Abstract

The purpose of this project was to investigate strategies for dynamic back-off for control of active constraints. The distillation column model "Column A" was chosen as a case system to study. Optimization of the column operation and implementation of a control structure was performed before testing two variants of back-off control of the distillate purity (which is an active constraint variable). The idea was to apply P control to change the setpoint of the distillate purity x_D every time the disturbance reached its amplitude. This did not work out, but it turned out that for this very simple case it was possible to continuously impose P control to keep x_D backed off from its constraint value.

Acknowledgements

First of all, I would like to thank my co-supervisors, Post doc. Johannes Jäschke and PhD student Chriss Grimholt for always having their doors open and patiently helping me with the project throughout the semester. I would like to thank my supervisor, Professor Sigurd Skogestad, for the support and in particular for teaching the course Advanced Process Control, which proved to be a very useful supplement while working with the project.

Finally, I would like to thank my fellow students at K4-230 for keeping good spirit and supporting each other, especially throughout the last days and nights before the deadline.

Contents

1 Introduction

To determine the optimal operation point of a process, the general approach is to formulate a cost function and minimize it subject to certain constraints. Examples of constraints can be maximum allowable temperature in the reactor, minimum product purity and maximum flow rate. The optimal operating point typically occurs where one or more variables are at their constraints. These are called active constraints. If a process is to be operated optimally, a disturbance might lead to the violation of active constraints. Sometimes an active constraint also might be a hard constraint, which means that it never can be exceeded. An example of this might be the explosion temperature in a tank. To avoid breaking these constraints, it is necessary to apply back-off, which means that the setpoint is kept a certain distance away from the constraint value. This does also mean that you are moving away from the economically most beneficial operating point. A way of minimizing the loss due to back-off, is to adjust the setpoint dynamically instead of having a constant setpoint. This entails that you can impose a smaller back-off when the disturbance is low, and thus move the operating point closer to the active constraint.

1.1 Problem description

A distillation column was chosen as a case for investigating strategies for dynamic back-off control. The existing model Column A (developed by Skogestad et. al), which is modeled in Matlab and have Simulink interfaces, was used in this study.

The project consisted of the following tasks:

1. Optimize the operation of the column to obtain active constraints regions
(Reproduce the results of Jacobsen[3]/Leer[4])
2. Select a control structure and tune the controllers
3. Test strategies for dynamic back-off control of active constraints

2 Background

2.1 Optimization

The general form of the optimization problem is formulated as follows:

$$\min_{u} J(u, x, d) \tag{2.1}$$

subject to

$$f(x, u, d) = 0 \tag{2.2}$$

and

$$g(x, u, d) \leq 0 \tag{2.3}$$

Where J is the economical objective function, f the process model equations and g the process constraints. u are the degrees of freedom (manipulated variables) that can be adjusted to minimize J, while d are the expected disturbances.

Different computational tools can be applied to solve the optimization problem. One of them is the function $fmincon$ in Matlab, which solves constrained nonlinear optimization problems, starting at an initial estimate.

2.2 Distillation

Distillation is an important unit operation which is often used for studying process dynamics and control, as a distillation column is a system itself. A typical distillation column is shown in Figure 2.1. A binary mixture is fed to the column and separated into light product (top) and heavy product (bottom).

2.2.1 Column A

Column A is a nonlinear model of a continuous distillation column, studied in several papers by Skogestad and Morari. It separates a binary mixture into products of 99% purity.

The following assumptions underlies the model, as described by Skogestad in [6]:

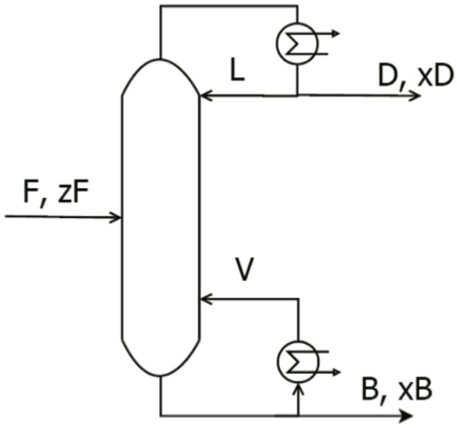

Figure 2.1: A conventional distillation column with one feed and two products, taken from Jacobsen[3]

1. Binary mixture
2. Constant pressure
3. Constant relative volatility
4. Equilibrium on all stages
5. Total condenser
6. Constant molar flows
7. No vapor holdup
8. Linearized liquid dynamics (but effect of vapor flow ("K2-effect") is included.)

2.3 Controller tuning

In this study, P and PI control will be applied. For tuning of PID controllers, Skogestad has developed a set of simple tuning rules, namely the SIMC (Simple/Skogestad Internal Model Control) tuning rules[9].

For a PI controller the steps will be like this:

Step 1: Obtain a first order + delay model (FOPDT) that approximates the process:

$$g(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} \tag{2.4}$$

Where $g(s)$ is the process transfer function, k is the plant gain, θ is the effective time delay and τ_1 is the dominant lag time constant (additional time to reach 63% of the response).

To obtain the FOPDT model, a step response experiment is performed: make a step change in the input u and plot the output y. Using this plot, the parameters can be obtained:
Steady-state gain:

$$k = \frac{\Delta y_\infty}{\Delta u} \tag{2.5}$$

Where Δy_∞ is the total change in the output, and Δu is the step in the input.
τ_1 is found as the time where the response reaches 63% of its final value:

$$y_{63\%} = y_0 + 0.63 \cdot \Delta y \tag{2.6}$$

Step 2: Obtain the controller settings for the PI controller:

$$c(s) = K_c \cdot (1 + \frac{1}{\tau_I s}) \tag{2.7}$$

Controller gain:

$$K_c = \frac{1}{k'} \frac{1}{(\theta + \tau_c)} \tag{2.8}$$

Integral time:

$$\tau_I = \min(\tau_1, 4(\tau_c + \theta)) \tag{2.9}$$

k' is the slope after response "takes off", and is calculated as follows:

$$k' = \frac{k}{\tau_1} \tag{2.10}$$

For an integrating process, k' is calculated directly from:

$$k' = \frac{\Delta y}{\Delta t \cdot \Delta u} \tag{2.11}$$

τ_c is a tuning parameter that needs to be selected. Skogestad[9] states that a small τc should be chosen for fast speed of response and good disturbance rejection, and a large τc for stability, robustness and small input usage. The first corresponds to tight control, the latter to smooth control.

2.4 Dynamic back-off

As described in the introduction, the back-off b is the difference between the setpoint and the constraint value:

$$b = |y_{constraint} - y_s| \tag{2.12}$$

This is shown visually in Figure 2.2. As back-off results in economical loss, it would be beneficial to at all times keep the back-off at a minimum instead of having a constant setpoint (as is the case in Figure 2.2). Dynamic back-off will require a controller that can adjust the setpoint of the output variable subject to the effect of disturbances.
The most direct approach will be to apply feedforward control where the disturbance is measured directly and used to give the new setpoint, or feedback control where the response in the output (subject to a disturbance) is measured and used to give the new setpoint.

The magnitude of the required back-off depends on whether the active constraint is soft or hard, as mentioned in the introduction. Soft and hard constraints can be described as following:

1. **Soft constraint:** A constraint that *can* be violated dynamically, which means

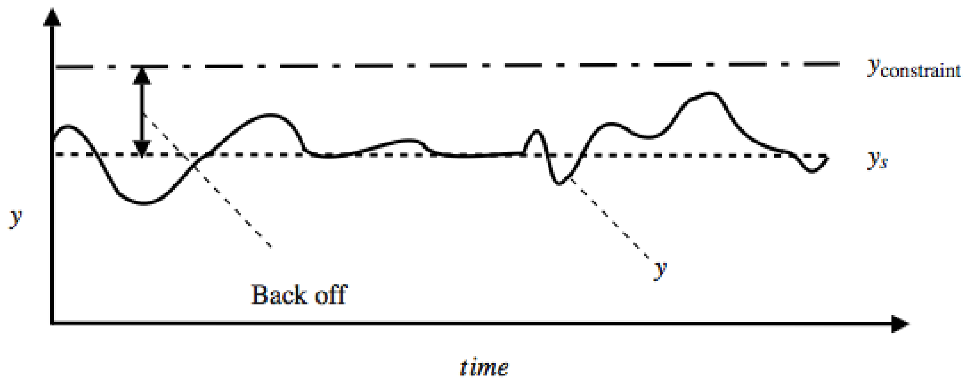

Figure 2.2: Illustration of back-off (Aske, 2003 [1])

that it can be violated as long as the average value of the output is satisfactory. Example: Product purity.

b = bias = the steady-state measurement error

2. Hard constraint: A constraint that *cannot* be violated dynamically, for example the explosion temperature in a tank.

b = bias + maximum dynamic control error

3 Optimization and control of the column

3.1 Column description

3.1.1 Operating conditions

The operating conditions are given in Table 3.1 and are the same as for the Case 1a studied by Leer[4]. The column has 41 stages (40 theoretical), where stage 20 represents the feed.

Table 3.1: Operating conditions for Column A

Variable	Value
α	1.5
z_F	0.5
F	variable 0-1.6 mol/s
p_F	1 \$/mol
p_B	1 \$/mol
p_D	2 \$/mol
p_V	0.01-0.02 \$/mol
$x_{B,max}$	0.010
$x_{D,min}$	0.950
V_{max}	4.008 mol/s

3.1.2 Degrees of freedom

A distillation column with a given feed and pressure will have four dynamical degrees of freedom, according to Skogestad, Lundström, Jacobsen (1990) [2]. The levels in the condenser and the reboiler need to be controlled dynamically, but have no steady-state effect. Thus, there are two degrees of freedom left to control the compositions of the distillate and the bottoms product, and Jacobsen[3] chose the vapor boilup V and the reflux L.

$$U = [L \;\; V] \tag{3.1}$$

3.1.3 Disturbances

For a distillation column, the feed conditions are important disturbances. In this case study the only disturbances are the feed flow rate F and the energy cost p_V.

$$d = [F \;\; p_V] \tag{3.2}$$

3.1.4 Constraints

There are three constraints:

1. The purity of the distillate has to be at least 99%
2. The fraction of light component in the bottom has to be equal to or smaller than 1%
3. The maximum boilup V_{max} cannot exceed 4.008 mol/s

3.2 Optimization

For a single distillation column with one feed stream and two products, no side streams and no heat integration, the cost function may be written[3]:

$$J(u,d) = p_F F + p_V V - p_D D - p_B B \tag{3.3}$$

Then the optimization problem becomes:

$$\min_u J(u,d) = p_F F + p_V V - p_D D - p_B B \tag{3.4}$$

subject to:

$$x_B \leq x_{B,max} \tag{3.5}$$

$$x_D \geq x_{D,min} \tag{3.6}$$

$$V \leq V_{max} \tag{3.7}$$

The optimization was carried out using the optimizer *fmincon* in Matlab. The scripts are attached in Appendix A.

The purpose of the optimization was to reproduce the active constraint regions map (Leer, 2012 [4]), shown in Figure 3.1. Even though an identical Matlab code was used, the values were not corresponding accurately, and x_D did not become active in all regions. Also when running scripts obtained from Minasidis[5] that should result in exactly the same values as Leer's, the results were not exactly the same.

The further work was therefore based on Leer's values. The optimization results for selected variables are presented in Table 3.2, for comparison with Leer's results and the values obtained with Minadisis' scripts.

Table 3.2: Selected optimal values, comparison with the results from Leer[4] and Minasidis[5]

	Haarsaker			Leer			Minasidis		
Region	I	II	III	I	II	III	I	II	III
$F[mol/s]$	1.2	0.7	1.4	1.2	0.7	1.4	1.2	0.7	1.4
$p_V[\$/mol]$	0.012	0.018	0.002	0.012	0.018	0.002	0.012	0.018	0.002
$L[mol/s]$	2.6712	1.6093	3.2760	2.7364	1.3275	3.2760	2.7337	1.5520	3.2751
$V[mol/s]$	3.2970	1.9689	4.0080	3.3631	1.6402	4.008	3.3604	1.9169	4.0071
$D[mol/s]$	0.6258	0.3596	0.7320	0.6267	0.3128	0.7320	0.6267	0.3649	0.7320
$B[mol/s]$	0.5742	0.3404	0.6680	0.5733	0.2872	0.6680	0.5733	0.3351	0.6680
x_D	0.9500	0.9638	0.9500	0.9500	0.9500	0.9500	0.9500	0.9500	0.9500
x_B	0.0096	0.0100	0.0068	0.0088	0.0100	0.0069	0.0081	0.0098	0.0068
$J[\$/s]$	-0.5862	-0.3242	-0.7240	-	-	-	-0.5863	-0.3304	-0.7240

As Figure 3.1 shows, the mole fraction of the heavy component in the distillate is always at its constraint value, i.e. it is always active. In region II, also the mole

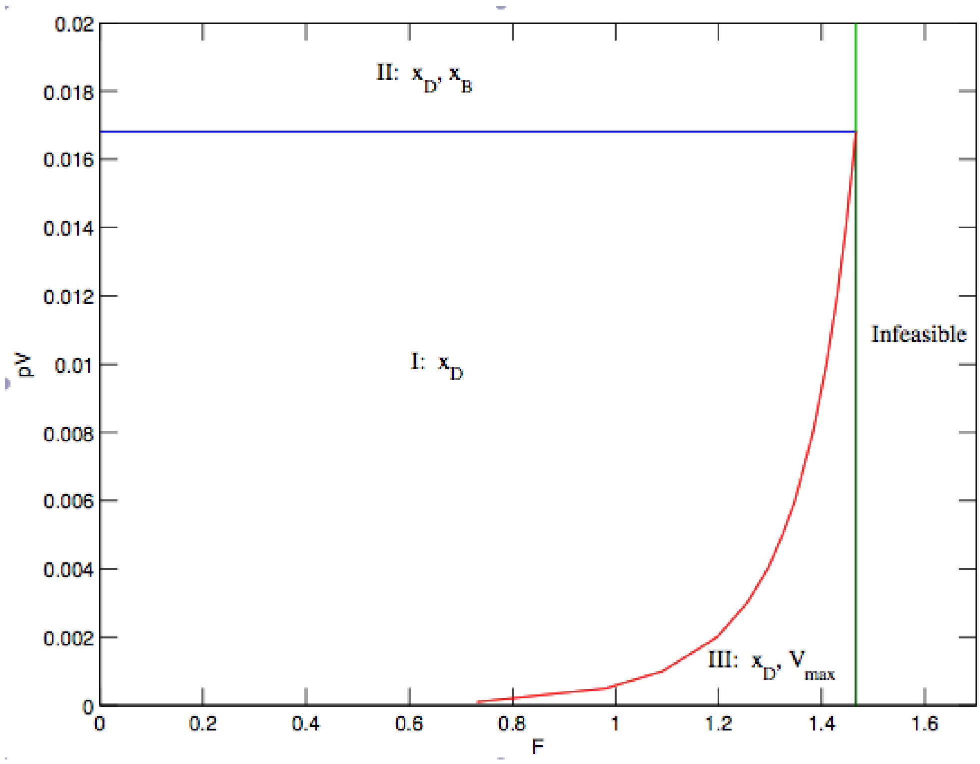

Figure 3.1: Active constraints regions, as found by Leer (2012)[4]

fraction of heavy component in the bottom becomes active. V_{max} becomes active in region III, and when F exceeds 1.48 the operation becomes infeasible.
Region III is selected for this study, as keeping both x_B and x_D at their constraint values eliminates the need of finding a self-optimizing variable.

3.3 Control structure

According to [6], a particular way of stabilizing the column is the LV configuration, where the distillate flow D is used to control the condenser holdup M_D and the bottoms flow B to control the reboiler holdup M_B. The existing Simulink model *colas_lv_nonlin* has P controllers with gain -10, but no composition loops closed. Then L is left to control x_D and V to control x_B.

A couple of small modifications were made to the Simulink model before starting the work with the control structure, as can be seen in Appendix B.

Because the purity of the distillate usually is the most critical issue, the distillate composition controller will be tuned before the bottoms composition controller, applying PI control in both cases.

3.4 Generating steady-state values

Since the column initially was configured to yield $x_B = 0.01$ and $x_D = 0.99$, new steady-state values have to be generated for the case with $x_B = 0.01$ and $x_D = 0.95$. This is done by applying and tuning the two PI controllers, saving the steady-state values and loading the new values as initial values. Then the controllers are tuned again. The new steady-state data is attached in Appendix C, and the new setpoints for M_B, M_D, L and V are shown in Table 3.3:

Table 3.3: Setpoints for key variables for different distillate purity.

	$x_D = 0.99$	$x_D = 0.95$
L_{nom}	2.7063	2.2125
V_{nom}	3.2063	2.7337
$M_{B\,nom}$	0.5000	0.5962
$M_{D\,nom}$	0.5000	0.6048

3.5 Tuning of composition controllers

The controllers were tuned according to the SIMC tuning rules described in Section 2.3. See Appendix D for plots and calculations.

Distillate composition controller: $K_c = 51$, $\tau_I = 4$

Bottoms composition controller: $K_c = -35$, $\tau_I = 16$

4 Back-off strategies

After finding the active constraints regions, and implementing and tuning controllers, back-off strategies can be investigated. In order to stay i Region II where both x_D and x_B are at their constraint values, p_V is kept constant at 0.18 \$/mol. The only disturbance will then be the feed rate F, which must never exceed about 1.4 mol/s in order to not make the operation infeasible.
To investigate back-off strategies, a disturbance in the feed has to be created.

4.1 Disturbance

A sinusoidal disturbance was constructed and imposed on the nominal feed rate of 1 mol/s. The disturbance has a constant frequency, but sequences of varying amplitude. The disturbed feed is plotted in Figure 4.1, while the corresponding response in x_D is shown in Figure 4.2. The amplitudes of F and x_D for each sequence are listed in Table 4.1. For x_D there were some variations in the amplitude within each sequence, so the largest amplitude during each section is listed.

Table 4.1: Amplitude of the disturbed feed and the corresponding response in x_D

Sequence	Feed amplitude	x_D amplitude (max)
1	1	0
2	0.20	0.0010
3	0.30	0.0014
4	0.15	0.0007
5	0.03	0.0002
6	1	0

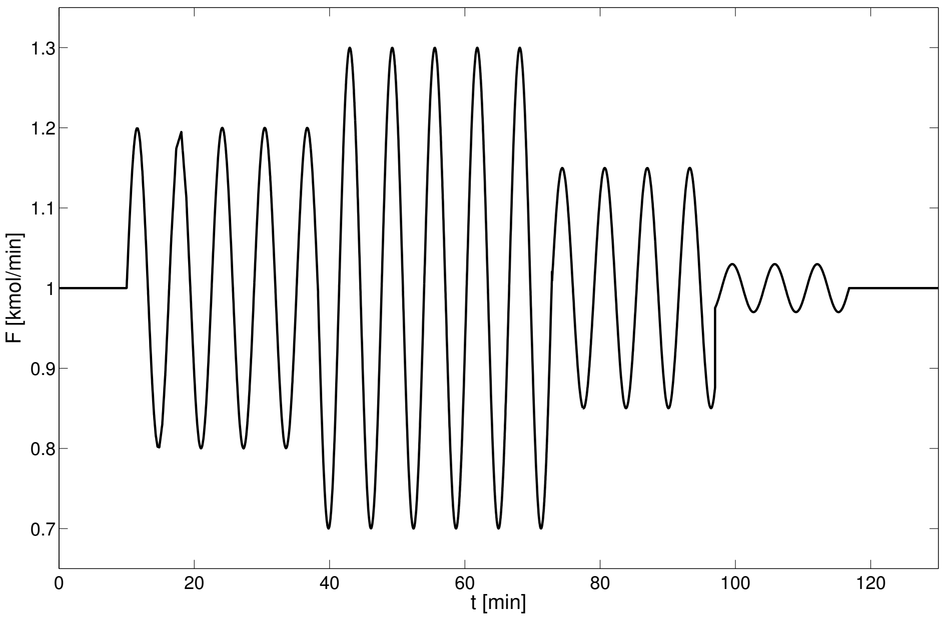

Figure 4.1: Sinusoidally disturbed feed rate, where F_{nom}=1.0 kmol/s

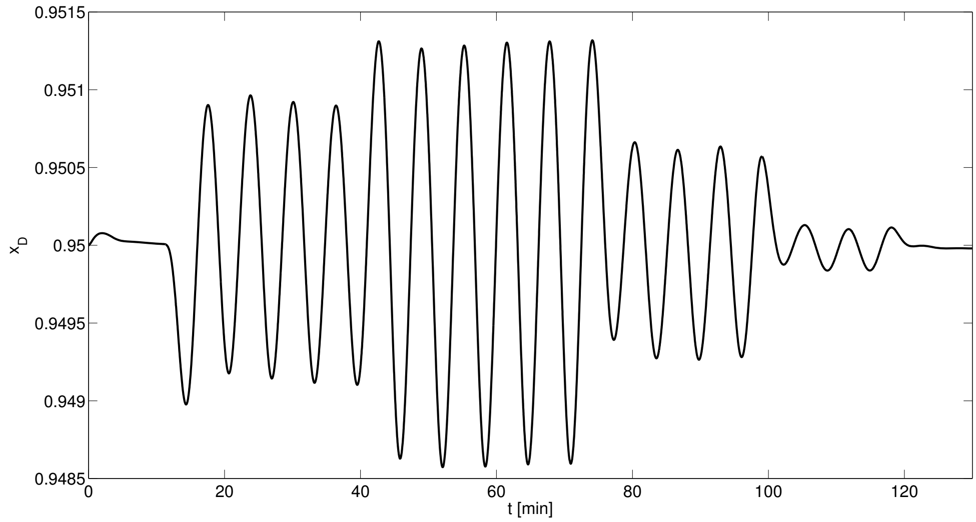

Figure 4.2: The response in x_D when subject to the disturbed feed shown in Figure 4.1

4.2 Back-off controller

A feedforward control strategy for the back-off control is investigated. The idea is to continuously monitor the disturbed feed rate, and whenever it reaches its amplitude, adjust the setpoint for x_D depending on the magnitude of the amplitude.

$$\text{When} \qquad \frac{dF}{dt} = 0 \tag{4.1}$$

$$\text{Calculate} \qquad A = |F_{nom} - F| \tag{4.2}$$

The new setpoint for x_D is then calculated from:

$$x_{D,sp} = x_{D,sp}^{nom} + (constant \cdot A) \tag{4.3}$$

This is equivalent to implementing a P controller with A as the gain.

Two cases are studied; the first one with the back-off controller always operating, and the second one with the back-off controller only operating when the disturbance was larger than a threshold.

4.3 Case study I: Back-off controller always operating

The idea of the first case study is to keep the setpoint at the constraint value and always keep the back-off controller operating. This means that the setpoint always will be larger than the constraint value, unless there is no disturbance at all.
Figure 4.3 shows a simplified Simulink block diagram with emphasis on the back-off control structure. The complete block diagrams are attached in Appendix E.

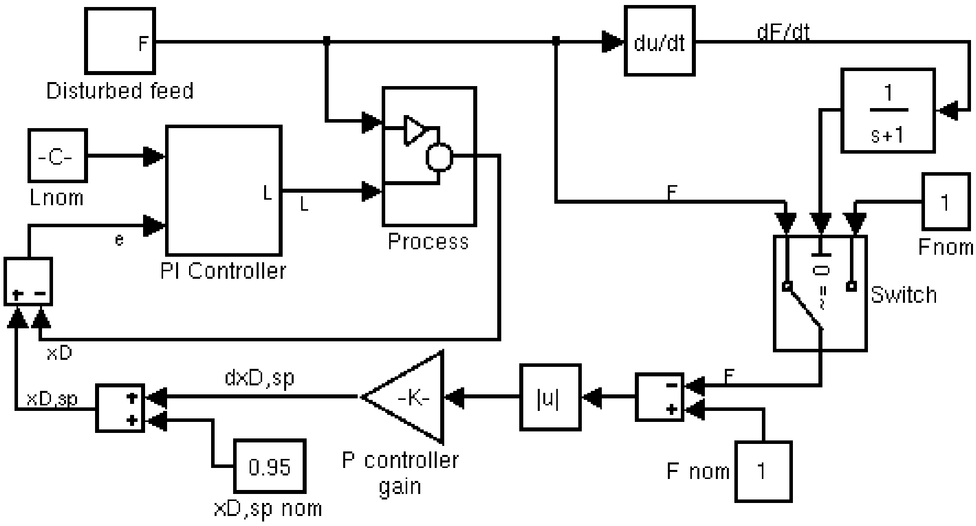

Figure 4.3: Back-off control implementation in Simulink

4.4 Case study II: Implying back-off only when the disturbance is sufficiently large

The idea of the second case study is to keep the nominal setpoint a little higher than the constraint value, such that there will be no need for back-off when the disturbance is small. Hence, the back-off controller will only be operating when the amplitude exceeds a threshold.
For this case study, the threshold is set to 0.29, which is slightly lower than the amplitude of the third and largest feed sequence. This will preferably make the back-off controller operate only during this feed sequence.

The nominal setpoint for x_D has to be set such that none of the other feed sequences will make x_D violate its constraint value. As seen from Table 4.1, the second largest feed sequence leads to a maximum amplitude of 0.001 in x_D, such that the nominal setpoint for x_D must be increased to $0.95 + 0.001 = 0.951$.
The control structure for Case II is the same as for Case I, with the only difference that $x_{D,nom}$ is set to 0.951 instead of 0.95, and that a switch that only passes the feed through to the controller when the amplitude is above a certain threshold, is added.

5 Results

5.1 Back-off controller

The idea of adjusting the setpoint only when F reaches its maximum did not work out. The switch that was applied to send F as an input to the P controller only when the derivative was equal to zero, did never pass anything through. However, by removing the switch such that P control always is imposed, the controller proved to be able to change the setpoint of x_D such that x_D never violated its constraint value. This is described further in Case I.

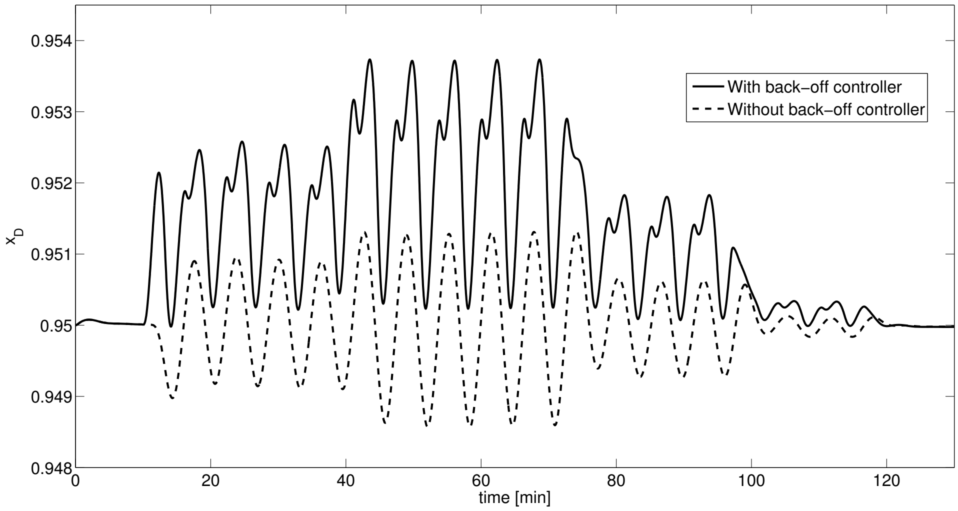

Figure 5.1: Response in x_D with the back-off controller from Case I (solid line) compared to response with no back-off controller (dashed line)

5.2 Case I

The minimum gain of the P controller that was required to never break the constraint value for x_D was found by trial and error to be 0.012.
Hence, the setpoint for x_D is calculated as follows:

$$x_{D,sp} = 0.95 + 0.012A \tag{5.1}$$

The setpoint change with the Case I P controller is tracked in Figure 5.2 together with ideal setpoint change for Case I. The response in x_D subject to the disturbance is shown in Figure 5.3 together with the response if the setpoint was changing ideally. Finally, the cost function is plotted in Figure **??**
The second plot shows that the back-off controller - in this specific case - is able to change the setpoint of x_D such that it never drops below 95% purity.

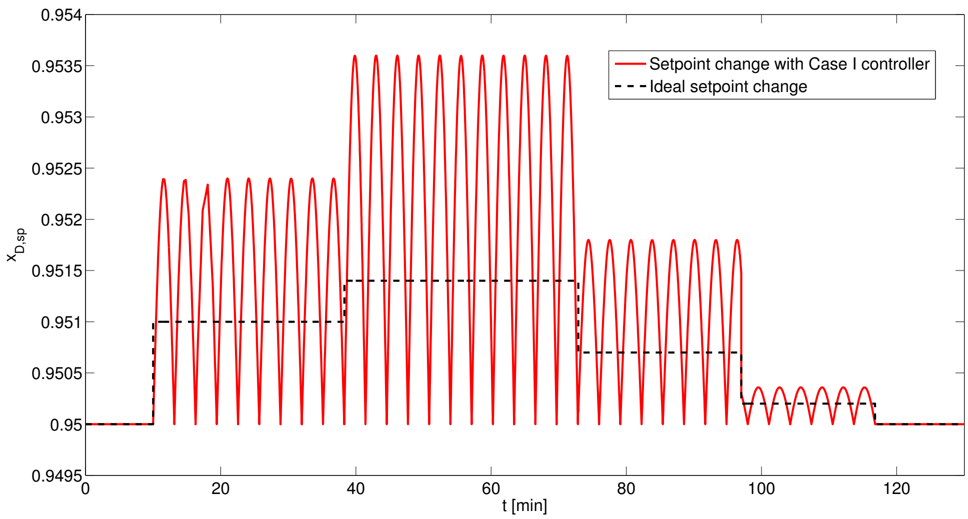

Figure 5.2: Change in the setpoint for x_D with the Case I controller vs. ideal setpoint change for Case I (dashed line)

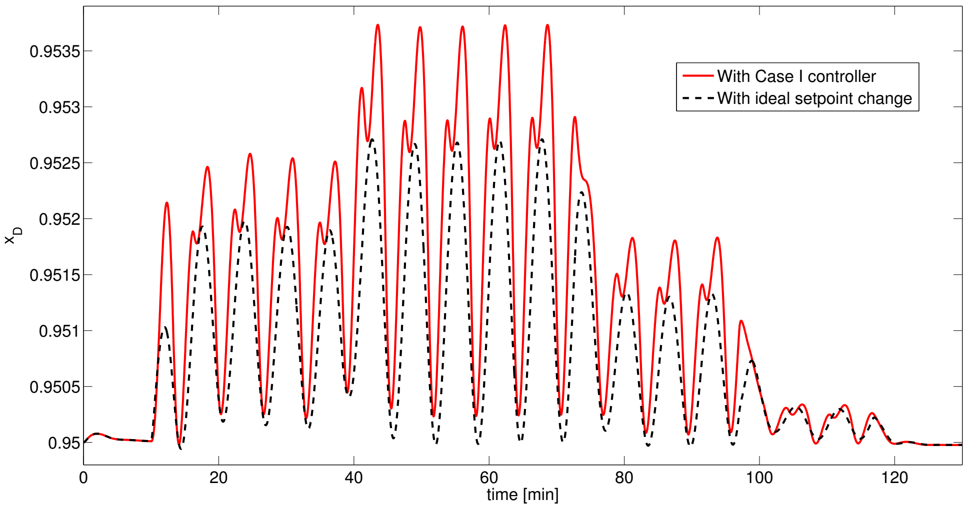

Figure 5.3: Response in x_D (solid line) compared to response with ideal setpoint change for Case I(dashed line)

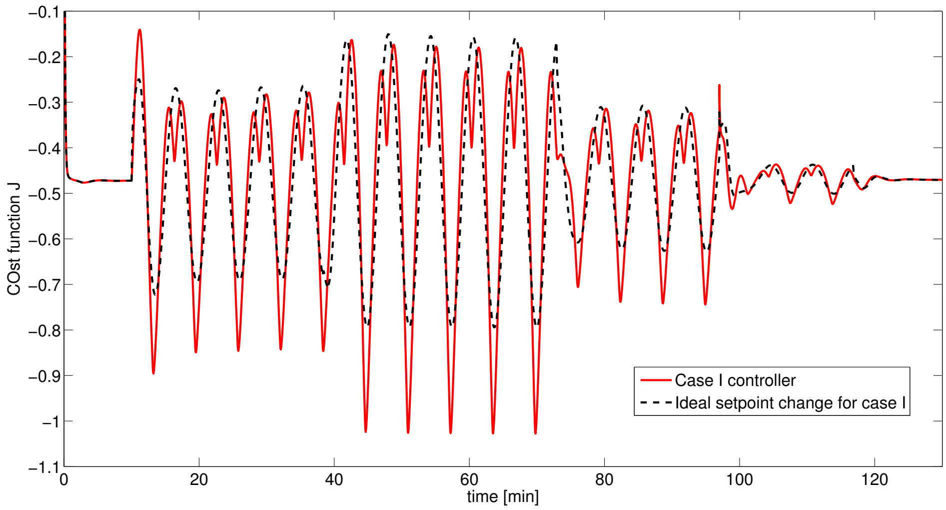

Figure 5.4: Cost function

5.3 Case II

With the threshold for the P controller set to A = 0.29, and $x_{D,sp} = 0.951$, the gain for the P controller was found by trial and error to be 0.038.
The setpoint for x_D is then calculated from:

$$x_{D,sp} = 0.951 + 0.038A \qquad \text{where A} > 0.29 \tag{5.2}$$

The setpoint change with the Case I P controller is tracked in Figure 5.5 together with ideal setpoint change for Case II. The response in x_D subject to the disturbance is shown in Figure 5.6 together with the response if the setpoint was changing ideally. Finally, the cost function is plotted in Figure 5.7.

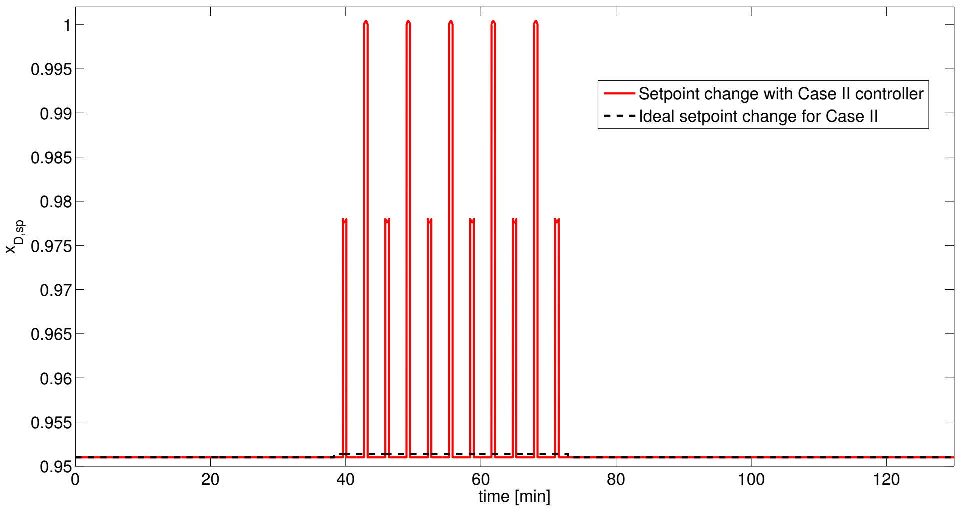

Figure 5.5: Change in the setpoint for x_D with the Case II controller vs. ideal setpoint change for Case II (dashed line)

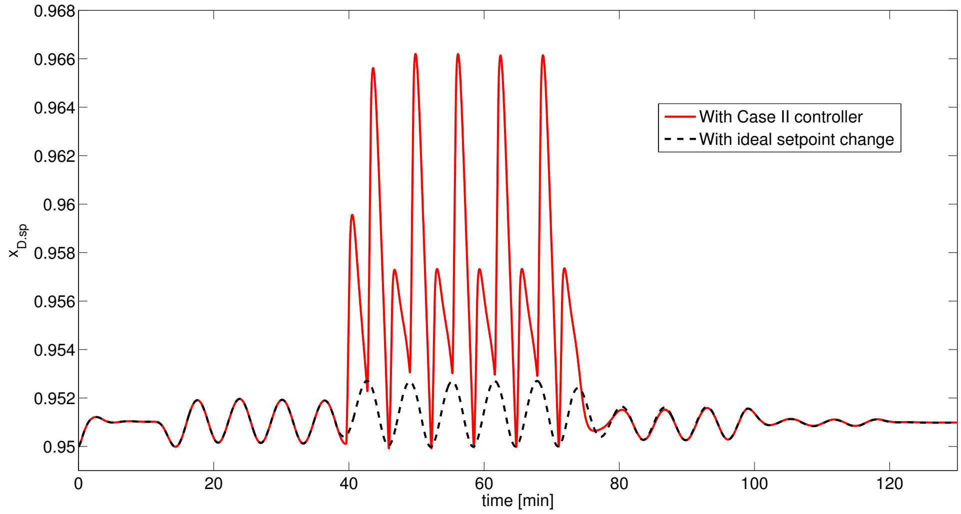

Figure 5.6: Response in x_D (solid line) compared to response with ideal setpoint change for Case II (dashed line)

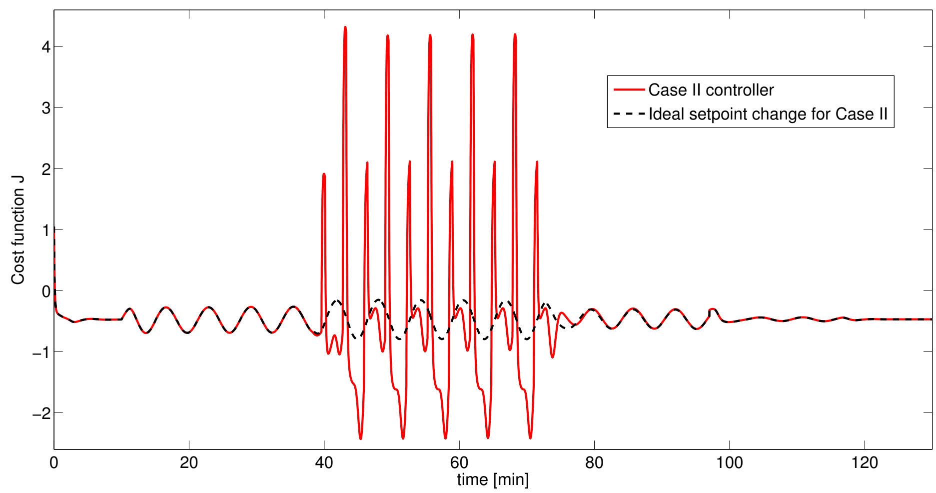

Figure 5.7: Cost function

5.4 Comparing the case studies

The responses in x_D in Case I and Case II are plotted together in Figure 5.8, while the cost function J for both cases is plotted in Figure 5.9. Table 5.1 shows the average value of x_D and J for both cases, together with the corresponding values for ideal setpoint change in each case.

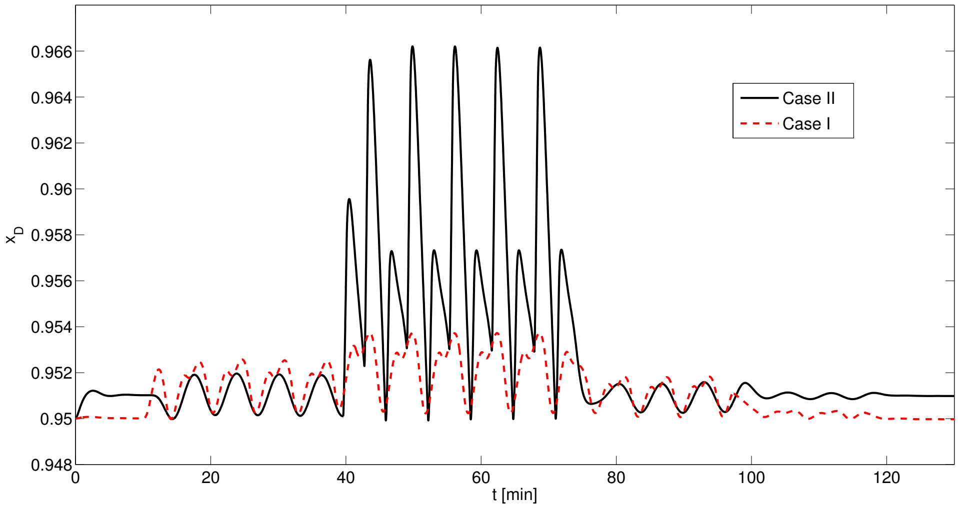

Figure 5.8: Response in x_D in Case I (dashed line) and Case II (solid line).

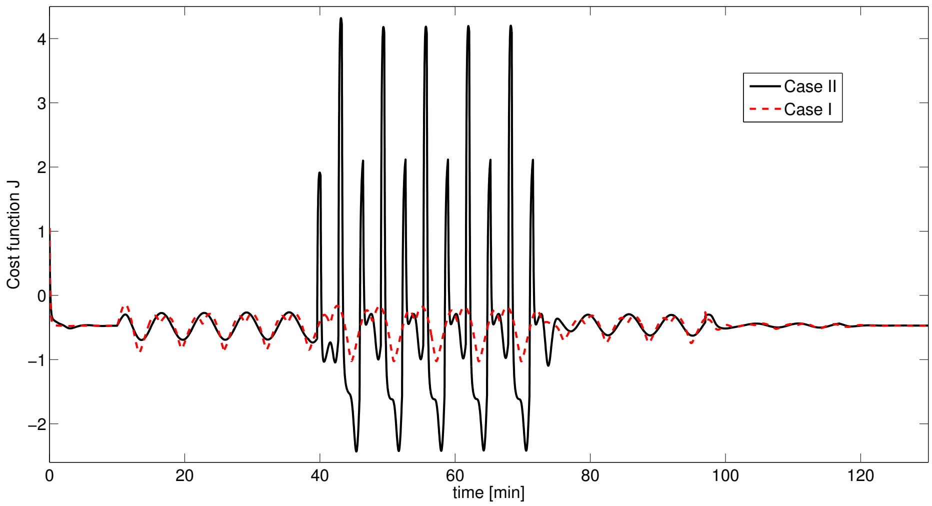

Figure 5.9: Cost function J for Case I (dashed line) and Case II (solid line).

Table 5.1: Average values of x_D and J over the simulation time interval

	Case I		Case II	
	w/SP controller	w/ideal SP change	w/SP controller	w/ideal SP change
$x_{D,sp}^{av}$	0.9513	0.9508	0.9537	0.9511
J^{av}	-0.4578	-0.4784	-0.1951	-0.4622

6 Discussion

Even though the idea of applying P control only when F reaches its maximum did not work out, for this specific case it was shown that a continuously working back-off P controller was sufficient to adjust the setpoint of x_D such that it never violated the constraint value. However, this might be considered a "toy case", where the disturbance was created such that P control actually was sufficient all the time. For a more realistic feed, simple P control will probably not be sufficient. For example, if there is a positive step in the feed, x_D is immediately brought far below its constraint value.

The controller in Case II lead to the least profitable operation, as might be expected. However, it is usually necessary to include a "safety margin" (such as here with a nominally higher setpoint) when controlling hard constraints, so this might be a more realistic case.

7 Conclusion and further work

From this study there are not many conclusions that can be drawn, except from the fact that the amplitude strategy did not work out, and that for this certain disturbance sequence a simple P controller was sufficient to keep x_D dynamically backed off from its constraint value. So for disturbances that show some degree of regularity, it might be possible to apply a P controller based on experience to keep the controlled variable dynamically backed off from the constraint.

However, dynamic back-off is a difficult subject to address, and the work with this project have been essential for gaining insight on the topic and has laid the foundation for the work I will carry out in the master thesis in the spring semester of 2013. It is possible that a pH neutralization process will studied instead of a distillation column, but the task will independent of the system be to further investigate strategies for dynamic back-off.

For example, for the disturbance in this case, it might be possible to monitor the amplitude of the feed, and whenever a *changed* amplitude is detected, use this to give a new setpoint, and keep this setpoint constant until the amplitude changes sufficiently.

Another approach might be to try to implement a feedback strategy, or do do a more statistical approach; such as counting incidents when x_D is below a warning limit and change the setpoint when there has been a certain number of subsequent incidents.

The disturbance that was constructed for this project was very simple, and in the further work more realistic disturbances should be considered.

References

[1] Aske, E., 2009: Design of plantwide control systems with focus on maximizing throughput. Ph.D. thesis, NTNU. only back-off illustration?

[2] Jacobsen, E., Lundström, P., Skogestad, S., 2: 1990: Selecting the Best Distillation Control Configuration, AiChe Journal

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

[4] Leer, R., 2012: Self-Optimizing Control Structures for Active Constraint Regions of a Sequence of Distillation Columns, Master Thesis at Dept. of Chemical Engineering, NTNU.

[5] Minasidis, V., Matlab files: *nonlcon.m* and *ObjFun.m*

[6] Skogestad, S., documentation on Matlab distillation column model ("Column A"): $http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html$

[7] Skogestad, S., Dynamics and control of distillation columns - a tutorial introduction. Dept. of Chemical Engineering, NTNU.

[8] Skogestad, S., 1992: Dynamics and control of distillation columns - a critical survey. In: IFAC-symposium DYCORD+'92, Maryland, Apr. 27-29, 1992

[9] Skogestad, S., 2001: Probably the best simple PID tuning rules in the world. Submitted to Journal of Process Control July 3, 2011. This version: September 12, 2001.

[10] Skogestad, S., 2007: The do's and don'ts of distillation columns. In: Trans IChemE, Part A, January 2007.

Appendix A Matlab codes used for optimization

A.1 Main script

The fmincon routine was used to optimize the operation of Column A:

```
clear all
clc

% x = [X U]', where X = [mole fractions (1 to 41) ; holdups (42 to 82)]
% U = [LT VB D B F zF qF]

%Disturbances: p_v, F, zF and qF

p_v = 0.002;     % 0.01 - 0.02 mol/s

prices = [p_v, 1, 1, 2];

F = 1.4 ;       % 1.0 - 1.6 mol/s
zF = 0.5;
qF = 1;

d = [F zF qF]';

%Inequality constraints (upper/lower bounds):
xB_max = 0.01;          % (1) x_heavy >= 0.99 --> x_light <= 0.01. xB  = X(1)
xD_min = 0.95;          % (2) x_light >= 0.95. xD = X(41)
Vmax = 4.008;           % (3)

%Upper and lower bounds:
lb = zeros(89,1);      % cannot have negative values
lb(41) = xD_min;        % Ineq. constraint 1. xD = X(41)
lb(83) = 0.1;
lb(84) = 0.1;

ub = ones(89,1);        %no fractions can be larger than 1.

ub(1) = xB_max;         % Ineq. constraint 2. xB = X(1)

i=42:82;
ub(i) = inf;            %maximum holdups

%i=83:89;
%ub(i) = 10;             %upper bounds for the U's

%i=84;
%ub(i) = Vmax;           % Ineq. constraint 3: vapor streams cannot be larger than Vmax

ub(83) = Vmax;           % Ineq. constraint 3: vapor streams cannot be larger than Vmax
```

```
ub(84) = Vmax;
ub(85) = F;
ub(86) = F;
ub(87:89) = 10;


%Controller:
par.KcB = 10;
par.KcD = 10;
par.MDs = 0.5;
par.MBs = 0.5;
par.Ds = 0.5;
par.Bs = 0.5;

%initial guess
x0 = 0.5*ones(89,1);
x0(1:41) = linspace(0.01,0.95,41);
x0(87)=F; x0(88)=zF; x0(89)=qF;

options = optimset('TolCon', 1e-8,'TolFun', 1e-8, 'TolX', 1e-8,'Algorithm','active-set');
[x,fval,Eflag] = fmincon(@(x)objfunc(x,prices),x0,[],[],[],[],lb,ub,@(x)nonlinconstr(x,d,par),
options);
Eflag


disp('Fraction of light component on each stage:')
disp(x(1:41))

disp('Holdups:')
disp(x(42:82))


disp('LT')
disp(x(83))

disp('VB')
disp(x(84))

disp('D')
disp(x(85))

disp('B')
disp(x(86))

disp('xB')
disp(1-x(1))

disp('xD')
disp(x(41))
```

```
disp('J')
disp(fval)

disp('Eflag')
disp(Eflag)
```

A.2 Nonlinear constraints

```
function [c, ceq] = nonlinconstr(x,d,par)

NT=41;

X=x(1:2*NT);             % i = 1:82
U = x(2*NT+1:end);       % i = 83:89, U = [LT VB D B F zF qF]
MB = X(NT+1);            % MB = X(42)
MD = X(2*NT);            % MD = X(82)
LT = U(1);
VB = U(2);
D = U(3);
B = U(4);

%Controller:
KcB = par.KcB;
KcD = par.KcD;
MDs = par.MDs;
MBs = par.MBs;
Ds = par.Ds;
Bs = par.Bs;

ceq = [colamodoriginal(0,X,U) ; U(5:7)-d ; D-KcD*MD-(Ds-MDs*KcD) ;  B-KcB*MB-(Bs-MBs*KcB)];
%D-KcD*MD-(Ds-MDs*KcD) ;  B-KcB*MB-(Bs-MBs*KcB)
%MB-0.5 ;  MD-0.5
%U(5:7)-d

c = [];


end
```

A.3 Object function

```
function f = objfunc(x,prices)            %takes in x = [X U]' and vector with prices

%Conversion of the state vector variables to standard notation:
```

```
d = x(87:89);           % d = [F zF qF]'
F = d(1);

U = x(83:89);           % U = [LT VB D B F zF qF]
VB = U(2);
D = U(3);
B = U(4);

%Prices:
p_v = prices(1);
p_f = prices(2);
p_b = prices(3);
p_d = prices(4);


%Cost function:
f = p_f*F + p_v*VB - p_b*B - p_d*D;
```

Appendix B Modifications of Simulink interface

1. Changed ODE-solver from ode45s to ode15s.

2. Changed the demux-block from [1,1,1,1,41] to [1,1,1,1,82]

3. Modification 1 of colas.m:

Changed

```
sys(5:NT + 4,1) = x(1:NT);
```

to

```
sys(5:2*NT + 4,1) = x;
```

4. Modification 2 of colas.m:
Changed

```
sys = [2*NT, 0, NT+4, 7, 0, 0];
```

to

```
sys = [2*NT, 0, 2*NT+4, 7, 0, 0];
```

Appendix C Steady-state data for Column A

How to load the new steady-state data:

```
Xinit = Comp(end,:)';
load('cola_init.mat');
save('cola_init.mat','Xinit');
Xinit=xinit;
save('cola_init.mat','Xinit');
```

Table C.1: Xinit: Initial values for Column A with x_D = 0.95 instead of x_D=0.99

Fractions of heavy component, x(i)				Molar holdups, M(i)			
Variable	Value	Variable	Value	Variable	Value	Variable	Value
x(1)	0.010000	x(21)	0.448739	M(1)	0.596224	M(21)	0.468890
x(2)	0.014191	x(22)	0.455460	M(2)	0.468890	M(22)	0.468890
x(3)	0.019477	x(23)	0.463746	M(3)	0.468890	M(23)	0.468890
x(4)	0.026113	x(24)	0.473901	M(4)	0.468890	M(24)	0.468890
x(5)	0.034393	x(25)	0.486252	M(5)	0.468890	M(25)	0.468890
x(6)	0.044649	x(26)	0.501139	M(6)	0.468890	M(26)	0.468890
x(7)	0.057239	x(27)	0.518888	M(7)	0.468890	M(27)	0.468890
x(8)	0.072521	x(28)	0.539773	M(8)	0.468890	M(28)	0.468890
x(9)	0.090822	x(29)	0.563977	M(9)	0.468890	M(29)	0.468890
x(10)	0.112386	x(30)	0.591532	M(10)	0.468890	M(30)	0.468890
x(11)	0.137315	x(31)	0.622276	M(11)	0.468890	M(31)	0.468890
x(12)	0.165508	x(32)	0.655816	M(12)	0.468890	M(32)	0.468890
x(13)	0.196609	x(33)	0.691519	M(13)	0.468890	M(33)	0.468890
x(14)	0.229993	x(34)	0.728549	M(14)	0.468890	M(34)	0.468890
x(15)	0.264791	x(35)	0.765931	M(15)	0.468890	M(35)	0.468890
x(16)	0.299970	x(36)	0.802651	M(16)	0.468890	M(36)	0.468890
x(17)	0.334454	x(37)	0.837770	M(17)	0.468890	M(37)	0.468890
x(18)	0.367246	x(38)	0.870505	M(18)	0.468890	M(38)	0.468890
x(19)	0.397544	x(39)	0.900299	M(19)	0.468890	M(39)	0.468890
x(20)	0.424801	x(40)	0.926829	M(20)	0.468890	M(40)	0.468890
		x(41)	0.950000			M(41)	0.604776

Appendix D Tuning of controllers

D.1 Tuning of distillate composition controller

Making a 3% step change in the input L and plotting the response of the output x_D:

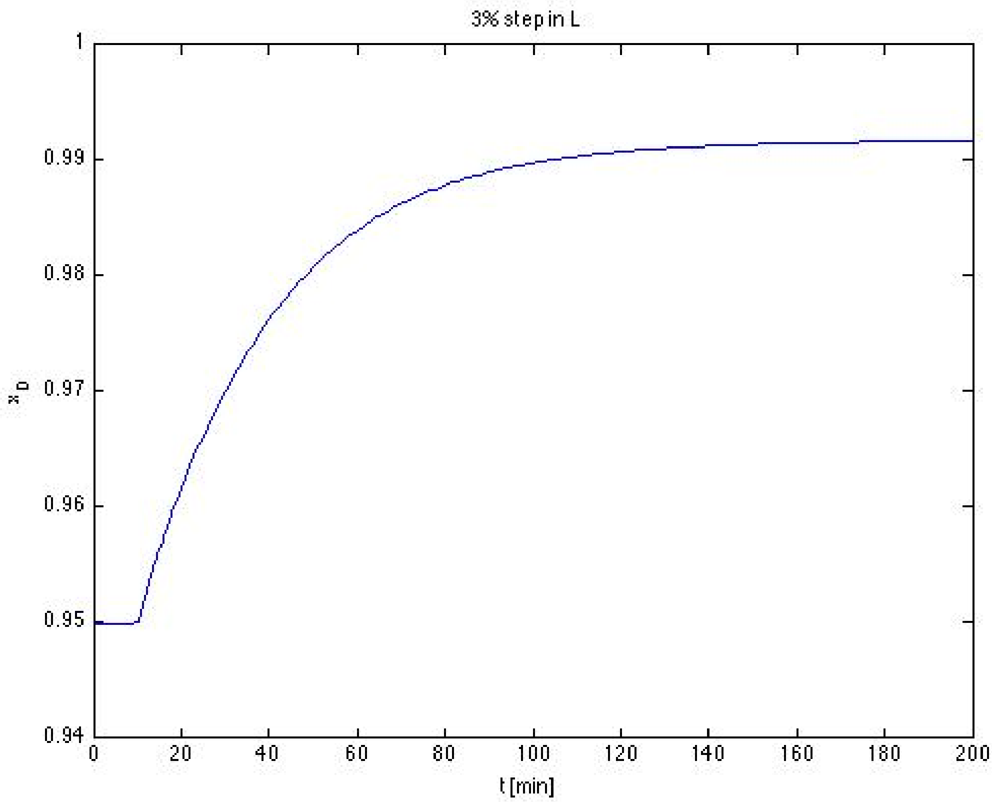

Figure D.1: Controller 1, 2nd tuning

Calculating the tuning parameters:

$$k = \frac{0.9916 - 0.9500}{0.03 \cdot 2.234544} = \frac{0.0416}{0.067066} = 0.018546 \tag{D.1}$$

63% of response:

$$y_{63\%} = y_0 + 0.63 \cdot \Delta y = 0.9500 + 0.63 \cdot 0.018546 = 0.9762 \tag{D.2}$$

This corresponds to $\tau_1 = (40.31 - 10)$ min $= 30.31$ min

$$k' = \frac{0.598813}{30.31} = 0.01976 \tag{D.3}$$

Setting $\tau_c = 1$:

$$K_c = \frac{1}{0.01976} \frac{1}{(0+1)} = 50.62 \tag{D.4}$$

$$\tau_I = min(30.31, 4\tau_c) = 4 \tag{D.5}$$

Closing the loop and making a step in the setpoint for x_D from 0.95 to 0.97 to test the performance of the controller:

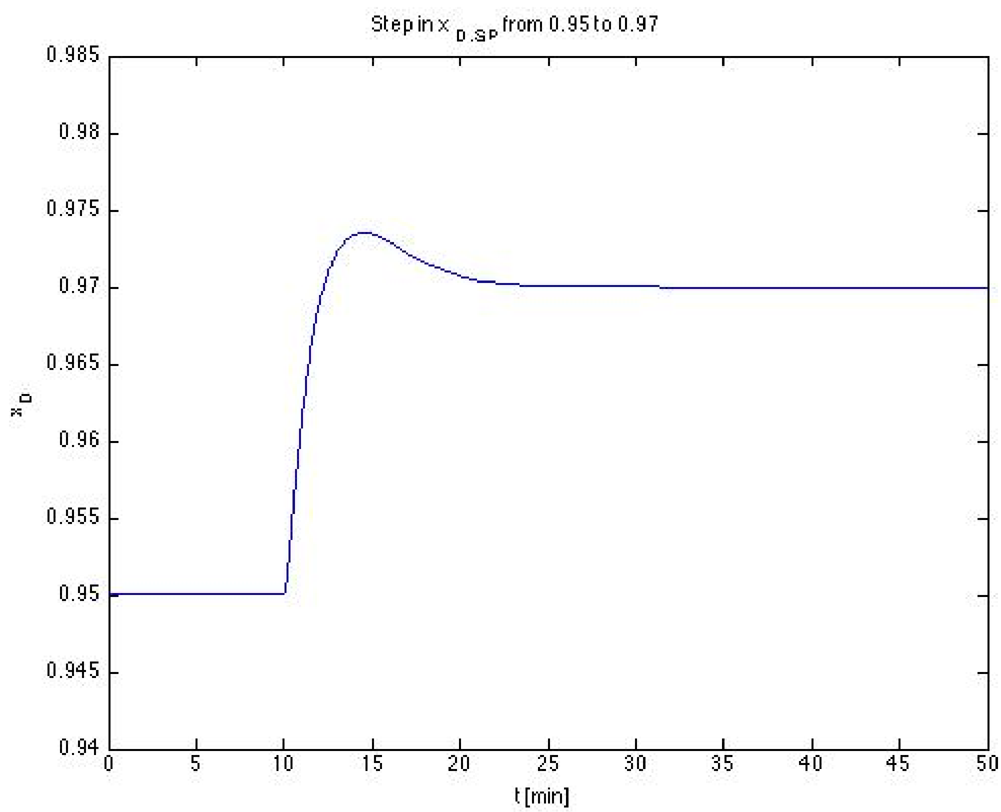

Figure D.2: Controller 1, 2nd tuning, sp change

D.2 Tuning of bottoms composition controller

Making a -3% step change in the input V and plotting the response of the output x_B:

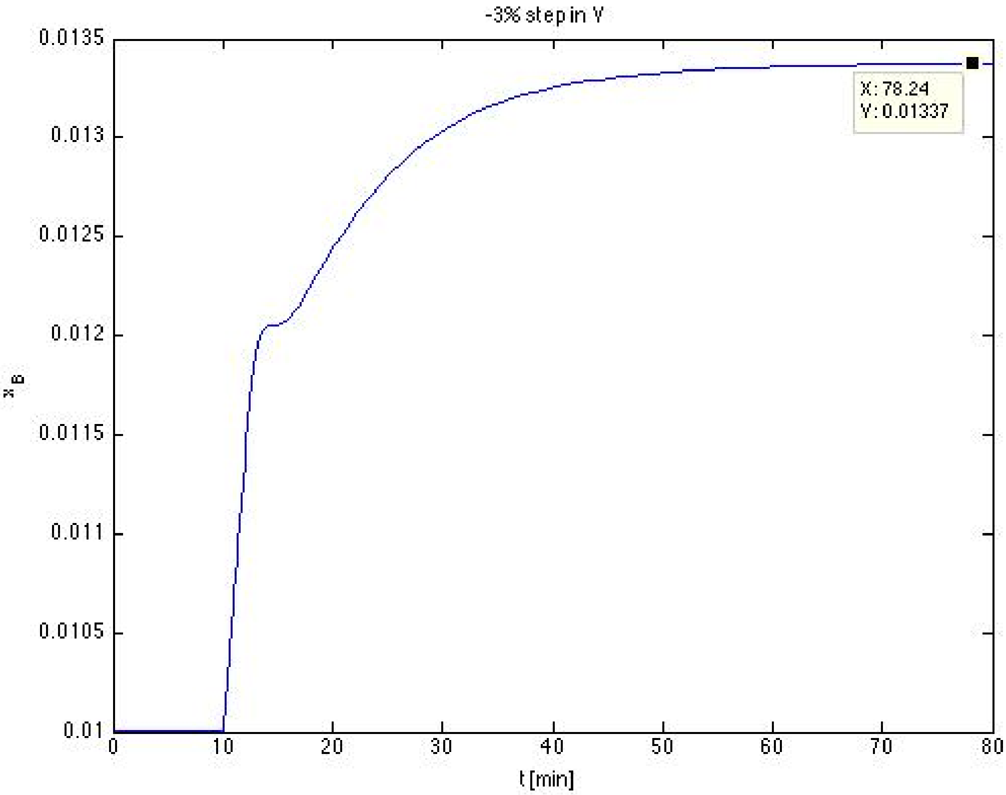

Figure D.3: Controller 2, 2nd tuning

Calculating the tuning parameters: Integrating process:

$$k' = \frac{\Delta y}{\Delta t \cdot \Delta u} = \frac{0.00146}{2.5 \cdot -2.5} = -0.007121 \tag{D.6}$$

Setting $\tau_c = 4$:

$$K_c = \frac{1}{-0.007121} \frac{1}{(0+1)} = -140.43/4 \tag{D.7}$$

$$\tau_I = 4\tau_c = 16 \tag{D.8}$$

Closing the loop and making a step in the setpoint for x_B from 0.010 to 0.015 to test the performance of the controller:

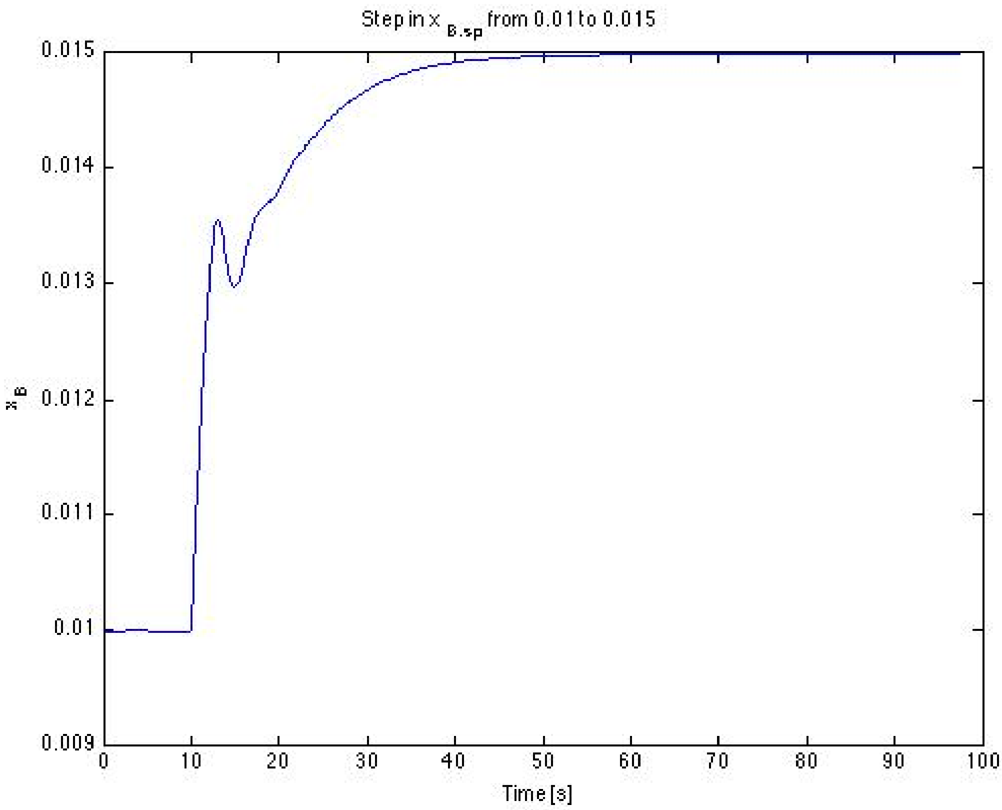

Figure D.4: Testing the performance of the controller

Choosing a larger τ_c for this controller (smoother control) gave better control of x_D, which is the most important variable to control in this case. This can be seen in Figure D.5.

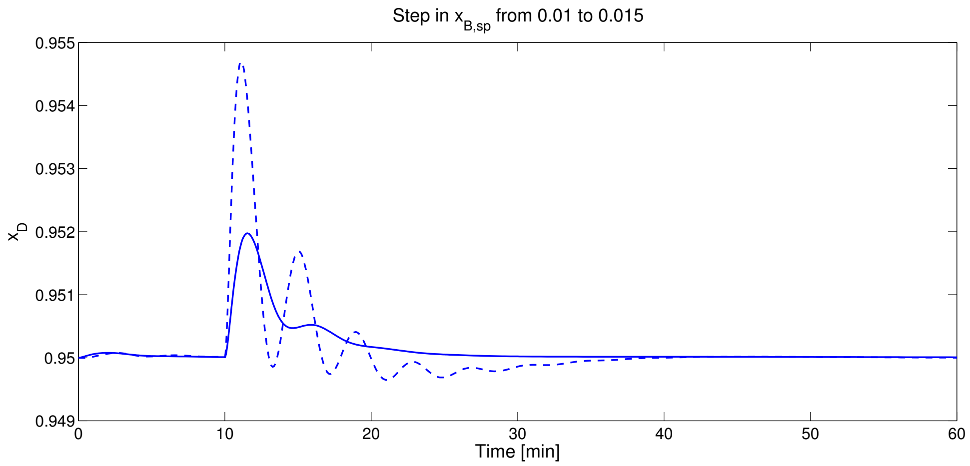

Figure D.5: Testing the effect of the value of τ_c on the response in x_D. Dashed line: $\tau_c = 1$. Solid line: $\tau_c = 4$

Appendix E Simulink block diagrams

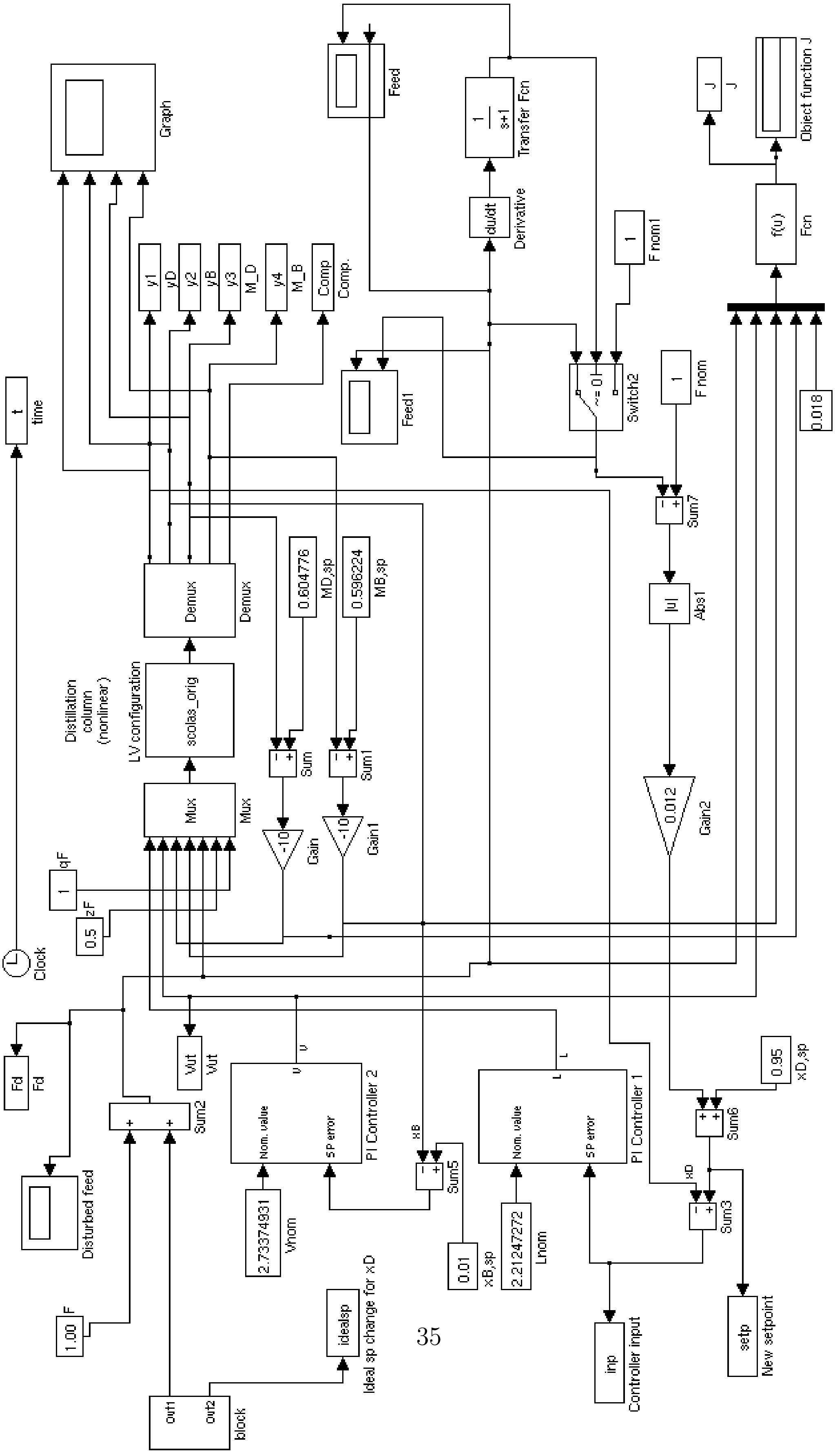

Figure E.1: Simulink block diagram for Case I

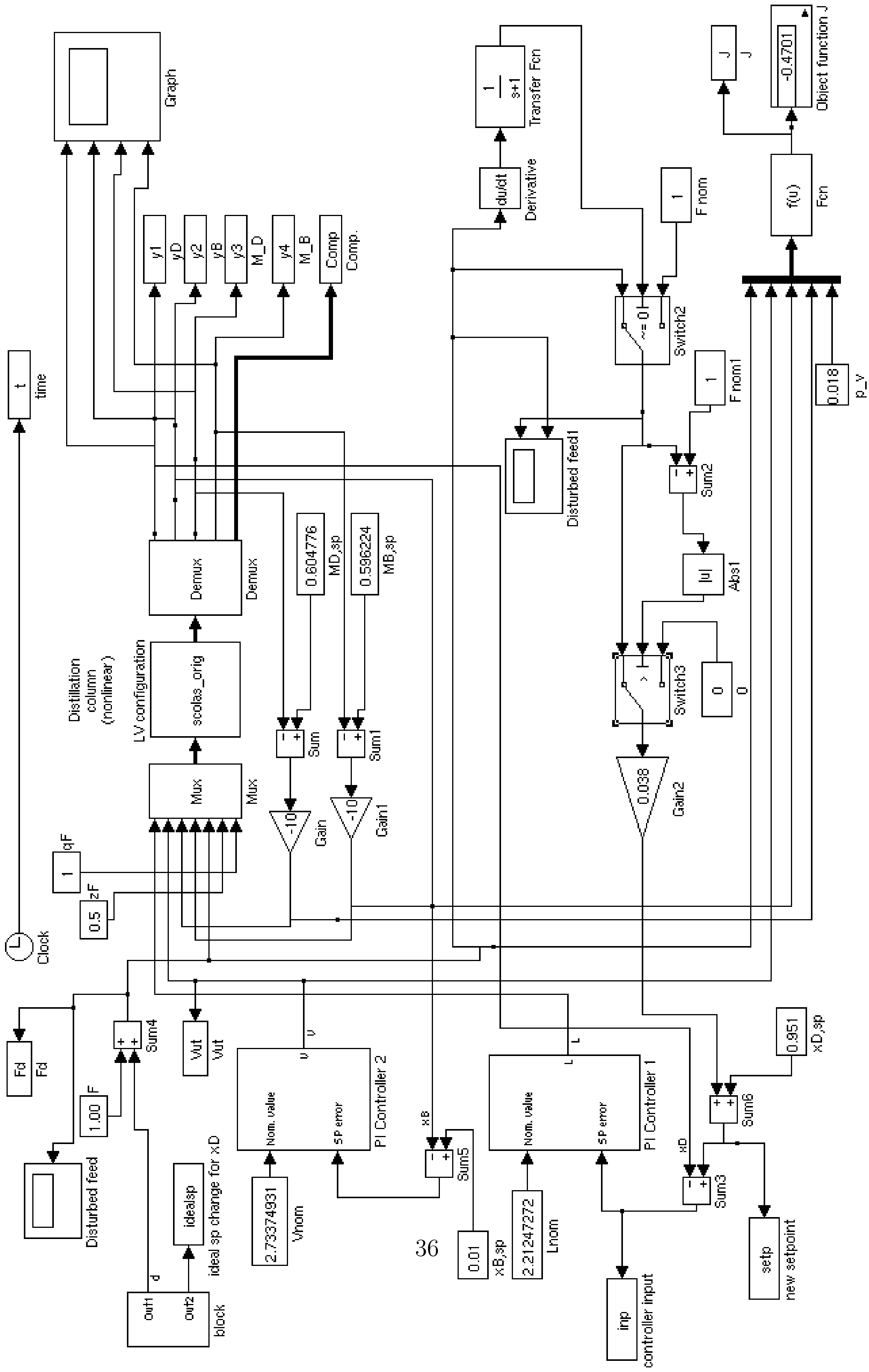

Figure E.2: Simulink block diagram for Case II