of these is to introduce scaling of the input and output variables, in order to optimize the conditioning of the optimization problem. The idea is to reduce the sensitivity of the output trajectories to model errors. A second strategy is to approximate the prediction model by one with better conditioning, if the original model is ill-conditioned with respect to inversion. This is done by identifying the active constraint set involved in the optimization, performing a singular value decomposition (SVD) of the corresponding matrix, and approximating it by discarding any very small singular values. The third strategy is adjusting the internal model used by the controller. The controller corresponding to a particular internal model is assessed by using simulation to evaluate the 'ISE' (Integrated Squared Error) criterion for a range of plant model errors. The internal model is then adjusted in a kind of extremum-seeking process until one is found for which the worst-case ISE criterion is the smallest.

CHAPTER 2

DEBUTANIZER PART 1

2.1 Background

This assignment is intended to show some of the practical configuration problems related to plantwide control. Different control structures for the main composition (PID, and MPC later) may be used for comparison. The plant in this assignment is a distillation column, or more specifically a *debutanizer* (see figure 2.1). The intention is to separate out the butanes (n-butane, i-butane, i-butene) of the input feed streams, which in this case are disturbances. The feed streams are compositions of:

- i-Butane
- n-Butane
- i-Butene
- i-Pentane
- n-Pentane
- n-Hexane
- n-Heptane
- n-Octane

The split is basically between pentanes (heavy key components) and butanes (light key components).

2.1.1 The default control configuration

The current control setup of the debutanizer is a standard *LV-configuration*. This basically means that the variables used for composition control are (figure 2.1):

- The **reflux** L, which is controlled by FIC-100 (flow controller)
- The **boilup** V, which is controlled by TIC-100 (temperature controller)

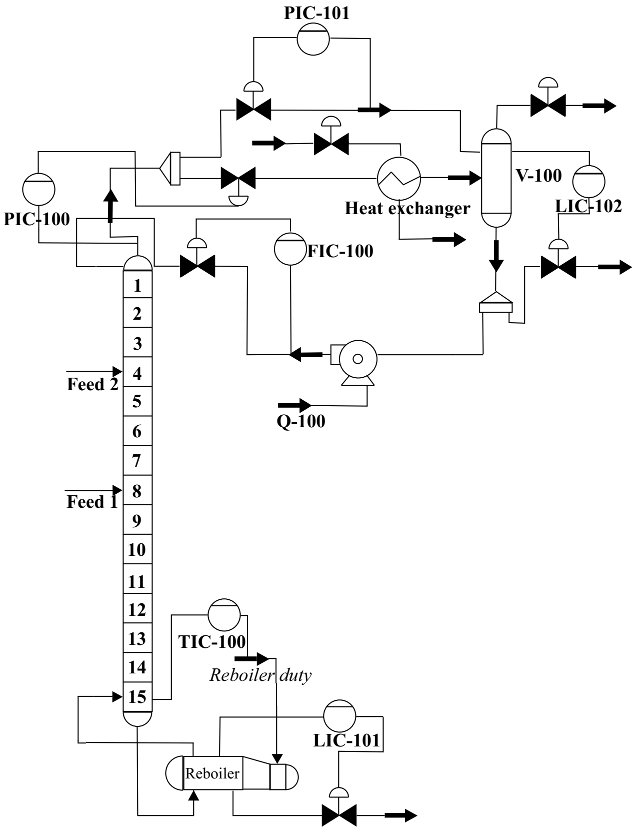


Figure 2.1: Process Flow Diagram for the debutanizer

2.1.2 About the simulation environment

UniSim provides two different types of simulation environments, namely *steady state* simulation and *dynamic* simulation. The debutanizer runs in *dynamic* mode, which means that the system is integrated at discrete time steps while the simulation is running. By pressing *CTRL+I* the user may start and stop the integrator, select step size, real time factor, acceleration and so on. The simulation tends to run smoother if the integrator is run at a user provided real time factor rather than full speed.

Process Flow Diagram

The model itself is visualized in the *Process Flow Diagram* (PFD). If you cannot find the PFD, press *CTRL+P*. UniSim will find the PFDs associated with the case, press *view* to open the desired PFD. If you take a look at the PFD, you will notice numerous blue arrows. These are *material streams*, which means they are gas and/or liquid streams. The streams have several features related to them, the most important being:

- Pressure
- Temperature
- Flowrate

By double-clicking one of the streams, the property window of the current stream will appear. All the information regarding the stream is gathered here, also dynamically updated while the case is running. By choosing *Composition* under *Worksheet* for the stream at hand, you will see the mole fraction of each element in the stream.

Controllers

All of the PID-controllers have *IC* in their acronyms, for Indicator Controller. *FIC*, *TIC*, *PIC* and *LIC* are respectively flow, temperature, pressure and level indicator controllers. When double clicking one of the controller icons, the property view for the controller will show up. In the bottom midst of the view you will find the *face plate* button. The face plate is a small display showing the controller setpoint, output and the controlled variable. It is recommended to have a face plate up for all of the controllers when the case is running, to keep track of your controlled variables.

To tune a controller, double click the controller to view the controller property view. Click the *parameters* tab, and the tuning face appears. At ”Current Tuning”, choose desired values for K_c, T_i and T_d. If you already have a face plate for the associated controller, press the ”tuning” button.

Strip charts

Strip charts provides a graphical view of variables specified by the user. A good idea is to put variables with the same scaling in the same strip chart. In order to make your own strip chart, select *Tools → Databook* from the menu bar. Press the ”Add” button to create a new strip chart, and name it. A number of variables is already specified in the databook, but you can add others if you need to. Select the ”Variables” tab, then ”Insert” to add a new variable. At ”flowsheet”, select ”T-100” (our PFD). A list of all the objects in the PFD will appear. Select the object the variable is related to, and then the variable itself. This may be a stream, a separator, a valve or any object in the PFD. An example is V-100 → Liquid Volume Percent. This will give you the variable ”Volume of separator V-100 in percent”.

When all interesting variables are added, go back to the strip chart tab and select the variables you want in your strip chart by checking the boxes next to the variables. By selecting ”Strip Chart” under ”View”, the current strip chart is visualized.

Exporting data to Excel/Matlab (optional, not necessary)

When you have made your strip chart, select ”Historical” under ”View” in the Databook property window. This features the option to save data to a CSV-file. The data will be saved as a row vector containing all selected variables for each time step. The variables are separated by commas (,-s). This data-file may now be imported in Excel (make sure you select the comma-separation when using the data import function), or parsed with a Matlab-script.

The simulation environment will not be discussed further here, since modelling is not a part of the assignment. The necessities for the assignment itself will be explained throughout the assignment text.

2.2 Assignment

The assignment itself consists of three parts, respectively:

1. Tuning of the basic controllers using SIMC and using Sigurds rule to verify the tunings

2. Introduce temperature control using reboiler duty

3. Apply the max gain rule on the temperature control loop

Throughout the assignment you will also keep track of the variations in the following variables:

- $x_{n-butane}$ - The mole fraction of n-butane in the bottom product flow

- $x_{i-pentane}$ - The mole fraction of i-pentane in the top product flow

- $D[\frac{kg}{h}]$ - The mass flow of the top product (D for Distillate)

- $B[\frac{kg}{h}]$ - The mass flow of the bottom product (B for Bottoms)

These variables will from now on be refered to as our **main variables**.

There will be strip charts for these variables available in the case. Select Tools → Databook (or *CTRL+D*), and select Strip Chart for ”Top and bottom compositions” and ”Top and bottom flowrates”. Appendix A.2 contains a table that you can fill inn with the different variations.

The **overall objective** of this assignment is to keep;

$$x_{n-butane} < 2\% \tag{2.1}$$

$$x_{i-pentane} < 2\% \tag{2.2}$$

with the given, constant feeds.

*In task **1**, the temperature control loop gain is set to a very low value, which means we have no temperature control. Temperature control will be introduced in task **2**.*

Task 1

Run the column by either pressing the green light in the menu bar, or open the integrator by pressing *CTRL+I* and start the integrator. Make sure you have a face plate for both level controllers up.
a) Note the current level controller tunings in table A.3. Do this for both LIC-100 and LIC-102. Also note the current setpoint for TIC-100.
b) *Double-click the stream Feed 2. In the lower right corner there should be two arrow-icons, "View upstream operations" and "View downstream operations". Click the arrow that says "View upstream operations". A new property view will show up, with some of the parameters highlighted in blue. Blue means we can change these values.*

Perform a step on the flowrate of Feed 2 from 6.865$\frac{m^3}{h}$ to 8.865$\frac{m^3}{h}$ (this is a step in disturbance). Keep an eye on the controller face plates, and watch the process variables. What/which controller(s) need to be tuned? Look for oscillating process variables, and estimate the period and amplitudes of the flow and level oscillations using the strip charts "Top and bottom flowrates", and "Levels". Rightclick and use "autoscale curve" or "autoscale all axes" for better view. Note the amplitudes of the flowrate oscillations in table A.1 , and the periods and amplitudes of the level oscillations in table A.2. What happens with the flowrate of butanes?
c) Put the controller(s) that need to be tuned in manual, and perform a step test. Tune the controller(s) using the SIMC tuning rules, and use Sigurds rule[1] to verify the tunings. See appendix A.1 for a guide on how to easily tune level controllers, and see appendix A.3 on how controller gain scaling in UniSim works.[2] Note your tuning parameters in table A.4.
d) Note the steady state values of our main variables, and step flowrate of Feed 2 back to 6.865$\frac{m^3}{h}$. Note the values of our main variables when the system is at steady state, and fill in the variations of our main variables in table A.1. Verify that the flows out of the column do not oscillate.

[1] Sigurds rule: To avoid slow oscillations, the product of the controller gain and integral time should be increased by a factor $f = 0.1(\frac{P_0}{\tau_{I0}})^2$, where P_0 is the period of oscillations, and τ_{I0} is the original integral time

[2] Hint: We do not need tight level control (ouput y), and certainly not any oscillating levels. The main objective is to have "smooth" control where the flowrates (input u) does not change unnecessary. For this assignment we assume that the flows out of the column are inputs to another operation, hence we want constant flows out of the column. Choose a closed loop time constant of about 30 minutes for LIC-102, and 10 minutes for LIC-100

Task 2

We will now utilize the temperature controller that controls one of the stage temperatures using reboiler duty. In order to tune this loop, we need to identify the process gain and the process time constant. This can be done by waiting for steady state to occur, and check the timepoint where the output has reached 63% *of its final value (use the "historical" option in the databook). This way you may measure the time constant directly by subtracting the time instant when the step occured from the time instant where the output has reached 63%. Use the strip chart called "Stage temperatures" for this purpose. The highest temperature is our current process variable*

a) Put the temperature controller in manual mode, and perform a step test. Tune the temperature controller using the SIMC tuning rules (1st order approximation). Choose a closed loop time constant of $5min$. Note your tunings (K_c and τ_i) in table A.4.
b) Find a set point of TIC-100 that fullfills the overall objective of our assignment. This means flowrate of Feed 1 is $13.09\frac{m^3}{h}$ and flowrate of Feed 2 is $6.865\frac{m^3}{h}$ (they should have this values by default at this point). Note that setpoint in table A.4.
c) Note the values of our main variables, and perform a step of Feed 2 like in **a**. Note the values of our main variables after the system is at steady state, and fill in table A.1.

Task 3

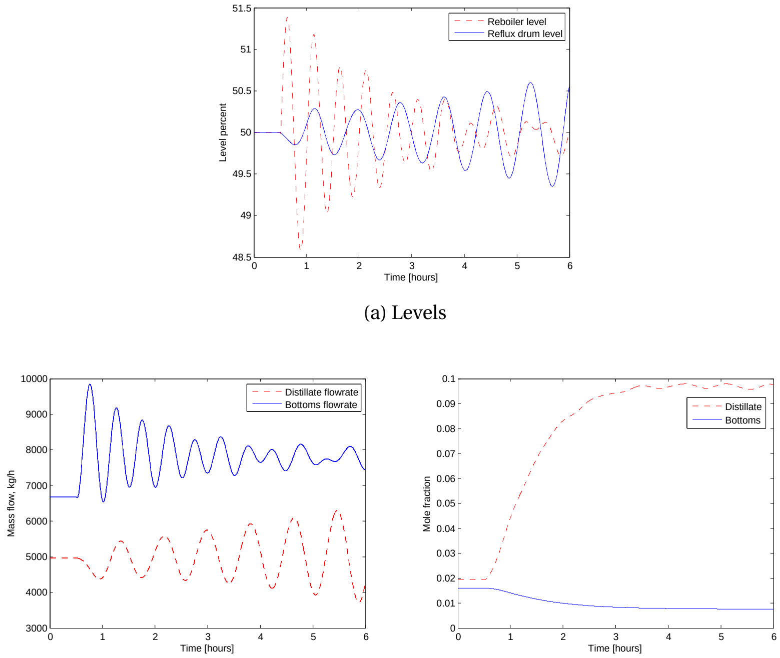
We will now look into the temperature measurement in the column for temperature control. The measurement is by default placed at the bottom of the column (see figure 2.1). According to the max gain rule, the temperature in the column that changes the most to a setpoint change in TIC-100 should be our Process Variable (PV).
a) Select Tools → Databook, and open the strip chart called "Stage temperatures". Fill inn the current values in table A.5 Perform a setpoint change of TIC-100, and study the temperature responses of each column stage. Fill in the steady state temperatures after the step in table A.5. How is the temperature change in our current stage (stage 15) compared to the other stages?
b) Find the stage with approximately most gain and move the temperature measurement to this stage. The temperature measurement may be moved by double clicing controller TIC-100, select the "Connections" tab, "Select PV", and select one of the other stage temperatures as PV. If the compositions aren't both under 2%, find a setpoint that fullfills this goal.
c) Note the values of our main variables, and perform a step of Feed 2 like in **a**. Note the values of our main variables after the system is at steady state, and fill in table A.1.
d) If we want to implement a MPC controller in our plant, what would the degrees of freedom be? What should be the constraints?

CHAPTER 3

Proposed solution for Debutanizer Part 1

Task 1

a) See table 3.3 for the initial tuning parameters of LIC-100 and LIC-102.
b) The two level controllers obviously needs to be tuned. Figure 3.1a shows the response to the disturbance step, and figure 3.1b shows the flowrates of distillate and bottoms as a consequence of the poor level controller tunings. The periods of the oscillations are about 49 minutes for level of the reflux drum, and 30 minutes for the reboiler level (values found in MATLAB). The butane flow (distillate flow) oscillates with increasing amplitude, and will eventually oscillate between an upper limit given by the maximum pressure and zero (control valve closed). Figure 3.1c shows the compositions response to the disturbance step.
c) A step test is performed to identify the parameters of the reflux drum and reboiler level dynamics. In both cases a step down of 2% in magnitude is used directly on the controller valves when the two level controllers are in manual mode. The valves are stepped separatly. Figure 3.2 shows the responses. Following the steps in the assignment, we identify the parameter $k' = \frac{\Delta y}{\Delta u \cdot \Delta t}$ for both levels.

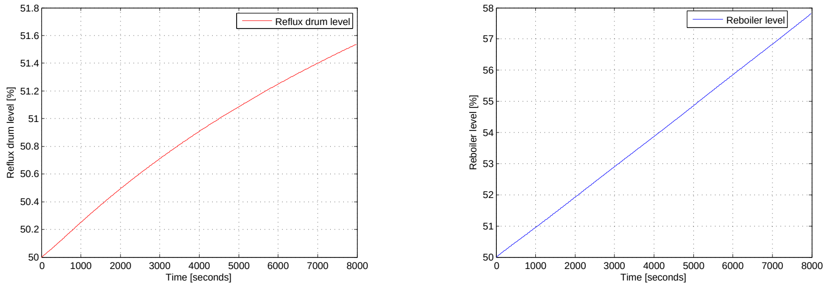


(a) Levels

(b) Flowrates

(c) Compositions

Figure 3.1: Level, flowrate and composition responses to disturbance step with poorly tuned level controllers



(a) Reflux drum

(b) Reboiler

Figure 3.2: Level responses to step down in controller valves by 2%

Reboiler: For the reboiler level our wanted closed loop time constant is 10 min. Calculating the parameter k' by using figure 3.2a, choosing data from time 0 to 4000 seconds:

$$k' = \frac{\Delta y}{\Delta u \cdot \Delta t} = \frac{53.8\% - 50\%}{-2\% \cdot \frac{4000}{60} min} = -0.0285 \tag{3.1}$$

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \underbrace{\theta}_{=0}} = \frac{1}{0.0285} \frac{1}{10} = -3.509 \tag{3.2}$$

$$\tau_I = \min(\tau_1, 4\tau_c) = 4\tau_c = 40\,\text{min (integrating process)} \tag{3.3}$$

Since the range of the CV (level) equal the range of the MV (valve position), the SIMC controller gain equal the "Unisim controller gain". It is also assumed that there is no deadtime present. The system gain is negative, hence the controller should be set to "Direct", and inserting the absolute value of K_c.

The tunings should be verified by using Sigurds rule. Sigurds rule states that if the process variable oscillates, the product of the initial controller gain and the initial controller integral time should be increased by a factor

$$f = 0.1\left(\frac{P_0}{\tau_{I0}}\right)^2 \tag{3.4}$$

where P_0 is the period of the oscillations, and τ_{I0} is the original integral time. Calculation of this factor for the reboiler level gives

$$f = 0.1\left(\frac{30}{0.1}\right)^2 = 9000 \tag{3.5}$$

when we use the fact that the period of the oscillations is about 30 min (found in MATLAB). We know that our new $\tau_I = 40$, since we have already decided on the closed loop response time. This way we may calculate the new controller gain:

$$K_{\text{c,new}} \cdot \tau_{\text{I,new}} = f \cdot K_{\text{c,old}} \cdot \tau_{\text{I,old}} \tag{3.6}$$

$$\Rightarrow K_{\text{c,new}} = f \cdot \frac{K_{\text{c,old}} \cdot \tau_{\text{I,old}}}{\tau_{\text{I,new}}} = 9000 \frac{0.2 \cdot 0.1}{40} = 4.5 \tag{3.7}$$

The error between the K_c found by Sigurds rule and the SIMC tuning rules is about $\frac{4.500-3.509}{3.509} \cdot 100\% = 28.2\%$, which may be caused by slightly wrong estimated period of oscillation due to nonlinearities, and that there are (weak) interactions between the two levels. Since the factor f is sensitive to the value of P_0 (quadratic, see equation (3.4)), wrongly estimated period of oscillations will have quite an impact on the resulting value of K_c. In addition, a difference between the two methods could occur when the parameter k' is estimated during the SIMC tuning procedure. In the derivation of Sigurds rule (see section 1.1.1), it is assumed that the oscillations are persistent, which is not true in this case (see figure 3.1a).

The two methods for tuning are in a mathematical sense the same, but there is a practical difference since the SIMC tunings are based on estimation of the process gain k', and the tuning using Sigurds rule is based on estimation of the oscillation period P_0.

Reflux drum: For the reflux drum level our wanted closed loop time constant is 30 min. Calculating the parameter k' by using figure 3.2b, choosing data from time

0 to 6000 seconds:

$$k' = \frac{\Delta y}{\Delta u \cdot \Delta t} = \frac{51.25\% - 50\%}{-2\% \cdot \frac{6000}{60} min} = -0.00625 \tag{3.8}$$

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \underbrace{\theta}_{=0}} = \frac{1}{-0.00625} \frac{1}{30} = -5.33 \tag{3.9}$$

$$\tau_I = min(\tau_1, 4\tau_c) = 4\tau_c = 120\,\text{min (integrating process)} \tag{3.10}$$

The same assumptions made for the reboiler yields for the reflux drum. The period of oscillation of the reflux drum level is about 49 min, and the tunings are verified using Sigurds rule:

$$f = 0.1\left(\frac{49}{0.1}\right)^2 = 24010 \tag{3.11}$$

$$K_{c,new} \cdot \tau_{\text{I,new}} = f \cdot K_{\text{c,old}} \cdot \tau_{\text{I,old}} \tag{3.12}$$

$$\Rightarrow K_{\text{c,new}} = f \cdot \frac{K_{\text{c,old}} \cdot \tau_{\text{I,old}}}{\tau_{\text{I,new}}} = 24010 \frac{0.2 \cdot 0.1}{120} = 4.00 \tag{3.13}$$

In this case the error between the SIMC tunings and Sigurds rule is about $\frac{5.33-4.00}{5.33} \cdot 100\% \approx 25\%$, which may be explained in the same way as for the reboiler level tuning.

d) See table 3.1 for values on the variations of the main variables.

The composition control is obviously bad, since the mole fraction of i-pentane in the distillate is almost 0.1 after the disturbance, and the objective is to keep it under 0.02. Figure 3.3 shows the level and flowrate responses to the same disturbance step as in task **1b)** after the level loops are tuned, and in addition the compositions. It is seen that the flowrates out of the column are no longer oscillating, and neither are the levels. The composition of i-pentane of the distillate was oscillating a bit before the level controllers were tuned, but these oscillations are no longer present, as seen by comparing figures 3.1c and 3.3c.

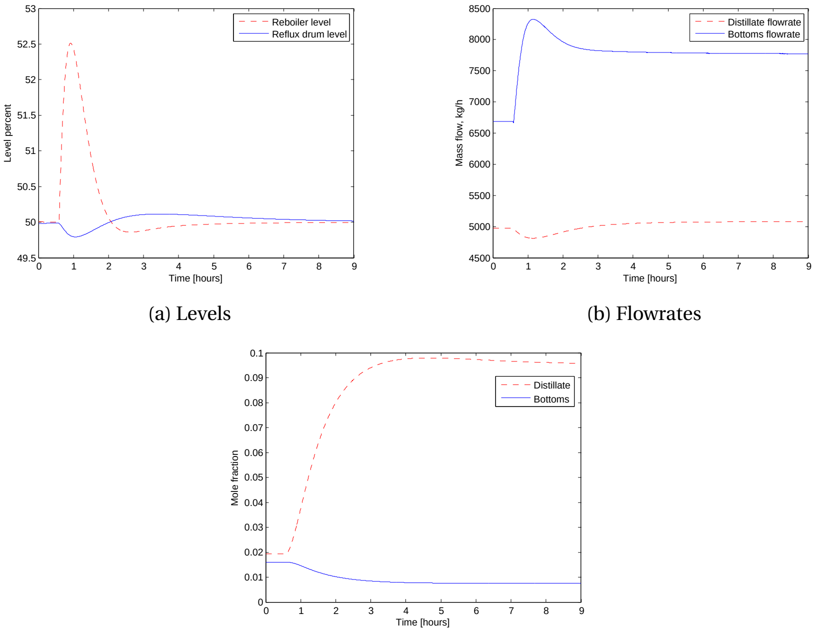
Task 2

a) In order to catch the dynamics of the stage temperature, a step test is performed. A step of 2% in magnitude of the reboiler is used, and the result is showed in figure 3.4. The timepoint where the temperature has reached 63% of its final value is outlined in figure 3.4. This gives us the timeconstant directly, it is 1260 s = 21 min. The steady state temperature is 151.9°C, and the initial temperature was 149.2°C. From this we may approximate a first order transfer function from reboiler duty to bottom stage temperature (no deadtime was detected):

$$G(s) = \frac{k}{\tau_1 s + 1} e^{0s} = \frac{k}{\tau_1 s + 1} \tag{3.14}$$

$$k = \frac{\Delta y}{\Delta u} = \frac{(151.9 - 149.2)^\circ C}{2\%} = 1.35 \left[\frac{^\circ C}{\%}\right] \tag{3.15}$$

$$\tau_1 = 21\,\text{min} \tag{3.16}$$



(a) Levels

(b) Flowrates

(c) Compositions

Figure 3.3: Levels, flowrate and composition responses to disturbance step with tuned level controllers

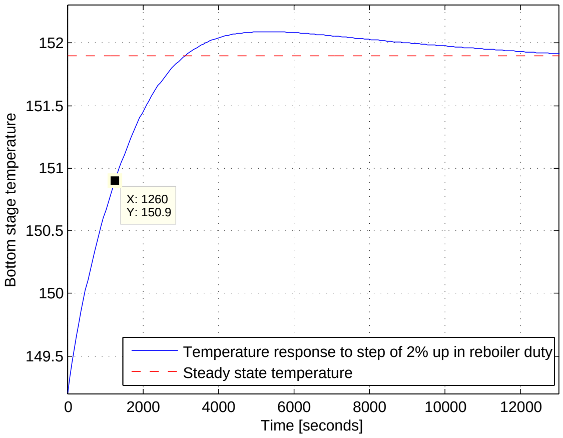


Figure 3.4: Step response of the bottom stage temperature, step of 2% in reboiler duty

Now it is possible to tune the temperature controller using the SIMC tuning rules, with a desired closed loop time constant of 5 min.

$$K_c = \frac{1}{k'}\frac{1}{\tau_c + \theta} = \frac{\tau_1}{k}\frac{1}{\tau_c} = \frac{21}{1.35}\frac{1}{5} = 3.11 \tag{3.17}$$

$$\tau_I = min(\tau_1, 4\tau_c) = min(21\,\text{min}, 20\,\text{min}) = 20\,\text{min} \tag{3.18}$$

In this case, we find the ranges of the process variable (bottom stage temperature) and the manipulated variable in the property view of the temperature controller to be

$$\text{range}(y) = 160\,^\circ\text{C} - 80\,^\circ\text{C} = 80\,^\circ\text{C} \tag{3.19}$$

$$\text{range}(u) = 100\,\% - 0\,\% = 100\,\% \tag{3.20}$$

By following the controller gain scaling procedure in appendix A.3, we calculate the scaled K_c, that is the K_c we actually use in Unisim:

$$K_{c,\text{scaled}} = K_c \frac{\text{range}(y)}{\text{range}(u)} = 3.11 \cdot \frac{80}{100} = 2.49 \tag{3.21}$$

b) We are looking for a good setpoint for the temperature controller, so that both compositions are under 0.02. Figure 3.5 shows a plot of the change of temperature setpoint and the respective composition responses. A setpoint of 149.5°C seems to work ok. This gives $x_{n-butane} = 0.0177$ and $x_{i-pentane} = 0.0173$, hence both are under 2%. The response of the tuned temperature controller to the setpoint change seems quite good, with a small overshoot and pretty fast dynamics.

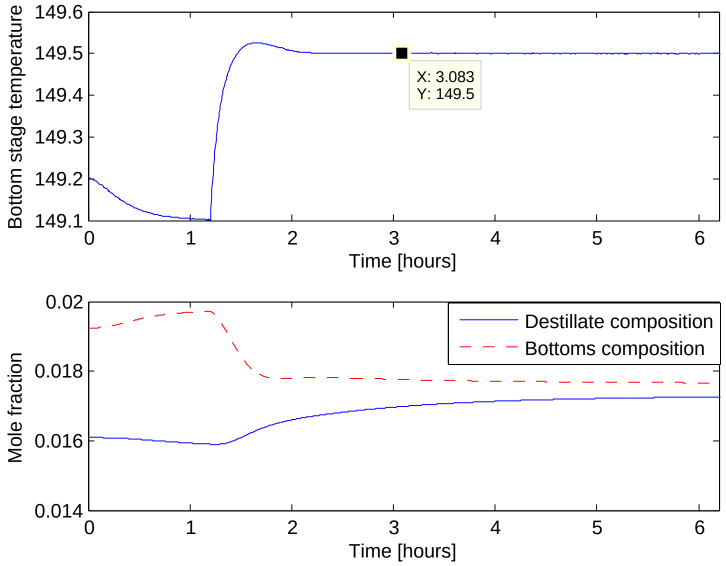


Figure 3.5: Temperature setpoint that gives good compositions

c)

See table 3.1 for the variations. Figure 3.6 shows the composition response and the temperature response to the same disturbance as in task **1b)**.

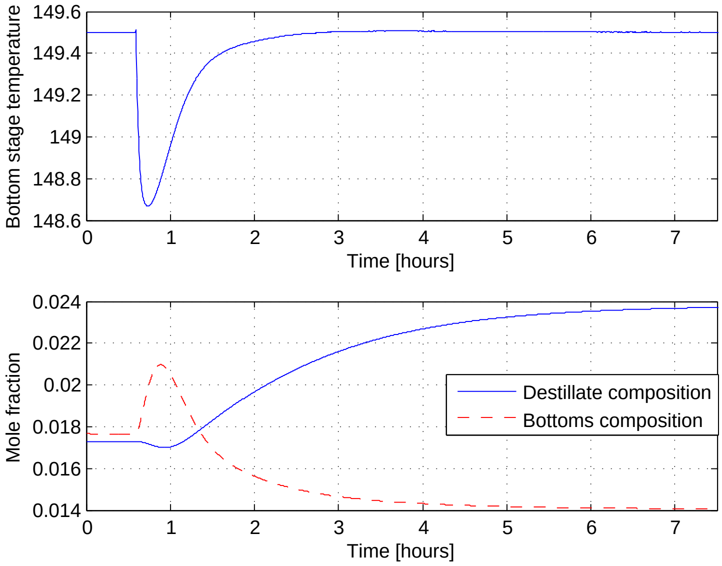


Figure 3.6: Variations of the compositions to disturbance step, and temperature response with tuned temperature loop

Task 3

a) In order to find the stage with the most gain from reboiler duty to temperature, a step of 1 °C on the setpoint of the temperature controller is performed, and all the temperatures are logged and imported into MATLAB. Figure 3.7 shows the responses of all stages to the setpoint change. By subtracting the steady state temperatures before the step from the steady state values after the step, we get an indication on which stage has the most gain. This is done in MATLAB, and the vector of temperature differences is, with the index of the vector equal to the stage number

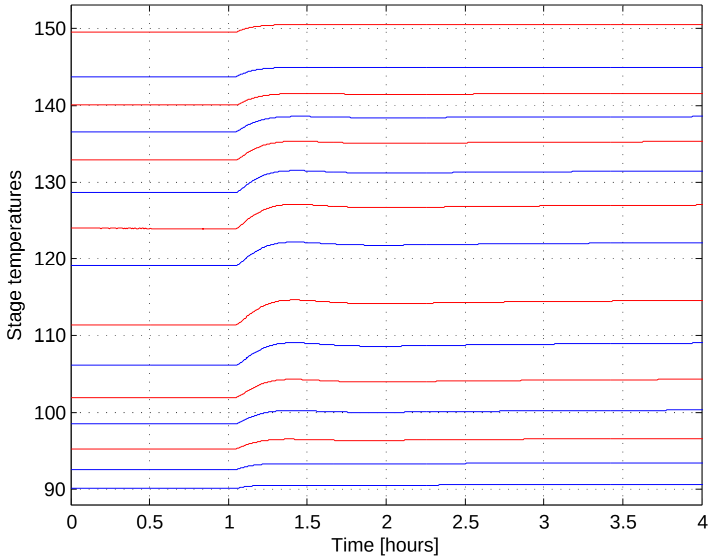


Figure 3.7: Response of all stages to setpoint change of TIC-100, ordered from stage 1 (lowest temp.) to stage 15 (highest temp.)

given by (3.22).

$$\Delta T_{\text{stages}} = \begin{bmatrix} 0.58 \\ 0.92 \\ 1.34 \\ 1.77 \\ 2.40 \\ 2.91 \\ 3.16 \\ 2.99 \\ 3.07 \\ 2.85 \\ 2.44 \\ 1.97 \\ 1.53 \\ 1.20 \\ 1.00 \end{bmatrix} \tag{3.22}$$

b) According to the max gain rule, we should choose the stage with most gain. By looking at max(ΔT_{stages}), it is clear that the stage with the most gain is stage 7. A setpoint of 111.4°C seems to give good compositions (same compositions as in task **2b**).
c) A step in the disturbance is performed, and the composition responses are shown in figure 3.8. See table 3.1 for the variations. The steady state values of the composition are used when calculating variations. The variations are significantly smaller after the max gain rule was applied to the temperature controller, as seen by comparing the ”before and after moving measurements” variations in table 3.1.

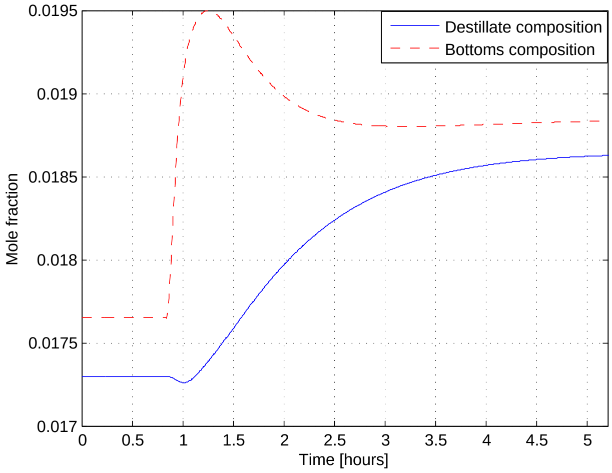


Figure 3.8: Composition responses to step in disturbance when controlling temperature in stage 7

d) Setpoints for FIC-100 and TIC-100 would be the DOFs. Important constraints are: $x_{n-butane} < 2\%$ and $x_{i-pentane} < 2\%$.

Table 3.1: Variations of the product flows and compositions. The variations are smallest for the last case, after moving the measurement. $x_{n-butane}$ varies a lot when we have no temperature control.

Case	Bottom ($x_{n-butane}$)	Top ($x_{i-pentane}$)	D (butanes)	B (C_{5+})
No tuning	0.0198 - 0.0980	0.0157 - 0.0076	±5250	±1600
Tuned basic controllers	0.0198-0.0980	0.0157-0.0076	*	*
With (tuned) temp. loop	0.0173-0.0237	0.0177-0.0141	*	*
After moving measurement	0.0173-0.0186	0.0177-0.0188	*	*

Table 3.2: Fill in the original tuning parameters here.

	Original tunings		
Controller	K_c (with dim.)	τ_i	Setpoint
LIC-100	$0.2[\frac{\%}{\%}]$	0.1	50%
LIC-102	$0.2[\frac{\%}{\%}]$	0.1	50%
TIC-100	(not used)	(not used)	(not used)

Table 3.3: Fill in the tuned parameters here.

	Tuned parameters		
Controller	K_c (with dim.)	τ_i	Setpoint
LIC-100	$3.509[\frac{\%}{\%}]$	40 min	50%
LIC-102	$10.67[\frac{\%}{\%}]$	120 min	50%
TIC-100	$3.11[\frac{°C}{\%}]$	20 min	149.1 °C

CHAPTER 4

DEBUTANIZER PART 2

4.1 Background

This assignment is an extension of the former ”Debutanizer case study”. In order to achieve good composition control when different disturbances excites our system, the degrees of freedom left in the system after closing the basic loops will be utilized. Figure 4.1 shows the system with the degrees of freedom outlined. The CV candidates are [1]:

- The mole fraction of the heavy key component in the top (i-pentane) - x_h^{top}
- The mole fraction of the light key component in the bottom (n-butane) - x_l^{btm}
- The pressure drop over the column, from bottom to top - ΔP_c
- The natural logarithm of x_h^{top} - x_{ln-h}^{top}
- The natural logarithm of x_l^{btm} - x_{ln-l}^{btm}

The DVs are the mass flow of the two feed streams.

4.2 The PID approach

One way to close the loops from the two DOFs to the output compositions is to use decentralized PID control. This means that one PID controller is used to control the bottoms composition using either reflux flow or column temperature, and one PID is used to control the top composition using the remaining DOF. This configuration may be implemented by following this procedure:

- In the PFD, select Object Palette. Find the object ”Control Ops”, and select this. A new menu will show up, with different controllers available. Select ”PID controller”, and insert it into the PFD.

[1] In order to keep track of the pressure drop over the column, the difference between bottom stage pressure and top stage pressure is included as an ”auxillary” CV in the identification routine.

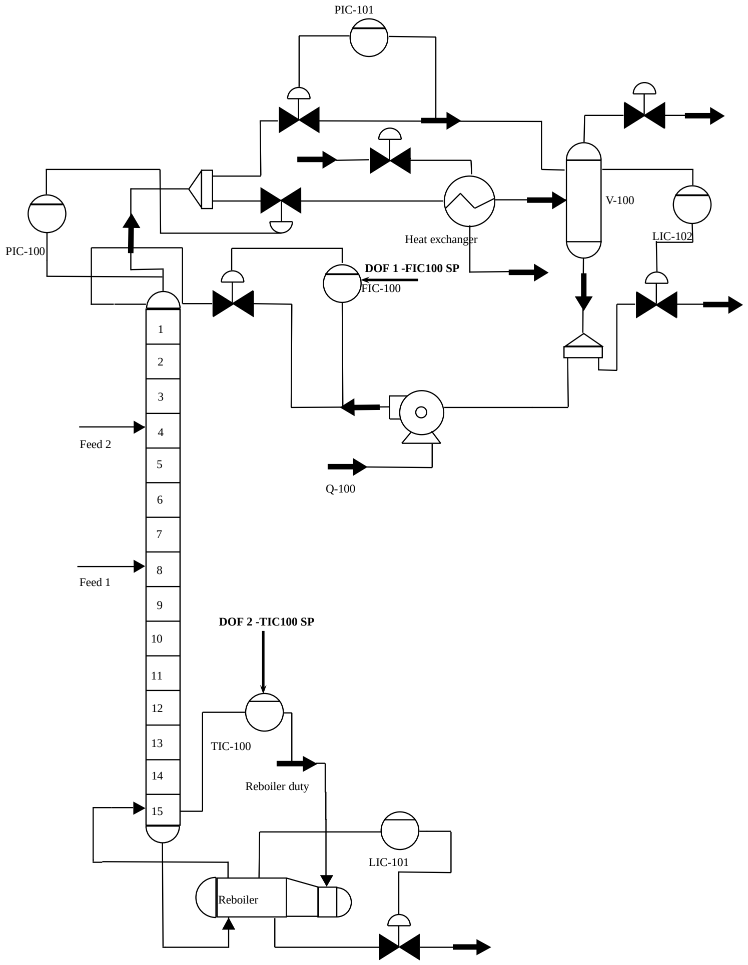


Figure 4.1: Debutanizer with the DOFs outlined

- Double click the controller, select the ”Connections” tab, and select the process variable (the variable you want to control). I.e., to control i-pentane in the top with FIC-100, select (PV - Butanes - Comp Mole Frac - i-Pentane). In addition, select ”Output Target Object” (the manipulated variable). In this case the manipulated variable will be the setpoint (SP) of either the reflux flow controller or the temperature controller.

- Select the ”Parameters” tab of the controller. Add suitable ranges for your controlled variable by filling in ”PV Minimum” and ”PV Maximum”.

After the controller is added, some auxilliary configuration is needed for the controllers to actually run. Since our controller is now writing to a slave controller, the slave controller needs to be set to ”Cascade”-mode. It is also possible to select ”Remote Setpoint” in the parameter property view of the slave controller. The master controller should be set to ”Auto” mode.

In order to tune the two master PID controllers, a model is needed. One way is to perform a steptest manually to aquire the model, but here we will use identification data from ”Profit Design Studio”. This tool will be discussed further in the MPC section.

4.3 The MPC approach

In this assignment a MPC controller is already implemented in the case file ”Debutanizer_MPC”. The objective of the assignment is to learn how to use the MPC, rather than the theory behind it.

4.3.1 Robust Multivariable Predictive Control Technology (RMPCT)

RMPCT is a trademark of Honeywell, and is their way of implementing Model Predicitive Control (MPC). It is a multivariable MIMO control application that controls and optimizes highly interactive industrial processes.

4.3.1.1 The controller model

RMPCT uses a model to predict process behavior. The overall process model is composed of a matrix of dynamic sub-process models, each of which describes the effect of one of the independent variables (MVs or DVs) on the controlled variables. A sub-process modell describes how the effect of an independent variable on a CV evolves over time.

RMPCT uses a generic form of sub-process model that provides a reasonably good description of the dynamic behavior of the process. This generic model contains a number of coefficients whose values determine the dynamic response of a sub-process. In order to make the generic model into specific models, the coefficient values where the predicted process responses agree with the actual process responses needs to be determined. This means that the model needs to be identified, or fitted to the actual process.

4.3.1.2 Identification of our model

In order to identify a model for this assignment, a step test is needed. Figure B.1 in appendix B shows an example of a step sequence along with the CVs, including the pressure drop over the column. This is the data used to identify the model for the MPC that is implemented in the case ”debutanizer_MPC”. See appendix B for more information regarding the identification of our model. A report of the identification is included in the file ”identification.xls”. Under the tab ”Model Views”, the identified transfer matrix is shown, with graphs of the impulse responses of each element of the matrix (from all MVs to all CVs, and all DVs to all CVs).

4.3.1.3 Loading the MPC (Profit Controller) in Unisim

In order to load the MPC controller in the case ”debutanizer_MPC”, you need to do the following:

- Press CTRL-P and select ”Case (Main)”. A controller named ”Profit_Controller” should show up, with connections to the output flows and the setpoints to reflux and reboiler duty.
- Click the controller, and select Model&Setup - Configuration. Load the files with extensions .xm and .xs respectively.[2] Press ”Load Profit Controller”, and check the ”Enable Profit Controller” box.
- The Profit Controller is initially set to ”Warm”-mode, which means it will only predict future CV values based on current inputs, and not control. Turning the controller on is part of this assignment.
- DO NOT turn on the controller (from WARM to ON) before MV and CV limits are configured correctly. The response might be scary.

[2] These are files generated from Profit Design Studio

4.4 Assignment

The assignment itself consists of two parts, respectively:

1. Controlling setpoints of the temperature and flow controllers with decentralized PID controllers, and tuning of these using the SIMC rules.
2. Control the same setpoints using an MPC controller (Profit Controller)

The following definitions yields for the rest of the assignment:

$$\begin{aligned} y_1 &= x_h^{top} \\ y_2 &= x_l^{btm} \\ u_1 &= \text{FIC-100 setpoint} \\ u_2 &= \text{TIC-100 setpoint} \\ d_1 &= \text{Feed1 massflow} \\ d_2 &= \text{Feed2 massflow} \end{aligned} \tag{4.1}$$

There are case files for each of these parts available, ”Debutanizer_PID” and ”Debutanizer_MPC”. Different strip charts are available in both cases. The strip charts you need will be specified throughout the assignment text.

The **overall objective** of this assignment is minimize x_h^{top} and x_l^{btm} while the pressure drop ΔP_c over the column is under a certain value. In Unisim, you will get a warning when ΔP_c is too high; the column will turn yellow.

It is expected that you comment on your results during the whole assignment. Try to relate your results to physical feasibility, and use control theory to explain the results.

Task 1

In this task we will use the case file ”Debutanizer_PID”. FIC-100 and TIC-100 will track setpoints from master PID controllers. You have to insert the PID controllers yourself as explained in section 4.2. The excel file ”Identification.xls” contains the transfer matrix from MVs to CVs, and from DVs to CVs.
a) Without disturbances, the system is on the form:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{G}(s) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.2}$$

From the excel file, we find the transfer matrix $\mathbf{G}(s)$ (angular frequencies have dimensions $\frac{1}{\text{min}}$):

$$\mathbf{G}(s) = \begin{bmatrix} \frac{-0.000322(116s+1)e^{-4s}}{1464s^2+76.5s+1} & \frac{0.00186(-2.36s+1)}{152s^2+59.1s+1} \\ \frac{-0.000306(-70.4s+1)}{344s^2+37.1s+1} & \frac{-0.00167(0.409s+1)}{31.8s^2+14.8s+1} \end{bmatrix} \tag{4.3}$$

Figure 4.2 shows a frequency plot of the RGA matrix[3]. Decide on how to pair inputs and outputs based on the RGA matrix *and* considering the different transfer functions of $\mathbf{G}(s)$[4].

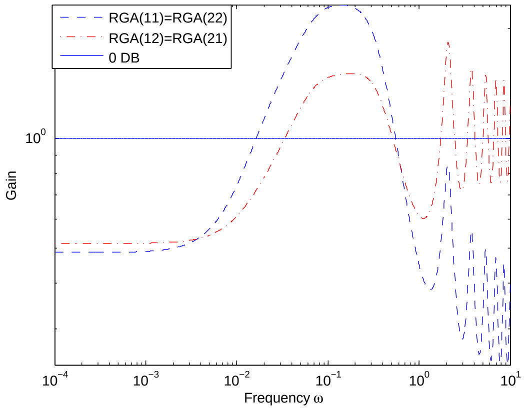


Figure 4.2: Frequency plot of the RGA matrix of $\mathbf{G}(s)$

b) Use the result from **a)** to implement PID controllers in Unisim for the key compositions. Use ranges for the PVs between 0 and 0.1. The output from the controllers will be given in %, hence the range of these are between 0% and 100%. Remember to put the slave controllers in cascade mode, and our new master controllers in auto.

[3] $RGA(G) = G \times (G^{-1})^{\top}$, ”×” meaning element-wise multiplication

[4] Hint: Pairing on transfer functions with large inverse responses should be avoided

c) *Read appendix A.3 about controller gain scaling in Unisim before you do this task. It is very important to scale the gain right in this case.*

Based on the transfermatrix and your pairings, tune the controllers using the SIMC tuning rules. Choose a closed loop time constant of about 15min for both controllers.

d) Select setpoints for the controllers so that both y_1 and y_2 are under 2% (with some margin). Have the spreadsheet for the compositions up, and perform some steps in the disturbances (relatively large steps). Write down some comments on how you think the control system is performing, and explain a scenario where a MPC controller would perform better than our decentralized PID controllers.

Task 2

In this task we will use an MPC controller to control compositions. The controller is already built, and consists of the two files ”mpc_abs.xs” and ”mpc_abs.xm”. The case file for this task is ”Debutanizer_MPC”. In the MPC, the pressure drop over the column is added as a CV, and is considered during composition control. This is to prevent the column from flooding. Absolute compositions are used in this MPC, mostly because it is more intuitive to set the limits when we use absolute compositions and not the logarithm of these.

a) *Do not start the simulation before the MPC is correctly configured.*

Load the MPC controller by following the procedure in 4.3.1.3. Apply the following configurations:

- In the ”Operation” tab of the Profit Controller, you set the constraints on the CVs and MVs. Initially, set the constraints listed in table 4.1. If profit controller will not let you set the limits, try checking and unchecking the SP box.

Table 4.1: Initial configuration of the constraints

Variable	LL Ent. (Low limit)	HL Ent. (High Limit)
Diffpress (ΔP_c)	15 kPa	19 kPa
Butanes (y_1)	0.0150	0.0250
C5+-1 (y_2)	0.0150	0.0250
TIC-100 (u_2)	110 °C	130 °C
FIC-100 (u_1)	$35\ \frac{\mathrm{m}^3}{\mathrm{h}}$	$45\ \frac{\mathrm{m}^3}{\mathrm{h}}$

- Under ”Operation”-”CTRL”, you may configure *Enginering Unit give-up factors.* These are factors used to compare different units, kind of like ”comparing oranges and apples”. These are specified for all CVs, and give the desired trade-off during control. A default configuration of these is proposed in table 4.2, but these factors may be used to tune the controller.

Table 4.2: Initial configuration of the EU Give-ups

CV	EU Give-up
Diffpress (ΔP_c)	5
Butanes (y_2)	0.001
C5+-1 (y_1)	0.001

b) The MPC controller is currently in ”Warm” mode, which is default. This means that it only predicts future CV values without any control action. Run the simulation, use the strip charts ”Bottoms”, ”Distillate” and ”Diffpress” and verify that the controller predicts correct steady state values. In the strip charts, CV values are red, while predictions are pink.

c) Since RMPCT is divided in two parts, control and optimization, it is possible to run the controller without running the optimization (pushing constraints). If the CVs are outside their bounds, the controller will open the limits by using *limit funnels* (see Profit Controller Concepts Reference Guide). After the controller have been run in warm mode for a while, change the controller mode to ”On” in the ”Operation” tab. Also change the two MV Modes to ”RMPC” mode. Verify that TIC-100 and FIC-100 are set to ”Cascade” mode (this should happen automagically). Open the strip charts ”Bottoms”, ”Distillate” and ”Diffpress”. Here are the limits, prediction and CV value plotted together (red should be the CV-value). Make sure all CVs are forced within their limits, and perform some relatively large steps on the disturbances (+20-30%).

d) Now it is time to run the optimizer! When the MPC is set to CTRL without any optimization, it will not push constraints, only assure that all CVs are inside their bounds. Select the ”OPT” tab under ”Operation” in the profit controller property window. As you see all the linear and quadratic coefficients are set to zero. By assigning values to these, the MPC will solve the optimization problem relative to our choice of coefficients. *Set the optimization speed factor under Model&Setup - Configuration.*

- The sign of the linear coefficients determines if the optimizer pushes the high or the low limit of the CV at hand. A positive value indicates minimization, while a negative value indicates maximization.

- The quadratic coefficients must be positive for the optimization problem to be convex.

- D. soft low limit and high limit are the backoffs from the limits. These should be used when noise is present, to assure some backoff from the limits.

- Tolerance is used to change the inital gap of the limit funnels.

Solve the overall objective of the assignment, which means minimizing y_1 and y_2 while the pressure drop over the column is maximized. Change the high limit of the pressure drop to find the maximal pressure drop before the column turns yellow. Try with different steps in disturbances, and write down comments on how the responses are to different disturbances.

CHAPTER 5

PROPOSED SOLUTION FOR DEBUTANIZER PART 2

Task 1

a) From the RGA analysis it seems like both pairings are equally good, the offdiagonal beeing slightly better at low frequencies, and for $0.02 < \omega < 10$. Since the closed loop dynamics will have time constants of about $15min$, the important frequency is about $\omega = \frac{1}{15}min = 0.067$. At this frequency the offdiagonal is clearly better. A problem with this RGA analysis is that it does not tell us anything about effective time delay elements of $G(s)$. Element $g_{21}(s)$ has an inverse response with a time constant of about $70min$, which is very bad when we want our dynamics in the scale of 15 minutes. This inverse response is in effect the same as a time delay ([Sko03]), hence we should not choose the offdiagonal pairing even if the RGA analysis tells us to.

b) Implementation of PID controllers, follow the assignment text. Remember that both PIDs should be configured to ”Direct”, since both the diagonal transfer elements have negative gain. In Unisim, processes with positive gain should be controlled with PIDs in ”Reverse” mode. This differs from other definitions of ”Direct” and ”Reverse”, but currently there is no standard notation for this.

c)

Reducing transferfunction $g_{11}(s)$

$$g_{11}(s) = -0.000322\frac{116s+1}{1464s^2+76.5s+1}e^{-4s} \tag{5.1}$$

$$\approx -0.000322\frac{116s+1}{(38.25s+1)(38.25s+1)}e^{-4s} \tag{5.2}$$

This transferfunction do have a complex conjugated pair of poles, but the imaginary parts are so small that we neglect them. First we approximate the zero using

Sigurds method:

$$\frac{116s+1}{38.25s+1} \approx \frac{116s}{38.25s} = \frac{116}{38.25} = 3.03 \tag{5.3}$$

This leaves us with a first order plus delay model:

$$g_{11,approx} = -0.000977\frac{1}{38.25s+1}e^{-4s} \tag{5.4}$$

This transfer function has the following parameters: $\tau_1 = 38.25min$, $\theta = 4min$ and $k = -0.000977$. The desired closed loop response time τ_c is 15 minutes. Tuning of the PID using the SIMC rules gives:

$$K_{SIMC} = \frac{\tau_1}{k}\frac{1}{\tau_c+\theta} = \frac{38.25}{-0.000977}\frac{1}{15+4} = 2060 \tag{5.5}$$

$$\tau_I = \min(\tau_1, 4\tau_c) = \tau_I = 38.25 \tag{5.6}$$

Unisim uses scaled controller gains, and we apply the formula stated in the assignment appendix (the ranges are also stated in the assignment text):

$$K_c = K_{SIMC}\cdot\frac{\text{range}(y)}{\text{range}(u)} = 2060\frac{0.01}{80} = 2.58 \tag{5.7}$$

Reducing transferfunction $g_{22}(s)$

$$g_{22}(s) = -0.00167\frac{0.409s+1}{31.8s^2+14.8s+1} \tag{5.8}$$

$$= -0.00167\frac{116s+1}{(12.19s+1)(2.61s+1)} \tag{5.9}$$

The small time constant zero is subtracted from the effective delay, and the half rule is applied:

$$k = -0.00167 \tag{5.10}$$

$$\tau_1 = 12.19 + \frac{2.61}{2} = 13.495 \tag{5.11}$$

$$\theta = \frac{2.61}{2} - 0.409 = 0.896 \tag{5.12}$$

The reduced transfer function is now

$$g_{22,approx} = -0.00167\frac{1}{13.495s+1}e^{-0.896s} \tag{5.13}$$

The tunings become:

$$K_{SIMC} = \frac{\tau_1}{k}\frac{1}{\tau_c+\theta} = \frac{13.495}{-0.00167}\frac{1}{15+0.896} = 508.35 \tag{5.14}$$

$$\tau_I = \min(\tau_1, 4\tau_c) = \tau_I = 13.495 \tag{5.15}$$

From the SIMC gain, we calculate the scaled gain:

$$K_c = K_{SIMC}\cdot\frac{\text{range}(y)}{\text{range}(u)} = 508.35\frac{0.01}{80} = 0.635 \tag{5.16}$$

Figure 5.1 shows the response of the controllers to setpoint changes. A spike down

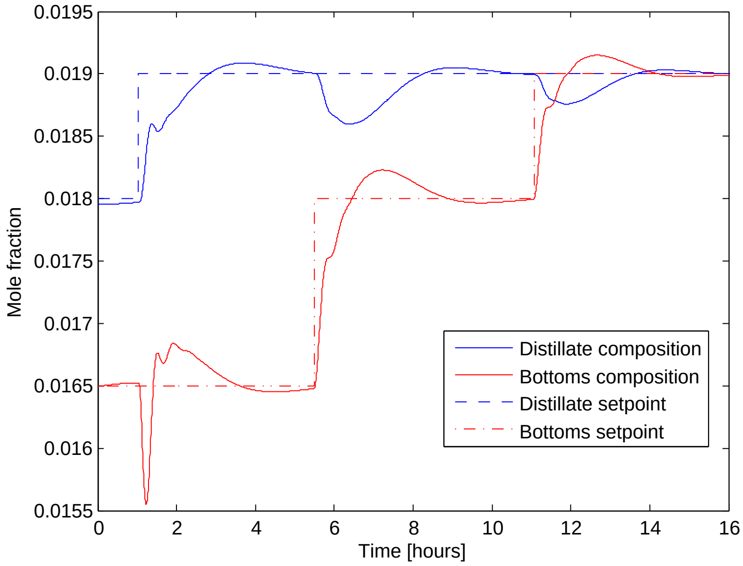
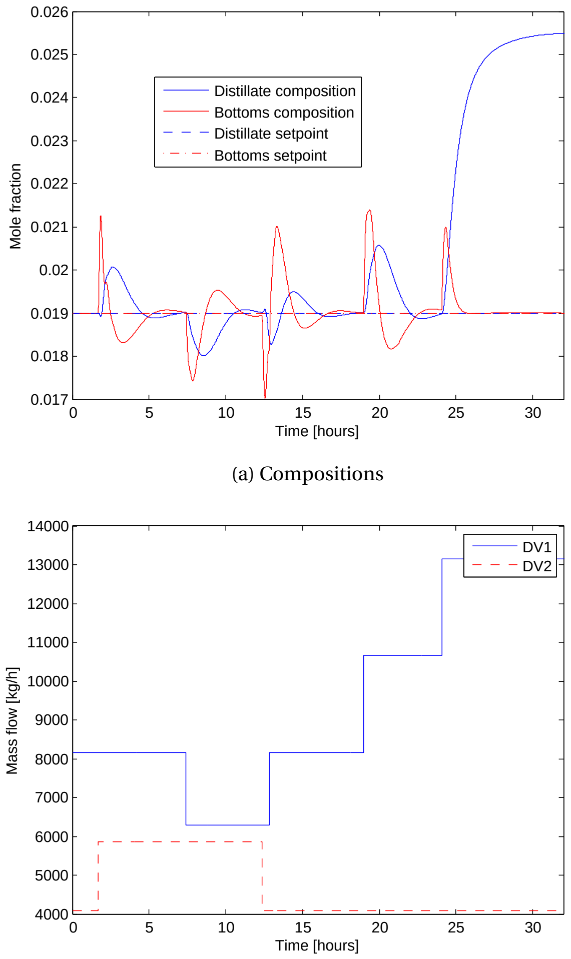


Figure 5.1: Response of the two composition controllers to setpoint changes

in bottoms composition occurs when the distillate setpoint is stepped up. This is because of the inverse response from reflux to bottoms composition, as seen by the transfer matrix. The time constant of this inverse response is reduced in closed loop due to the integral action in the bottoms composition controller. As seen by the responses, the interactions in the system are not very large. Some reasonably small over- and undershoots are observed when the setpoints changes. This corresponds well to the RGA analysis; all the RGA elements are pretty close to 1 (about 0.5) inside the bandwidth of the controllers.

d) By selecting setpoints of about 0.019 for both controllers, the response is pretty good to disturbances (both CVs are under 2% with steps in disturbances of about 30%). The responses to setpoint changes and steps in disturbances are surprisingly good with decentralized PID control. This may be because the system is not very interactive, as seen by the RGA matrix. However, we will get problems if one of the PIDs is unable to hold its setpoints. This may be viewed as ”greedy” control, because each of the PIDs is just trying to control its own variable, without considering the other. By using an MPC controller, it is possible to achieve a solution that is optimal with respect to different disturbances and our criteria for CV limits. This means that the MPC will not necessarily punish only one of the CVs when large disturbances are present, but even out the disturbance between the two of them (if we assume both compositions are equally worthy). It is also easy to set up configurations for economics with an MPC, by setting up the objective function and constraints so that profit is maximized, rather than just keeping the same setpoint. In addition we may include the differential pressure over the column as a constraint. Figure 5.2 shows the composition responses to disturbance steps. The disturbance steps are relatively large, as seen by the scale of figure 5.2b. The responses seem quite good, except some short spikes over 2% of the bottoms composition (usually the least valuable one). A problem occurs in the last step of DV1, as the distillate



(a) Compositions

(b) Disturbances

Figure 5.2: Composition responses to disturbances

composition escapes from the setpoint due to saturation of the reflux flow. This does not seem as a problem for the bottoms composition, the controller achieves its setpoint. This illustrates the problem with this control configuration.

Task 2

a) This should work ok by following the instructions in the assignment. Try unchecking the ”Use PSD settings” box before filling in the limits. If you run Windows Vista, you have to do this for the predictions to be correct.
b) The steady state predictions and the future predictions should equal the CV values since the case is in steady state at this point. See figure 5.5 for the steady state prediction errors. These errors are very small to zero in steady state.

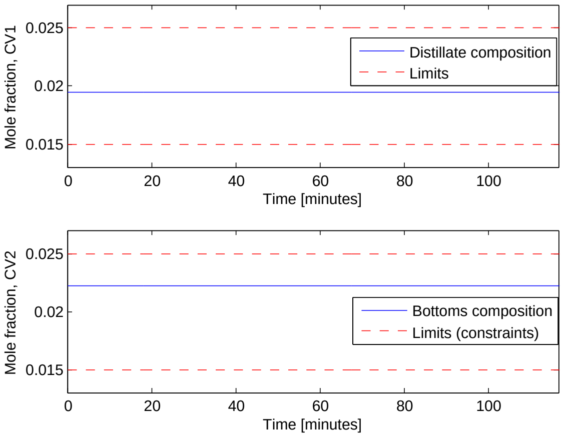


Figure 5.3: Steady state compositions values when the MPC runs predictions only

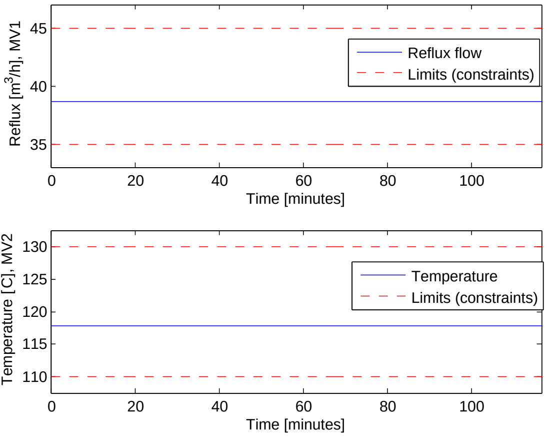


Figure 5.4: Steady state input values when the MPC runs predictions only

c) When running the MPC, nothing should happen to the CVs. This is because the values already are within their limits, and we are currently running with no optimization. When performing steps in the DVs, the MPC counteracts the disturbances in a satisfying manner. The difference from the decentralized PID approach is that we now have the pressure over the column as a constraint, so the MPC will prevent the column from flooding when large disturbances are present (if we select lower EU giveup factors on the differential pressure, the MPC will be more aggresive on keeping the pressure constraint). See figure 5.6 for the response to different disturbance steps, and figure 5.7 for the prediction errors. Note that the MPC does not change inputs unecessarily when the CVs are inside their limits.

d) An optimization speed factor of 5 is used, and the linear coefficients of y_1 and y_2 are set to 1 (positive value = minimizing), while the linear coefficient of the pres-

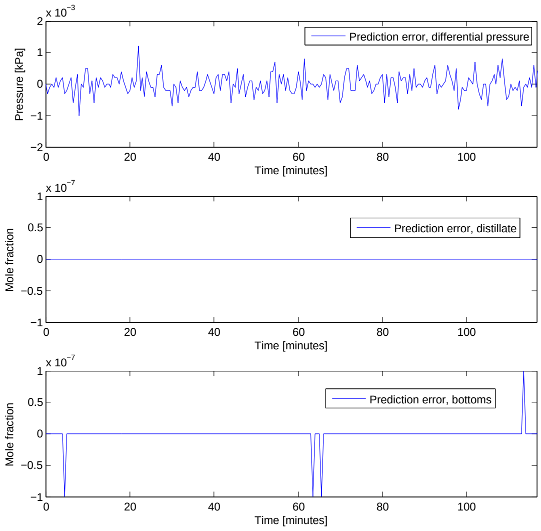
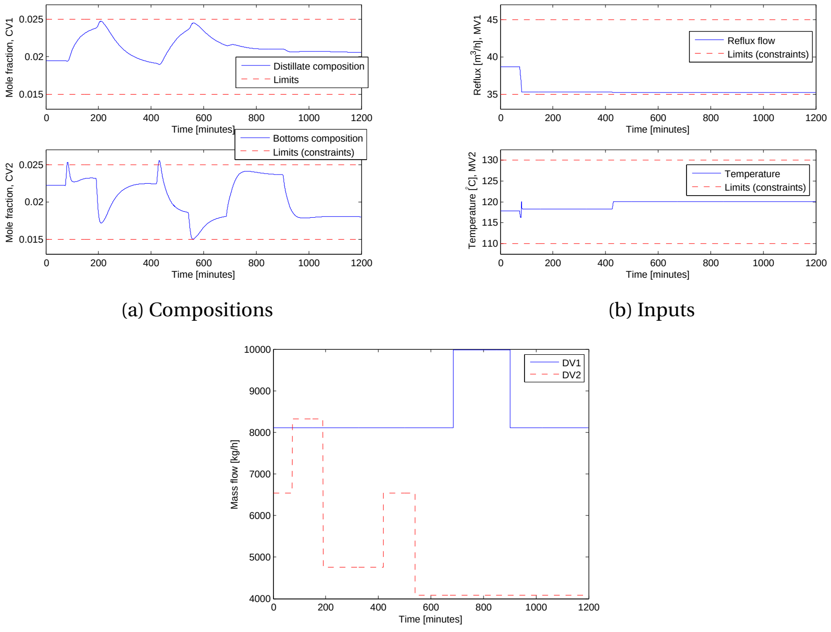


Figure 5.5: Steady state prediction errors



(a) Compositions

(b) Inputs

(c) Disturbances

Figure 5.6: Composition responses and input usage to disturbances with MPC in control mode

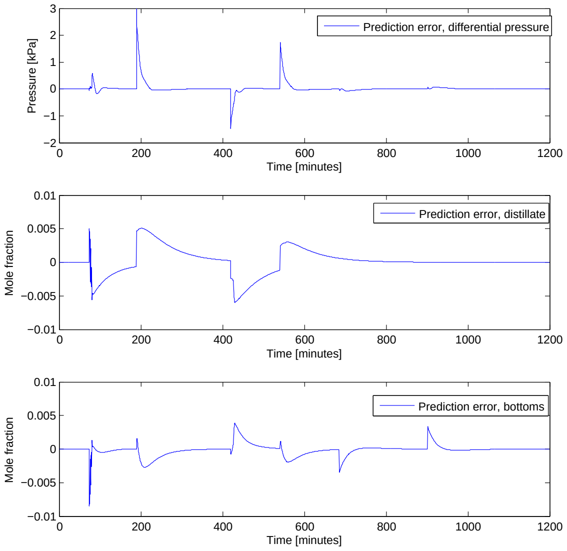
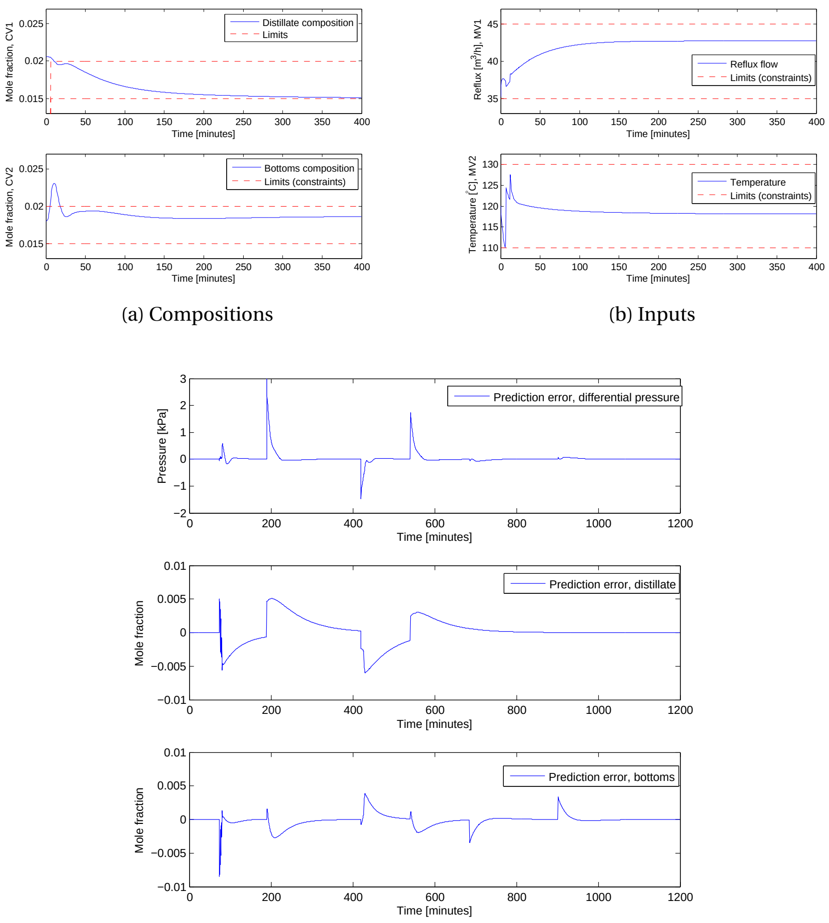


Figure 5.7: Prediction errors after steps in disturbances

sure is set to −1 (negative value = maximizing). Quadratic coefficients of 1 is used on all CVs. No optimizing regarding the MVs are performed. It is observed that it is not really necessary to maximize the pressure, because this will automatically be maximized when minimizing the compositions (this is one of the properties of the process). The responses are relatively quick with this optimization factor, and the pressure constraint is active while the compositions are minimized with respect to their weights. Selecting a higher value on the linear coefficient of the top composition would be a good idea in many cases, since this is usually the most important product.

Constant disturbances

Figure 5.8 shows the response after the optimizer is turned on, with constant disturbances. Notice that the distillate compositions has a higher priority, so in the optimal solution the low limit of the distillate composition is an active constraint. In addition, the high limit of the differential pressure is active.



(a) Compositions

(b) Inputs

(c) Prediction errors

Figure 5.8: Composition responses, input usage and prediction errors with MPC in control mode with optimization

Step in disturbances

In order to ”compare” the response of the MPC controller to the response of the decentralized PID controllers, the disturbances are stepped. The steps are so big that the MPC controller is unable to hold its CVs inside their respective limits. Figure 5.9 shows the disturbance step sequence. The responses of the CVs are showed in figure 5.10, together with the input usage. From the theory behind RMPCT (section 1.2.1) it is recalled that when the controlled variables are outside their limits, the distances from the limits to the actual values of the variables are added up in a cost function that is to be minimized. In this case, the controller choose to let both compositions exceed their limits, but at the same time the pressure limit is not exceeded. This is supposed to be the optimal solution relative to the configurations applied to the controller, which is that it is more important to hold the pressure constraint than the composition constraints (hence a high EU give-up factor on the pressure). Note that the disturbance $DV1$ in this case is of very large magnitude, so this test is pushing the limits to see how the MPC controller is performing in extreme cases. When applying the same disturbance with the PID approach, there was no constraint on the pressure, hence the column was very likely to flood in addition to one of the composition values being far away from its setpoint. In the MPC case, the set-point error is distributed between the two compositions, and the pressure is outside the ”flooding zone” as well. A value of 19.26kPa for the upper limit of the pressure with the disturbances at default configuration is found by adjusting the limit while simulating.

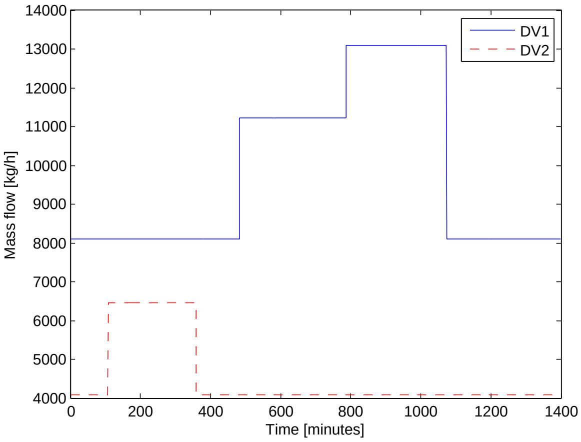
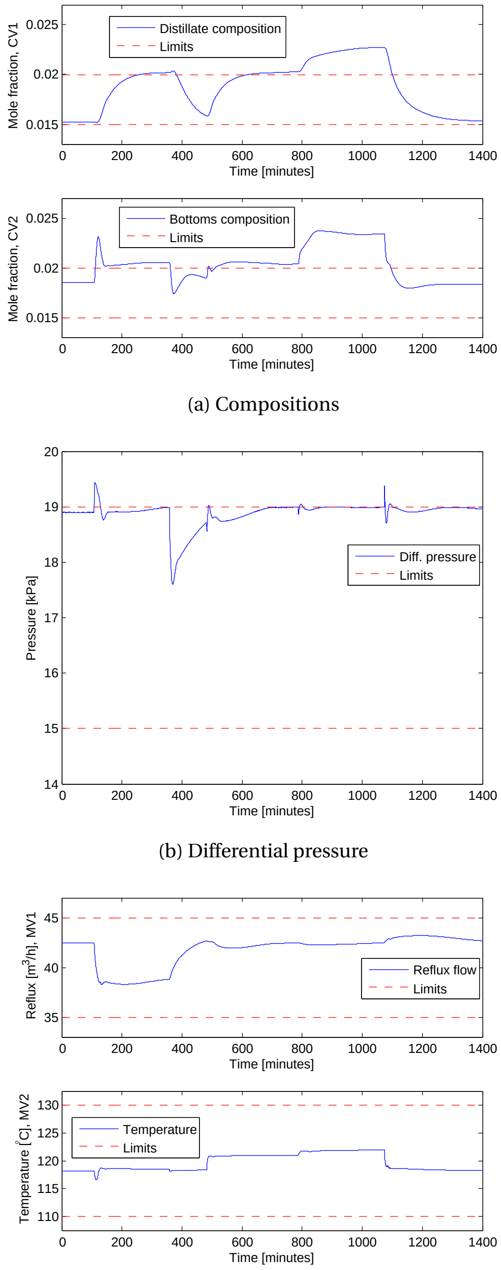


Figure 5.9: Disturbance sequence used to test the MPC controller



(a) Compositions

(b) Differential pressure

(c) Inputs

Figure 5.10: Composition responses, input usage and differential pressure with MPC in control mode with optimization, with steps in disturbances

CHAPTER 6

DISCUSSION

The main purpose of this project was to develop case studies that shows how control theory is utilized in a more 'practical' setting than other academic assignments. The idea is that students should encounter a simulation environment that is somewhat closer to a real plant installation. The abstraction of viewing actuators as "*u*'s" and outputs as "*y*'s" is the common academic approach when teaching control theory. By controlling a plant in UniSim, the user hopefully gets a more "realistic" view of what the manipulated and controlled variables actually are (e.g controller valve positions / flows or temperatures). The two first sections of this chapter are discussions considering the pedagogics of the assignments, and the last section is regarding the course in which the assignments were used during the fall of 2008, and some feedback from students who did the assignments.

6.1 Basic control theory

The first part of the case study involves tuning of some oscillating level loops, in addition to introduce temperature control. Oscillating level loops are not necessarily a general problem for a process plant, it is dependent on what the disturbances are. In this case the level loops are oscillating after a step in one of the disturbances, and the controller valves used for level control affects the output flows (distillate, bottoms) directly. Because of this poor tuning, the flows out of the column are oscillating with high amplitudes. The intention is that students should understand the difference between "tight" and "smooth" control. By tuning these levels using Sigurds rule or the SIMC tunings with a relatively high closed loop response time, the change of controller valve positions should be so smooth that the flows out of the column are about constant, while the level oscillations are removed for low frequency disturbances. Hence, the goal is that level control is achieved using slow variations on the controller valves. If the goal had been "tight" level control, the closed loop response times could be dramatically reduced. The problem is that this would result in an extensive use of actuation, which is bad for the flows out of the column (which could be input disturbances to other process units).

The students are asked to perform a step in one of the disturbances after each task is completed in order to see how this affects the compositions, which are the variables we really want to control in this case. It is shown that tuning of the levels alone do not have significant effect on the output compositions, only that small oscillations are removed. In order to achieve good control of the compositions, temperature control using reboiler duty is introduced. The intention is here to show how to identify a first order process model by performing a simple step test, and to use the SIMC tuning rules [Sko03] to tune the temperature controller based on this identification experiment. In order to reduce the composition variations to disturbances while the temperature is at constant set-point, a simplified version of the *maximum-gain rule* (see section 1.1.3) is applied. In this simplified version, the distillation column stage with most gain from reboiler duty to temperature is selected without considering optimal variations or implementation errors, and the results are verified by confirming that the variations of the output compositions are lower after the rule is applied (assuming both bottoms and distillate components are equally ”worthy”). This is supposed to show an example of the essence of the maximum gain rule, namely that variations of the (indirect) controlled variables (compositions) may be reduced by only changing a measurement location.

6.2 Supervisory control

For the supervisory control layer, two different approaches are used in order to compare the behavior of the system, namely decentralized PID control and multivariable control (MPC). The objective is to line out the main differences between decentralized and multivariable control, and for students to understand the basics of cascaded control.

6.2.1 Cascaded PID control

The ”PID approach” is used as an example in this case because the system at hand is not very interactive (see figure 4.2). This means that decentralized PID controllers may perform relatively well when it comes to set-point tracking and disturbance rejection. The main pedagogic objectives of this part is to show how to apply frequency dependent RGA analysis, and in addition to understand the limitations of this analysis, and to show how cascaded control works. The focus is also on that students should understand why decentralized control is not always desired even if the system is only slightly interactive (constraint handling is easier with an MPC controller). Yet another issue is that the students will have to decide on if the controllers should be set to ”Direct” or ”Reverse” mode. This decission should be made by looking at the sign of the system gains; if the gain is positive the controller should be set to ”Reverse” mode, and if the system gain is negative the controller should be set to ”Direct”.

6.2.2 The use of RMPCT

By using Honeywells MPC implementation, a step away from the theory behind MPC is taken, and the focus is in a wider sence on the use of MPC. This project is in no way trying to challenge teaching of theory behind MPC, the idea is that students should get the feeling of using an MPC controller for a practical purpose. Hopefully this could be a nice supplement to the teaching of theory on MPC.

Since the purpose of this project is to show how to use an MPC-controller, the focus of the assignment regarding MPC is not to build the controller from scratch. A summary on how the model was derived is included in the assignment, but students are not supposed to do this themselves in this case. There are numerous possible extensions to the MPC assignment, e.g let the students read papers on dynamic control of distillation columns (e.g. [Sko07] and [Sko04]) and perform an identification experiment themselves. The controller may then be generated using Profit Design Studio.

6.3 Application and feedback from students

The two assignments were given in prof. Sigurd Skogestads course ”Advanced Process Control” at NTNU during the fall of 2008. This course is taken by graduating master students, as well as PhD students. The course consist of three main parts, namely

- **Plantwide control**
- **Tuning** - with focus on using the SIMC tuning rules and reducing high order transfer functions.
- **Multivariable control** - RGA analysis and model predictive control (MPC). It is expected that students that attend this course have a basic knowledge about MPC.

With respect to this course, the two assignments are mainly focusing on the two latter parts, tuning and multivariable control, but elements from plantwide control is also present (e.g. maximum gain rule). In order to get some feedback after the assignments were carried out in this course, the following survey were given to the students:

1. What is your academic background, and what is your current academic level? (Master student or PhD student)
2. How well do you feel you know basic control theory (from 1 to 7)?
3. How well do you feel you know more advanced control theory (from 1 to 7)?
4. Regarding debutanizer part 1: What new elements were introduced for you, or increased your understanding of those elements? (E.g. direct/reverse PID, maximum gain rule, SIMC-tuning, Sigurds rule or other things)
5. Regarding debutanizer part 2: What new elements were introduced for you, or increased your understanding of those elements? (E.g. cascaded control, decentralized control, pairing inputs/outputs, controller gain scaling, model identification, constrained control using MPC)
6. To what extent did you feel the assignments were useful for you? (From 1 - unnecessary, to 7 - most necessary)

Six answers to this survey were delivered from (anonymous) participants of the course. All the different answers from each participant will not be listed here. The answers that requires a number from 1 to 7 will be gathered to a mean value from all six participants, and for the questions that requires a written answer, the answers will be gathered in a list. This is not intended to be a proper statistical analysis, it is merely to get an indication on how this kind of assignments are viewed upon by the participants. The results from the survey is as follows:

1. The participants of the survey includes both master and PhD students, and also with different backgrounds. Some of the participants have some background in control theory but mostly chemistry, while others are participants with a pure control theory background.

2. The mean value here is 5, which shows that the participants in average have pretty strong knowledge about basic control theory. Note that this is very likely to influence the participants opinion on the assignments, so this survey will give some general opinions for a sample with a pretty strong background in basic control theory.

3. The mean value here is 4, which is a bit lower than for basic control theory. It is assumed that this sample has mediocre knowledge about advanced control theory.

4. To this question there were several different answers which will be listed here:

    - ”I understood the practical use of the SIMC rules, and use of the maximum gain rule”
    - ”I understood the importance of scaling in a practical sense”
    - ”I gained understanding of the maximum gain rule and Sigurds rule”
    - ”I learned the difference between direct and reverse PID control and use of Sigurds rule”
    - ”Most of the things were new to me”
    - ”I understood how direct/reverse PID worked, how to use the SIMC tuning rules and Sigurds rule”

5. Same as for the question above:

    - ”Engineering unit give-up factors were new to me”
    - ”I gained understanding regarding pairing of inputs/outputs, MPC controllers in general, and the funnel concept was new to me”
    - ”Most of these things were new for me”
    - ”I understood the concept of decentralized and cascaded control, controller gain scaling and pairing of inputs/outputs better after the assignment”
    - ”Good to know how to use/implement cascaded controllers in a more practical setting”
    - ”Gained basic knowledge of practical use of MPC controllers”

6. For this question, the mean value was 5.83. This indicates that, for this sample, the assignments worked out pretty well. Note that this result yields for this sample only, the results could be completely different if the assignments were given in e.g. an undergraduate course.

CHAPTER 7

CONCLUSIONS

In this chapter, conclusions on two different topics will be given. First, the results from the two assignments will be summarised (technical conclusion), and then a conclusion on the main objective of this project will be given (pedagogical conclusion).

7.1 Technical conclusion

This report have shown some of the classical issues and solutions regarding control of a distillation column with the LV-configuration. In part 1 of the case study, the importance of proper tuning of levels was motivated, and temperature control of the distillation column was motivated by studying the variations of the output compositions with and without temperature control. It was shown that keeping the temperature at a given stage in the column at a constant setpoint would give significantly lower variations of the output compositions due to disturbance steps. As an extention to this, the use of a simplified version of the maximum gain rule was shown to provide even lower variations due to disturbance steps.

The second part of the case study showed two different ways of implementing the supervisory control layer for the debutanizer. First, the cascaded PID approach where tested, where decentralized PID controllers controlled the setpoints of the regulatory level controllers in order to control compositions directly. This turned out to work surprisingly well (good disturbance rejection and set-point tracking) with moderate disturbance magnitudes, mostly due to the small interactions in the system. Problems ocurred when the disturbances where so large that one of the controllers suffered from saturarion in its manipulated variable. This problem was handled better by the MPC controller, which minimized the error from the limits on compositions, and in addition monitored the differential pressure over the column to prevent flooding. This emphasises one of the main advantages of MPC controllers, namely the ability to handle constraints.

7.2 Pedagocic conclusion

During the work with the assignments, students encountered expressions such as ”CV”, ”MV”, ”PV” and ”OP”. From conversations with some of the students who attended the course, the author was told that all these expressions were new to them. This was one of the ideas with this case study, that the students should get familiar with these expressions, and see the link between the acedemic way of viewing control theory to the industry way. The survey results shows that the general opinion was that this case study was useful, maybe in particular for the students with a pure chemistry background. It seemed like students were positive to this way of learning control theory, by applying the theory directly to a simulated model, closer to practice than just a ”stack of transfer functions”.

The final conclusion to this project is that it seems useful to motivate students with a practical case study where the theory is applied directly, and at the same time try to tighten the gap between theory and practice (academia and industry).

References

[DS89] D.A Mellichamp D.E Seborg, T.F Edgar. *Process Dynamics and Control*. John Wiley & Sons, 1989.

[Hon07a] Honeywell Process Solutions. *Advanced Process Control - Identifier, users guide*, r310 edition, July 2007.

[Hon07b] Honeywell Process Solutions. *Advanced Process Control - Profit Controller, concepts reference guide*, r310 edition, December 2007.

[Mac02] J.M. Maciejowski. *Predictive Control: With Constraints*. Prentice Hall, 2002.

[Sko03] Sigurd Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13:291–309, 2003.

[Sko04] Sigurd Skogestad. Control structure design for complete chemical plants. *Computers and Chemical Engineering*, 28:219–234, 2004.

[Sko07] Sigurd Skogestad. The dos and don’ts of distillation column control. *Trans IChemE Part A*, 85:13–23, 2007.

[SP05] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control - Analysis and design*. John Wiley & Sons, Ltd, 2005.

A Regarding debutanizer assignments

A.1 Tuning of level controllers in UniSim

This is a brief guide on how to tune level controllers in UniSim using the SIMC-tuning rules.

Since the process from input to output is a pure integration, a step test can easily be done in UniSim by following these steps:

1. Put the controller you want to tune in manual by double-clicking the controller, choose the ”Configuration” tab, and set ”Mode” to ”Man”. Make sure that the system is at steady state before you do this.

2. Stop the integrator by pressing the red light in the menu bar, or open the integrator by pressing *CTRL+I*. Press the Reset Sim Time button to, as it implies, reset the simulation time.

3. Perform a step on the actuator by changing the OP-value in the controller property view, e.g 5%.

4. Start the integrator, and stop it after some time (e.g 100 minutes).

5. Calculate $k' = \frac{K}{\tau}$ by the formula $k' = \frac{\Delta y}{\Delta t \cdot \Delta u}$

6. Choose a appropriate closed loop time constant τ_c

7. Calculate the controller gain K_c and integral time τ_i using the SIMC rules. $\tau_i = 4(\tau_c + \theta)$, but in our cases the deadtime θ is neglectible. Remember that a integrating process has an ”infinite” time constant.

A.2 Tables

Table A.1: Variations of the product flows and compositions. Fill in the on the form: ”Steady state value before step in disturbance - Steady state value after step in disturbance” for $x_{n-butane}$ and $x_{i-pentane}$. For D and B we are interested in the amplitudes of the oscillations a while after the step in disturbance. A * means that no values are necessary.

Case	Bottom ($x_{n-butane}$)	Top ($x_{i-pentane}$)	D (butanes)	B (C_{5+})
No tuning	-	-	±	±
Tuned basic controllers	-	-	*	*
With (tuned) temp. loop	-	-	*	*
After moving measurement	-	-	*	*

Table A.2: Fill in the periods and the amplitudes of the oscillations of the two levels here.

Process variable	Level oscillation period	Level oscillation amplitude
Reflux drum level (V-100)		
Reboiler level		

Table A.3: Fill in the original tuning parameters here.

Original tunings

Controller	K_c (with dim.)	τ_i	Setpoint
LIC-100	$[\frac{\%}{\%}]$		
LIC-102	$[\frac{\%}{\%}]$		
TIC-100	(not used)	(not used)	(not used)

Table A.4: Fill in the tuned parameters here.

Tuned parameters

Controller	K_c (with dim.)	τ_i	Setpoint
LIC-100	$[\frac{\%}{\%}]$	40 min	
LIC-102	$[\frac{\%}{\%}]$		
TIC-100	$[\frac{^\circ C}{\%}]$		

A.3 Controller gain scaling in Unisim

The controller gain in Unisim is defined as a dimensionless value, where both the output from the controller (u) and the controlled variable (y) are scaled to be in the range $[0,1]$. The equation for the controller gain is:

$$K_c = \frac{\Delta u/\text{range}(u)}{\Delta y/\text{range}(y)} \tag{A.1}$$

Table A.5: Fill in the temperatures in the column before and after the setpoint step of TIC-100

Stage	Temperature before step	Temperature after step
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

where K_c is defined as the Unisim controller gain. If you calculate K_c using the SIMC rules, you will get values for K_c **with** dimensions. This value is on the form

$$K_{SIMC} = \frac{\Delta u}{\Delta y} \tag{A.2}$$

where K_{SIMC} is the controller gain you get when using the SIMC rules.

Hence, after calculating the correct SIMC gain, you calculate the ”Unisim”-gain by the following formula:

$$K_c = K_{SIMC} \cdot \frac{\text{range}(y)}{\text{range}(u)} \tag{A.3}$$

As an example, both ranges for the reflux flow and the column temperature are 80, with different units. This means that if you apply formula (A.3) to calculate the Unisim gain when controlling mole fractions with range 0.1 with master controllers, you will get

$$K_c = K_{SIMC} \cdot \frac{\text{range}(y)}{\text{range}(u)} = K_{SIMC} \cdot \frac{0.1}{80} = \frac{1}{800} K_{SIMC} \tag{A.4}$$

Note that the reflux and temperature are manipulated variables (u’s) for the master controllers.

B Identification for debutanizer part 2

B.1 Definition of the control configuration

In order to identify a model for the debutanizer, a step test of the system will be performed. Since it is not known before the test what the best controlled variables (CVs) for this purpose are, different CVs will be logged during the steptest for comparison. Skogestad [Sko07] proposes logarithmic transformations for the purpose of counteracting the strongly nonlinear response of distillation columns. This benefit is good for the initial part of the dynamic response, which is important for feedback control purposes. Hence the logarithm of the top and bottom compositions are included as CV candidates.[1]

B.1.1 The manipulated variables (MVs)

The current setup provides two degrees of freedom for optimal control, namely the setpoints of the reflux controller and reboiler controller. At this stage it is assumed that both controllers are tuned so that the dynamics from reflux and reboiler duty is much faster than the dynamics from column temperature to compositions.

B.1.2 The disturbance variables (DVs)

In order to simplify the analysis, the disturbances are modelled as changes in feedrates (Feed 1 and Feed 2) into the column, and not as changes in compositions of the feeds.

B.1.3 The different CV candidates

The main purpose of the distillation column is to separate light key components from the heavy key components. In this case the input feed streams are compositions of the components in table B.1. We separate between heavy and light between butane and pentane, i.e. all components heavier than butane are ”heavy” components. The *key components* are the components that varies the most to input usage, namely i-pentane in the distillate flow, and n-butane in the bottoms flow. This corresponds to the max gain rule, see section 1.1.3. In order to control composition, one variable of D and one variable of B should be chosen as CVs. In addition, the differential pressure over the column is added as an auxilliary CV. This is because we want to constrain this variable in the MPC controller to prevent the column from flooding.

[1] For the assignment, an MPC with the normal compositions as CVs is used. This way it is more intuitive to adjust the constraints on compositions, i.e. directly in the controller property view.

Table B.1: Components of the input feed streams

Component type	**Molecular mass** $[\frac{g}{mol}]$
i-Butene	56.108
i-Butane	58.120
n-Butane	58.120
i-Pentane	72.150
n-Pentane	72.150
n-Hexane	86.170
n-Heptane	100.200
n-Octane	114.220

Table B.2: Mole fraction variables and definitions

Variable	**Description**
B	Bottoms output flow
D	Distillate output flow
ΔP_c	The pressure difference between the bottom and top stage in the column
$\mathbf{x}_D$	Mole fraction of the heavy key component of the distillate flow
$\mathbf{x}_B$	Mole fraction of the light key component of the bottoms flow
$\mathbf{x}_D^{ln}$	The natural logarithm of $\mathbf{x}_{D_h}$
$\mathbf{x}_B^{ln}$	The natural logarithm of $\mathbf{x}_{B_l}$

B.2 Identification experiment

A good model requires a good identification experiment. The objective is to catch both steady state properties, and the dynamics up to about one and three system time constants respectively. Figure B.1 shows the step sequence along with the CV candidate responses. The identification tool used is Honeywells Profit Design Studio Identifier ([Hon07a]). This identifier generates a multivariable ARX model based on the data from the step test, and also represents the model in Laplace form. The ARX model is used by the Profit Controller (RMPCT) in UniSim, while the Laplace model gives us information about the different gains, time constants and zeros of the system transfer matrix. This tool will not be discussed any further here, see [Hon07a] for more information. The model identified by Profit Design studio is, on Laplace form:

$$\mathbf{y}(s) = \mathbf{G}(s)\mathbf{u}(s) + \mathbf{G}_d(s)\mathbf{d}(s) \tag{B.1}$$

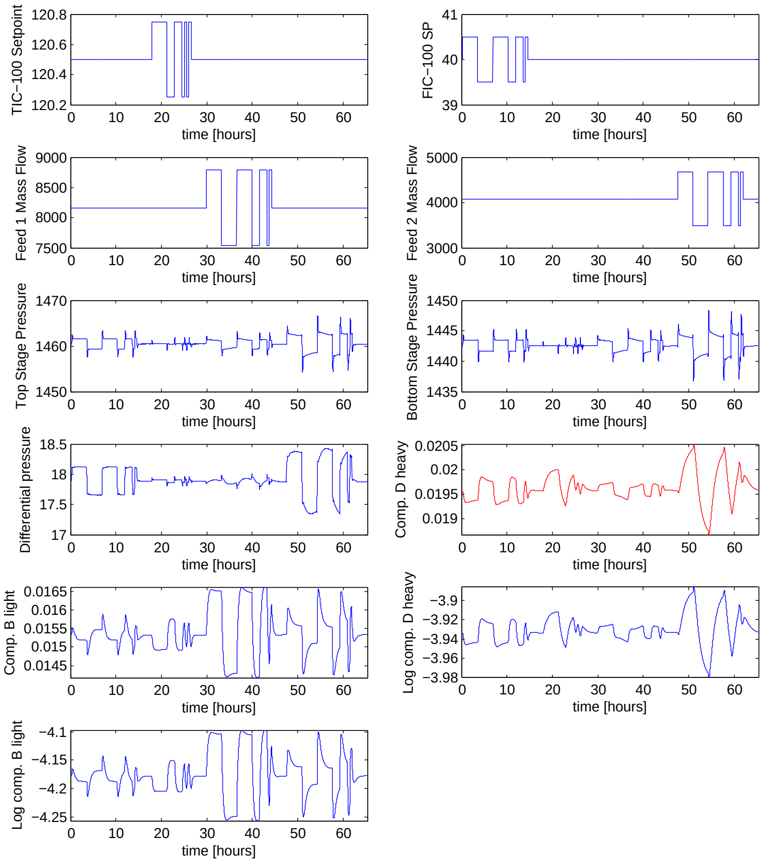


Figure B.1: Step test for identification

where

$$
\begin{aligned}
\mathbf{y} &= \begin{bmatrix} \Delta P_c & \mathbf{x}_D & \mathbf{x}_B & \mathbf{x}_D^{ln} & \mathbf{x}_B^{ln} \end{bmatrix}^\top \\
\mathbf{u} &= \begin{bmatrix} \text{FIC-100}_{\text{set-point}} & \text{TIC-100}_{\text{set-point}} \end{bmatrix} \\
\mathbf{d} &= \begin{bmatrix} \text{Feed 1}_{\text{mass flow}} & \text{Feed 2}_{\text{mass flow}} \end{bmatrix} \\
\mathbf{G}(s) &= \begin{bmatrix}
0.466\frac{(44.3s^2+162s+1)}{305s^3+116s^2+21.9s+1} & 0.0941\frac{12s+1}{4.24s^2+4.12s+1} \\
-0.000322\frac{116s+1}{1464s^2+76.5s+1}e^{-4s} & 0.00186\frac{-2.36s+1}{152s^2+59.1s+1} \\
-0.000306\frac{-70.4s+1}{344s^2+37.1s+1} & -0.00167\frac{0.409s+1}{31.8s^2+14.8s+1} \\
-0.0188\frac{-187s^2+106s+1}{4618s^3+1492s^2+90.8s+1} & 0.0931\frac{1}{52.3s+1}e^{-8s} \\
-0.0183\frac{-72.5s+1}{278s^2+33.4s+1}e^{-1.5s} & -0.111\frac{1}{13.8s+1}e^{-s}
\end{bmatrix} \\
\mathbf{G}_d(s) &= \begin{bmatrix}
5.79\cdot 10^{-5}\frac{-29.8s+1}{308s^2+35.1s+1} & 0.000855\frac{1}{6.29s+1} \\
3.53\cdot 10^{-7}\frac{166s+1}{3742s^2+282s+1}e^{-5s} & 1.56\cdot 10^{-6}\frac{-13.4s+1}{1218s^2+85s+1} \\
1.76\cdot 10^{-6}\frac{48.6s+1}{835s^2+57.9s+1} & 3.89\cdot 10^{-7}\frac{152s+1}{306s^2+42.7s+1}e^{-s} \\
1.03\cdot 10^{-5}\frac{-0.0547s^2-0.285s+1}{97.3s^3+102s^2+16.1s+1} & 7.95\cdot 10^{-5}\frac{-13.3s+1}{1223s^2+85.1s+1} \\
0.000114\frac{48.5s+1}{830s^2+57.8s+1} & 2.48\cdot 10^{-5}\frac{-84.6s^2+161s+1}{315s^3+346s^2+44.9s+1}
\end{bmatrix}
\end{aligned}
\tag{B.2}
$$