TTK4210 Advanced Control of Industrial Processes
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Spring 2018 - Assignment 6
Due date: Sunday 15 April at 23:59

Note: You will have to deliver **individual reports** - remember that this project counts for **30% of your grade** in this course!

Kårstø Statpipe Butane Splitter

The project deals with the control of distillation columns, based on a model of a butane splitter at Kårstø. See Figure 1 for a simple overview of the model.

If you download `ButaneSplitter-model-detailed.pdf` from Blackboard, you might print the detailed model in large scale (for example A2 or A1 format).

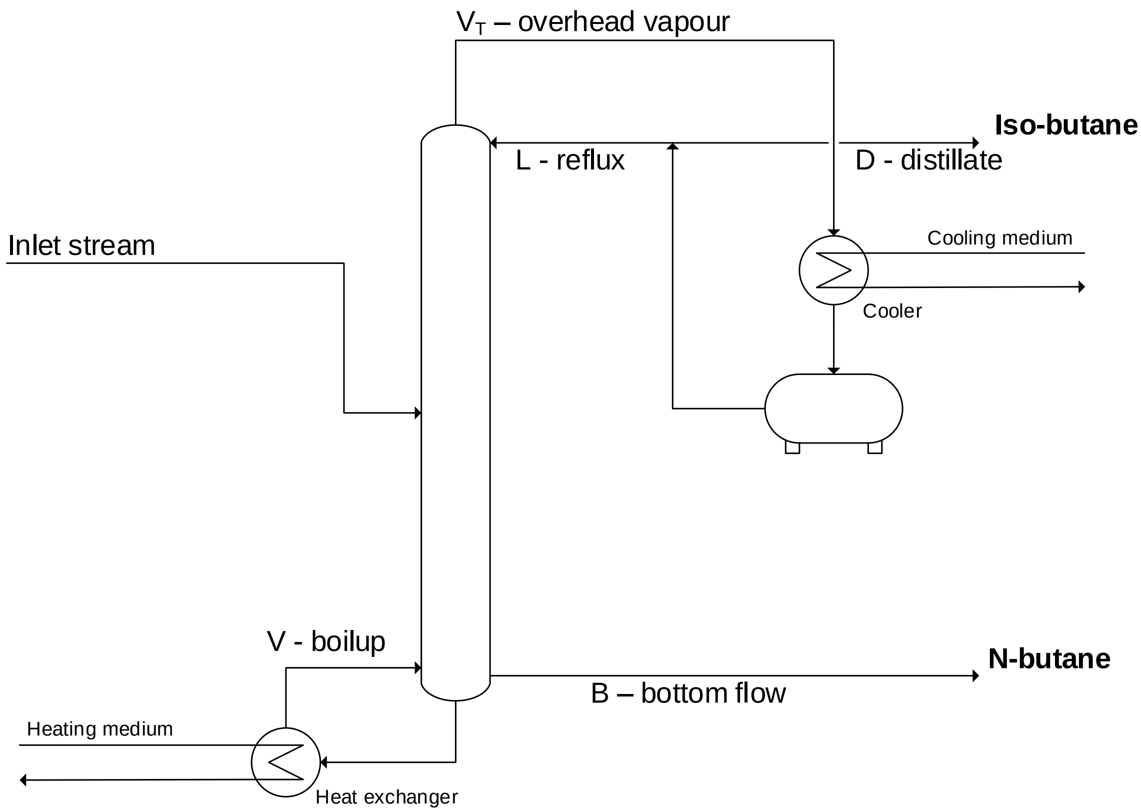

Figure 1: Simple overview of the butane splitter.

Introduction

A butane splitter separates a mixture of (mostly) **n-butane** and **iso-butane** into separate product streams. The separation is not perfect, and our task is to control the composition of both product streams.

For operation and composition control of ordinary distillation columns, there are five variables that need to be controlled:

- The pressure (p) of the distillation column
- The liquid level (M_D) in the top accumulator (reflux drum)
- The liquid level (M_B) in the bottom of the distillation column
- The composition (x_D) of the top product of the distillation column
- The composition (x_B) of the bottom product of the distillation column

For practical reasons, the composition control is often achieved indirectly through the control of temperature at selected locations in the top and bottom of the distillation column. This is the approach taken in the present project. It should also be noted that controlling both top and bottom composition often results in severe interactions between the top and bottom composition control loops (especially if high purity is desired in both products), and in some cases one therefore chooses to control only one of the compositions.

The available manipulated variables are:

- The vapour flowrate (V_T) leaving the top of the column
- The reflux flow (L) from the reflux drum back to the top of the column
- The top product flow (D - distillate) leaving the column
- The vapour flowrate (V) in the bottom of the column
- The bottom product flow (B) leaving the column

It is here assumed that the feed flowrate (F) is determined by upstream process units, and is not available as a manipulated variable. It should also be noted that V_T is often manipulated indirectly by manipulating the rate of condensation in the top of the column (especially for columns with a liquid product). Likewise, V is typically manipulated indirectly through manipulating the rate of heat transferred to the bottom of the column.

Three manipulated variables will need to be used to control pressure and liquid levels, leaving two manipulated variables available for composition control. The manipulated variables used for composition control then give name to the corresponding (composition) control structure.

In this project, **LV control structure** will be studied.

LV control

If V_T is used to control pressure (nearly always the case), D controls reflux drum level, and B controls the bottom level, one is left with L and V for composition control - where L controls the top composition and V controls the bottom composition. This is known as the **LV control structure**. The LV control structure may be the structure that is easiest to understand. The composition control is independent of level control. The level control may therefore be slow - allowing for slow changes in the product flowrates, and thus slow flowrate disturbances to downstream process units (if any). However, the composition control is known to be rather interactive, in particular if a similar bandwidth is desired in both composition control loops.

Secondary controllers

For the Kårstø column, several of the primary control loops operate in cascade, with secondary controllers in an inner loop. The secondary controllers are:

24_FC1005 Controlling the distillate flow D leaving the top of the column

24_FC1015 Controlling the reflux flow L back to the column top

24_FC1019 Controlling the bottom product flow B.

24_LC1028 Controlling the area for heat exchange in the bottom heat exchanger, indirectly controlling the bottom vapour flowrate V.

The tuning of these controllers will not depend much on which control structure is used for composition control. The same holds for the pressure controller 24_PC1024, since it is relatively fast compared to the composition control.

Product specification

The maximum allowed impurities in the products for the Kårstø Butane Splitter has been specified to:

- Max. 4% of **n-butane** in the top product (D - distillate)
- Max. 2.5% of **iso-butane** in the bottom product (B)

These purities can be achieved by keeping the temperatures at the 88th and 19th tray in the column to $35.85°C$ and $47.69°C$.

To ensure that these constraints are not violated due to disturbances and measurement noise, it is necessary to apply some ”back-off”, which means that the set-points are kept a certain distance away from their constraint values. Therefore, the set-points for the temperature controllers have been set to $35.30°C$ and $48.51°C$, which results in 2% of **n-butane** in the top product and 1.25% of **iso-butane** in the bottom product.

Data logging (from K-spice)

We have proposed a file `Logging.lst` which you can use for data logging in K-spice. After running the model (for example by using an MCL file, as described in the next section), you can choose `Export Trend Data` under the `Tools` tab. This opens up a dialog box, where you need to find the correct `lst` file and then press `Export`. The `lst` file contains a list of the parameters you want to log - and in the proposed file. We have written the variables you (probably) need to complete this assignment, some of which you will need for your system identification experiment. If you want to extend the list, this is possible.

In the `Spreadsheet` dialog box, you may also choose the sampling rate and time span for the logging. If you for example choose the first one, you will have a sample time of 1 second and a time span of 18000 seconds (5 hours).

The logging will generate a file `logging.lst.txt` in the same folder as the `lst` file and may be imported in Matlab, preferably using a script.

An example of a Matlab script `Import_Logging_lst.m` that imports the logged data into Matalb and the `Logging.lst` file can be found in the zip file `Log_Data_to_Matlab.zip` from Blackboard.

If the recorded variables do not appear in the log files, make sure that "History" is checked in the faceplate of the relevant controller.

Model Control Language (MCL) for running experiments

For running a specific experiment on the model, you may use MCL. It gives you the opportunity to specify exactly how the model should be run; for example you might set a controller in manual mode and to a specified output value to see how this affects the system.

Download the file `MCL.zip` from Blackboard. It contains an example MCL file and the MCL reference guide, so that you have the possibility to try this. It is voluntary.

You can run an MCL file on a K-Spice model by choosing `Tools` → `Run MCL...` or alternatively `MCL Manager...`

Choice of experiments

You may perform the experiments in the way you want, but we advice you to use MCL for running the experiments and LST files for logging. You should also think about how to choose the excitations. As an example, for level control you may perform the excitations in closed-loop, i.e. make steps in the reference, while for composition control you may set the controller in manual mode and make a step for the control signal instead of the reference. Do also think about what time-frame you should use for each excitation.

Remember to report in details **how** you performed the experiments - and **why** you performed them in the way you did.

System identification (in Matlab)

In this problem, you have two alternatives:

1. **MIMO system identification** For multivariable system identification, you may choose between the the free Matlab toolbox D-SR by Dr. David DiRuscio (Høgskolen i Telemark) or the algorithm `n4sid` in the Matlab System Identification Toolbox.

   To install the D-SR toolbox, download d-sr.zip from `http://www2.hit.no/tf/fag/sce2206/d-sr/d-sr_e.html` or from Blackboard, unzip the files to a folder and add this folder to the Matlab path. The identification is done by using `dsr`. To read more about the algorithm, type `doc dsr` in Matlab (if file is added to the Matlab path).

   The System Identification Toolbox is installed on the computers in G-116, G-118, and Sanntidssalen. The description and the schedule for these computer labs are available at `https://tp.uio.no/ntnu/timeplan/?type=room`; please check the schedule and go there when it's not occupied by other courses! To read more about the algorithm, type `doc n4sid` in Matlab.

2. **System identification based on SISO models** You may identify SISO (single input, single output) models, and put these together for the overall model. The resulting frequency responses should still be reasonable, and reasonable tuning of the individual loops should still be possible. The model will, however, not be very good for simulation of partially controlled systems (in particular, the responses in uncontrolled outputs will be misleading). Anyway, to use this method, download the file `SISO-SystemIdentificationRoutines.zip` from Blackboard, containing files based on lecture notes and solutions from the course `TTK4215 System identification and adaptive control`. You may use them as guidelines for SISO system identification in Matlab.

Approach to identification and control

It is desirable (possibly even necessary) to break the control design problem into smaller sub-problems. Some comments and recommendations:

1. The flow controllers can typically be tuned without considering effects on other loops, using your favourite tuning method. Try to use realistic tunings, bear in mind that the simulation is in continuous time (or tries to be), whereas in real life the controllers may be implemented in a DCS with a sampling time of (for example) 1 second.

2. The design of the level controllers is much simpler if the temperature controllers are put in manual. It is probably sufficient to approach the tuning of the level controllers also as SISO problems

3. Heat input in the bottom is manipulated through a level control in a flooded condenser (Note: this is a separate level control problem, and is different from what is referred to as level control in the bottom above). We probably want this level loop to be quite fast, to avoid any unnecessary bandwidth limitations in the temperature control in the bottom (which manipulates the heat input).

4. The temperature (composition) loops are expected to be interactive, in particular for the LV configuration. All other loops should be in automatic when identifying dynamics and designing controllers for the temperature control loops.

Problem description

1. **Installation**

   (a) Download `ButaneSplitter.zip` from Blackboard.

   (b) Unpack (extract) the `ButaneSplitter.zip` file into your K-Spice projects folder.

2. **Secondary controllers**
   Tune the secondary controllers (inner loops in cascades) and the pressure controller.

3. **Level controllers**

   (a) **System identification**
   Perform an identification experiment for each of the level controllers. The resulting model will include the inner loops in the cascades.
   The manipulated variables for level control will have to be excited through changing the setpoints for the level controllers.
   You will have to log experiment data using the proposed `Logging.lst` file and import the data in Matlab.

   (b) **Controller tuning**
   Tune the level controllers, based on the identified model (in Matlab).

4. **Composition controllers**

   (a) **System identification**
   Perform an identification experiment for the composition controllers. The resulting model will include the inner loops in the cascades.
   In order to perform the identification experiment, the level controllers will have to be in operation. The manipulated variables for composition control may be excited directly.
   You will have to log experiment data using the proposed `Logging.lst` file and import the data in Matlab.

   (b) **Controller tuning**
   Tune the main composition controllers, based on the identified model (in Matlab).

5. **Simulation**
   Verify the controller tunings using dynamic simulation - first in Matlab and then in K-spice.

If you have any problems with the K-Spice license server, please contact the teacher assistant.

Evaluation criteria

The project will be evaluated according to:

1. The resulting performance of the control system, prioritized in this order:
   (a) Overall stability of the system; any oscillations should be well damped.
   (b) Temperature controllers should be able to reach their setpoints, as long as they are within a feasible region, in order to control composition. In this context, a reference step of one degree is considered to be large step.
   (c) Performance of the temperature (composition) control. Both the speed of response (measured by, e.g., dominant closed loop time constants) and degree of interaction between loops are relevant for performance. Performance of other loops (levels, flows,...) is relevant to the extent that these loops affect achievable performance for the temperature control.
2. The documentation of the control performance and why this is good. That is, the report should:
   - Be clear and well structured, and be understandable without access to the assignment text
   - Be written with a reasonably competent control engineer as the intended audience (i.e., NOT for the professor who has seen this a hundred times before)
   - Discuss all results (e.g. explain whether and why you feel that the resulting control performance is acceptable/good).
   - Explain the approach/procedure used for identifying the required model(s) and designing/tuning the controllers.
3. The degree to which you have been able to use any of the theory/methodology covered in the course (desirable, but secondary to good control and good documentation).

Report

- When you deliver reports for these assignments, you do not need to attach any Simulink models or other Matlab files. We do not have time to run your programs. Instead, please paste all relevant models, code snippets and graphs into your report.
- When it comes to reporting your results for a specific problem, all graphs need to clearly define what signals are shown in the graphs, so that it is possible to analyze the results.
- All controller parameters need to be reported if relevant for the exercise.
- All results need to be reported/presented and discussed.
- You will have to deliver **individual reports**.
- Remember that this project counts for 30% of your grade in this course!