Vilde Haarsaker

Optimal setpoints for a pH neutralization process

Master thesis
for the degree of Master of Science

Trondheim, June 2013

Norwegian University of Science and Technology
The Faculty of Natural Sciences and Technology
Department of Chemical Engineering

NTNU
Innovation and Creativity

NTNU

Norwegian University of Science and Technology

Master thesis
for the degree of Master of Science

The Faculty of Natural Sciences and Technology
Department of Chemical Engineering

ISBN (printed version)
ISBN N/A (electronic version)
ISSN 1503-8181

Master theses at NTNU,

Printed by NTNU-trykk

Abstract

The purpose of this thesis was to determine the optimal setpoints for a pH neutralization process in three stages. A strong acid is to be neutralized by adding a strong base (reagent) in a series of three mixing tanks. The setpoints for the outlet concentration of each tank have to be selected such that the outlet concentration of the last times always satisfies the specification 0 ± 10^{-6} mol/l excess H^+. The critical issue is that we can never have a negative reagent flow rate.

We considered a worst-case scenario where all disturbances occur at the same time. The disturbances can be positive or negative, so we first had to determine the worst-case combination of signs. Then several cases with different setpoints were tested by invoking the disturbance combinations in a nonlinear Simulink model to obtain the maximum deviation from the setpoint in each tank. We also checked whether the reagent flow saturated or not.

This "wild guess" approach turned out to be very time consuming and not very iterative. A set of setpoints that would satisfy the specification for the last tank was not found. The best result was a deviation of $2.19 \cdot 10^{-6}$. However, it became clear that the setpoint for Tank 2 must not be too low, as this leads to saturation of the reagent flow rate q_r to Tank 3. A better approach would be to develop a model that could simulate a number of cases without changing the parameters for each case manually, and at the same time check whether q_r saturates or not.

Preface

This thesis was written as the final part of my M.Sc degree in Chemical Engineering at the Norwegian University of Science and Technology.

I would like to thank my co-supervisor, Post doc. Johannes Jäschke, for patiently helping me with the thesis throughout the semester. I also want to thank my supervisor, Professor Sigurd Skogestad for valuable input during the work with the thesis.

Finally, I would like to thank my fellow students at K4-230 for keeping a good spirit and contributing to a good working environment.

Declaration of Compliance

I hereby declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology (NTNU).

Trondheim, June 6th, 2012

Vilde Haarsaker

Contents

1 Introduction 1

1.1 Case study 1

1.2 Assumptions 3

1.3 Disturbances 3

2 Tank model 5

2.1 Nonlinear model 6

2.2 Linearized model 6

2.3 Control structure 10

2.3.1 Tuning of PI controllers 10

2.3.2 Saturation and anti-windup 12

2.3.3 Smith predictor 15

2.3.4 Controller performance 16

3 Determining the optimal setpoints 17

3.1 Overview of the procedure 17

3.2 Finding the worst-case disturbance combination 18

3.2.1 Linear simulation 22

3.2.2 Nonlinear simulation / Verification of the results . . . 24

4 Results 25

4.1 Worst-case disturbance combinations 26

4.2 Maximum deviation from setpoints 27

5 Discussion 31
5.1 Optimal setpoints 31
5.2 The Method 31
5.3 Method Improvements 32

6 Conclusions and further work 33

7 Appendices 37
A Representing in terms of excess H^+ 37
B Tank model 39
B.1 Component mass balance 39
B.2 Total mass balance 39
C Linearization 41
D Calculation of nominal values 43
E Tuning of PI controllers 44
F Closed-loop response for the linear model 46
G Matlab scripts 47
G.1 Finding the worst-case disturbance combinations . . . 47
G.2 Calculating SG_d (sgd.m) 54
G.3 Calculating nominal values (nomcalc.m) 54
G.4 Calculation process and disturbance gains (kcalc.m) . 55
G.5 Nonlinear Matlab model 57
G.6 Simulink interface to the nonlinear model 58
H Simulink models 60
H.1 Linear Simulink model 60
H.2 Nonlinear Simulink model 60

Chapter 1

Introduction

Faanes and Skogestad[1] have investigated control related design issues for neutralization plants, where a strong acid is neutralized by adding a strong base (from now on denoted reagent) in a series of mixing tanks. The aim of their work has been to give recommendations for the number of tanks and tank sizes. How to select the setpoints for the pH of the outlet stream from each tank is also discussed briefly, but there are not given any specific directions. The paper only states that "In conclusion, it is preferable to choose the setpoints as close as possible, but such that we never get negative reagent flow". The main objective of this thesis has been to determine what the setpoints of each tank should be.

1.1 Case study

The process that is studied is a plant where a strong acid with $pH = -1$ is to be neutralized by adding a reagent with $pH = 15$. The product specification is $pH = 7 \pm 1$. Faanes and Skogestad[1] have found that the lowest cost is obtained when using three tanks, and they recommend identical tank sizes with a total volume of $2V_0$, where for 3 tanks, $V_0 = 40.7\ \mathrm{m}^3$.

The volume of each tank is thus:

$$V_i = \frac{2 \cdot 40.7\ \mathrm{m}^3}{3} = 27.1333\ \mathrm{m}^3 = 27133.3\ \mathrm{l} \tag{1.1}$$

A simplified process flowsheet is shown in Figure 1.1.

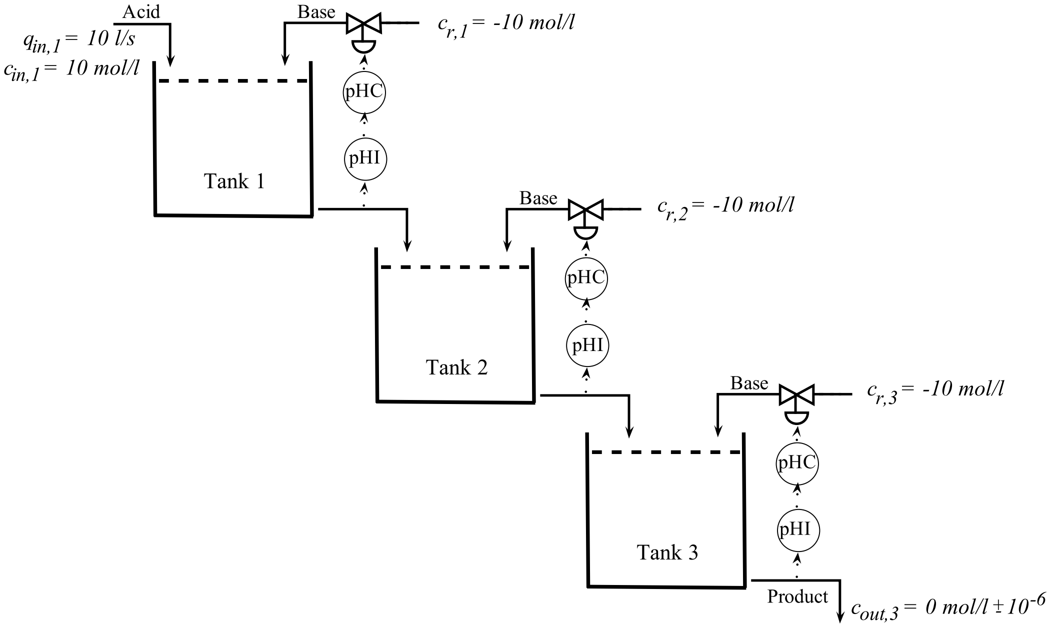

Figure 1.1: Flowsheet of the pH neutralization process

1.2 Assumptions

Faanes and Skogestad[1] follow Walsh[2] and assume that the overall effective delay in each tank is $\theta = 10$ s.

They also conclude that it is optimal to have identical controllers, provided that there are no restrictions on the controller. With PI control there may be a small benefit in having different volumes, but this benefit is most likely too small to offset the practical advantages of having identical units.

We assume constant tank volumes. This is a reasonable assumption, as we can imagine that we have overflow tanks.

We also neglect changes in the outlet flow rates of each tank. Faanes and Skogestad[1] explains this with "Often we may assume that the level controller is very slow, which leads to $q_i' \approx 0$." q_i' denotes the deviation from the nominal value of q_i, e.g. the disturbance in q_i.

1.3 Disturbances

We are considering four disturbances to Tank 1: The concentration and flow rate of the feed stream, and the concentration and flow rate of the reagent. In tank 2 and 3, the concentration disturbance is equal to the deviation from the setpoint in the previous tank. Since we are neglecting changes in the outlet flow rate of each tank, there will be no disturbances in the inlet flow rate to Tank 2 and 3, but disturbances in the reagent concentration and flow rate will be considered.

Generally, for Tank i, we have the following disturbances:

$$d_i = \begin{bmatrix} d_{cin,i}(s) \\ d_{qin,i}(s) \\ d_{cr,i}(s) \\ d_{qr,d,i}(s) \end{bmatrix} \tag{1.2}$$

Where $d_{qin,i} = 0$ for i = 2 and i = 3.

We are considering as disturbances a ±50% change in the inlet concentration and flow rate to Tank 1, and a ±10% change in the reagent concentration and reagent flow rate to all tanks.

Chapter 2

Tank model

A linear and a nonlinear model of the tank system are presented in the following sections. It is important to note that the linearized model is in terms of deviation variables, while the nonlinear model is not. E.g. in the linear model we will work with variables such as c'_{in1}, which denotes the deviation from the nominal value, while in the nonlinear model we use the actual values c_{in1} and so on.

It is convenient to derive a model in terms of excess H^+ concentration (See Appendix A), such that we define:

$$c = c_{H^+} - c_{OH^-} = 10^{-pH} - 10^{-14+pH} \tag{2.1}$$

We thus have, for the inlet stream, that $pH = -1$ corresponds to $c_{in,1} \approx 10$ mol/l, and for the reagent, $pH = 15$ corresponds to $c_r \approx -10$ mol/l.

The product specification, e.g. the outlet concentration in Tank 3, is $pH = 7 \pm 1$, which corresponds to:

$$c_3 \approx 0 \pm 10^{-6} \text{ mol/l} \tag{2.2}$$

2.1 Nonlinear model

The mixing process for each tank is described by the total and component mass balances, which are derived in Appendix B. A description of the symbols is given in Table 2.1.

Component mass balance:

$$\frac{dc_i}{dt} = \frac{1}{V_i}[(c_{in,i} - c_i)q_{in,i} + (c_{r,i} - c_i)q_{r,i}] \tag{2.3}$$

Total mass balance:

$$\frac{dV_i}{dt} = q_{in,i} + q_{r,i} - q_i \tag{2.4}$$

Table 2.1: Symbol list

Symbol	Description	Unit
i	Tank number	-
V_i	Tank volume	l
$c_{in,i}$	Inlet concentration	mol/l
$q_{in,i}$	Inlet flow rate	l/s
$c_{r,i}$	Reagent concentration	mol/l
$q_{r,i}$	Reagent flow rate	l/s
c_i	Outlet concentration	mol/l
q_i	Outlet flow rate	l/s

2.2 Linearized model

Linearization of the component mass balance (Eq. 2.3) around the steady-state nominal point (c_i^*, $c_{in,i}^*$, $c_{r,i}^*$, q_i^* $q_{in,i}^*$, $q{*}_{r,i}$) and Laplace transform yields (See Appendix C for the derivation):

$$c_i'(s) = \frac{1}{(\frac{V_i^*}{q_i^*})s + 1}\left[\frac{q_{in,i}^*}{q_i^*}c_{in,i}'(s) + \frac{c_{in,i}^* - c_i^*}{q_i^*}q_{in,i}'(s) + \frac{q_{r,i}^*}{q_i^*}c_{r,i}'(s) + \frac{c_{r,i}^* - c_i^*}{q_i^*}q_{r,i}'(s)\right]e^{-\theta s} \tag{2.5}$$

Where $c_i'(s)$, $c_{in,i}'(s)$, $q_{in,i}'(s)$, $c_{r,i}'(s)$ and $q_{r,i}'(s)$ are deviation variables:

$$c_i'(s) = c_i(s) - c_i^*(s) \tag{2.6}$$

A description of the symbols is given in Table 2.2.

Table 2.2: Symbol list

Symbol	Description	Unit
i	Tank number	-
V_i^*	Nominal tank volume	l
$c_{in,i}^*$	Inlet concentration, nominal value	mol/l
$c_{in,i}'$	Inlet concentration, deviation from nominal value	mol/l
$q_{in,i}^*$	Inlet flow rate, nominal value	l/s
$q_{in,i}'$	Inlet flow rate, deviation from nominal value	mol/l
$c_{r,i}^*$	Reagent concentration, nominal value	l/s
$c_{r,i}'$	Reagent concentration, deviation from nominal value	l/s
$q_{r,i}^*$	Reagent flow rate, nominal value	l/s
$q_{r,i}'$	Reagent flow rate, deviation from nominal value	l/s
c_i^*	Outlet concentration, nominal value	mol/l
c_i'	Outlet concentration, deviation from nominal value	mol/l
q_i^*	Outlet flow rate, nominal value	l/s
θ	Overall effective time delay	s

In each tank, the flow rate of base, q_r is used to adjust the concentration in the outlet stream from the tank. But q_r may also be considered a disturbance, as there are uncertainties associated with the valve. Thus, we have that:

$$q_{r,i}'(s) = q_{ri,u}'(s) + q_{ri,d}'(s) \tag{2.7}$$

Where $u_i'(s) = q_{ri,u}'$ is determined by the controller, and $q_{ri,d}'$ is the disturbance term. We then have that:

$$c_i'(s) = \frac{1}{(\frac{V_i^*}{q_i^*})s+1}\left[k_{d,i}d_i(s) + k_i u_i'(s)\right]e^{-\theta s} = \left[G_d(s)d_i(s) + G_i(s)u_i'(s)\right]e^{-\theta s} \tag{2.8}$$

Where the steady-state gains for different disturbances are:

$$k_{d,i} = \begin{bmatrix} k_{d,cin,i} \\ k_{d,qin,i} \\ k_{d,cr,i} \\ k_{d,qr,i} \end{bmatrix} = \begin{bmatrix} \frac{q^*_{in,i}}{q^*_i} \\ \frac{c^*_{in,i}-c^*_i}{q^*_i} \\ \frac{q^*_{r,i}}{q^*_i} \\ \frac{c^*_{r,i}-c^*_i}{q^*_i} \end{bmatrix} \tag{2.9}$$

The steady-state process gain is:

$$k_i = \frac{c^*_{r,i} - c^*_i}{q^*_i} \tag{2.10}$$

The concentration disturbance in Tank 2 is equal to the deviation from the setpoint for the outlet concentration of Tank 1, and equivalently from Tank 2 to Tank 3. Since the model is in terms of deviation variables, we can write that $d_{cin1} = c'_{in1}$ and so on:

$$d_1 = \begin{bmatrix} c'_{in1}(s) \\ q'_{in1}(s) \\ c'_{r1}(s) \\ q'_{r1,d}(s) \end{bmatrix} \tag{2.11}$$

$$d_2 = \begin{bmatrix} c'_{1,max}(s) \\ c'_{r2}(s) \\ q'_{r2,d}(s) \end{bmatrix} \tag{2.12}$$

$$d_3 = \begin{bmatrix} c'_{2,max}(s) \\ c'_{r3}(s) \\ q'_{r3,d}(s) \end{bmatrix} \tag{2.13}$$

A block diagram of the process is given in Figure 2.1.

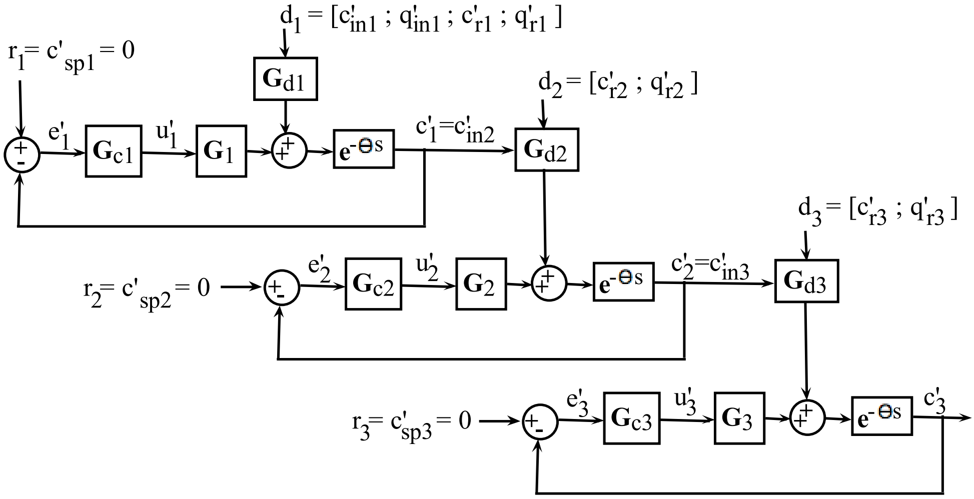

Figure 2.1: Block diagram of the process

2.3 Control structure

To compensate for the time delay in the tanks, a good choice of controller is the Smith predictor. The Smith predictor has an inner loop with a controller that can be designed without the deadtime. The outer loop has a process model $\widetilde{G}(s)$ that predicts the output of the process $G(s)$, which is fed to the primary controller.

A PI controller was selected for the inner loop. The controller was built in the following manner:

1. The SIMC Tuning Rules were used to obtain the tuning parameters for a simple PI controller.
2. A saturation block was added to prevent a negative reagent flow, and anti-windup was implemented.
3. The PI controller was incorporated as a part of the Smith Predictor.

2.3.1 Tuning of PI controllers

For tuning of PID controllers, Skogestad has developed a set of simple tuning rules, namely the SIMC (Simple/Skogestad Internal Model Control) tuning rules[4]. For a PI controller the steps will be like this:

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

$$G(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} \tag{2.14}$$

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:

$$k = \frac{\Delta y_\infty}{\Delta u} \tag{2.15}$$

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:

$$y_{63\%} = y_0 + 0.63 \cdot \Delta y \tag{2.16}$$

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

$$c(s) = K_c \cdot (1 + \frac{1}{\tau_I s}) \tag{2.17}$$

Controller gain:

$$K_c = \frac{1}{k'} \frac{1}{(\theta + \tau_c)} \tag{2.18}$$

Integral time:

$$\tau_I = \min(\tau_1, 4(\tau_c + \theta)) \tag{2.19}$$

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

$$k' = \frac{k}{\tau_1} \tag{2.20}$$

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

$$k' = \frac{\Delta y}{\Delta t \cdot \Delta u} \tag{2.21}$$

τ_c is a tuning parameter that needs to be selected. Skogestad[4] states that a small τ_c should be chosen for fast speed of response and good disturbance rejection, and a large τ_c for stability, robustness and small input usage. The first corresponds to tight control, the latter to smooth control.

The linear model of the first tank with $c_{1,sp} = 0.05$ was used for the tuning of the controllers, and the following tuning parameters were obtained (See Appendix E):

$$K_c = -133 \tag{2.22}$$

$$\tau_I = 80 \tag{2.23}$$

These parameters were used for all controllers, regardless of setpoint. It is shown in Appendix E that almost exactly the same parameters were obtained when using Tank 3 with $c_{3,sp} = 0$ to tune the controller.

2.3.2 Saturation and anti-windup

In practice, the controller output is limited due to physical constraints. In our case, the reagent flow rate cannot become negative. To avoid this, a saturation block can be added to the controller. The lower limit of the saturation block is equal to $-q_{r,nom}$, as the controller input and output are in terms of deviation variables.

However, when the the output saturates, we need to ensure that the error is not further integrated. When the actuator (in this case the control valve) reaches its limits, the feedback loop will be broken and the system will run as open loop because the actuator will remain at its limit independently of the process output [5]. This will cause the error to continue to be integrated, such that the integral term becomes very large - it "winds up". This buildup of the integral term while the controller is saturated is called integral windup or reset windup. The only way for the controller to return to normal operation by itself, is that the error must have must have opposite sign for a long period.

To prevent integral windup, an approach is to use the the back-calculating anti-windup method[6] . This method uses a feedback loop to discharge the integrator when the controller hits specified saturation limits. The integral action resumes when the output is no longer saturated.

Figure 2.2 shows a PI controller with a saturation block and a feedback loop with the back-calculation gain K_b. When the controller output exceeds its limit, the saturation block will keep the output at its limit, such that the input and output of the saturation block is no longer the same. Then the difference between the saturation block input and output will be multiplied with K_b and added to the integral gain $I = K_c/\tau_I$, which then is multiplied with the integral term $1/s$. In this case I has a negative sign, so K_b must be a positive number in order to reduce the impact of the integral term.

As long as the controller output is not saturated, the input and output of the saturation block will be the same, and nothing will happen.

In this case, the value of K_b was arbitrarily set to 10, as the magnitude of K_b proved to have no impact on the controller performance.

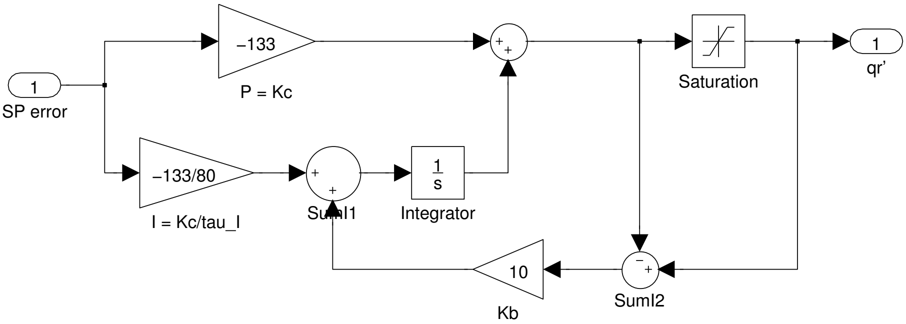

Figure 2.2: The PI controller with saturation block and back-calculating anti-windup.

2.3.3 Smith predictor

After tuning the PI controller and including anti-windup, the PI controller could be implemented as a part of the Smith predictor. For the linear Simulink model, $G(s) = \widetilde{G}(s)$, while for the nonlinear Simulink model, the process is described by the mass balances modeled in Matlab.

The process model $\widetilde{G}(s)$ is in terms of deviation variables, so the input and output of the Smith Predictor shown in Figure 2.3 are also deviation variables. u' is the deviation from the nominal reagent flow rate $q^*_{r,u}$, the process output c' is the deviation from the nominal value $c^* = c_{sp}$, and we always have that $c'_{sp} = 0$.

The closed-loop output from the Smith Predictor is (See Appendix F):

$$u'(s) = \frac{G_{C,PI}(s)e(s)}{1 + G_{C,PI}(s)\widetilde{G}(s)(1 - e^{\widetilde{\theta}s})} \tag{2.24}$$

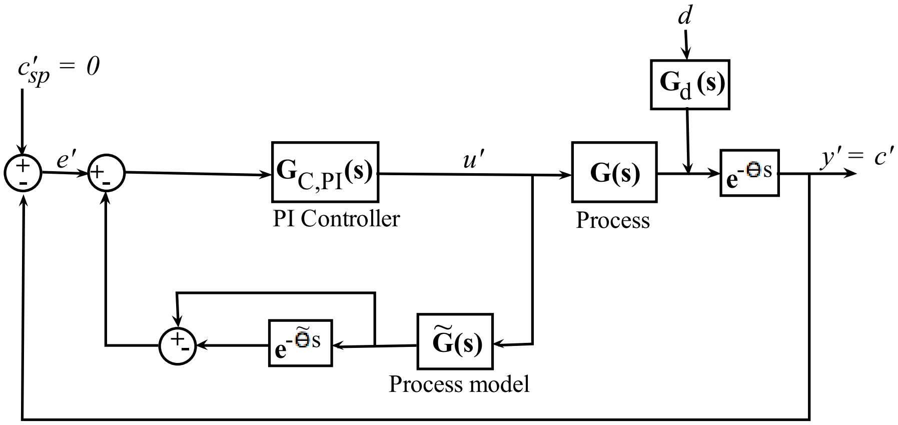

Figure 2.3: The Smith Predictor

2.3.4 Controller performance

Fig.2.4 compares the performance of the simple PI controller, the PI controller with anti-windup and the Smith Predictor with anti-windup. A step in $c_{1,sp}$ from 0.05 to 0.1 is made, and the response in c_1 plotted.

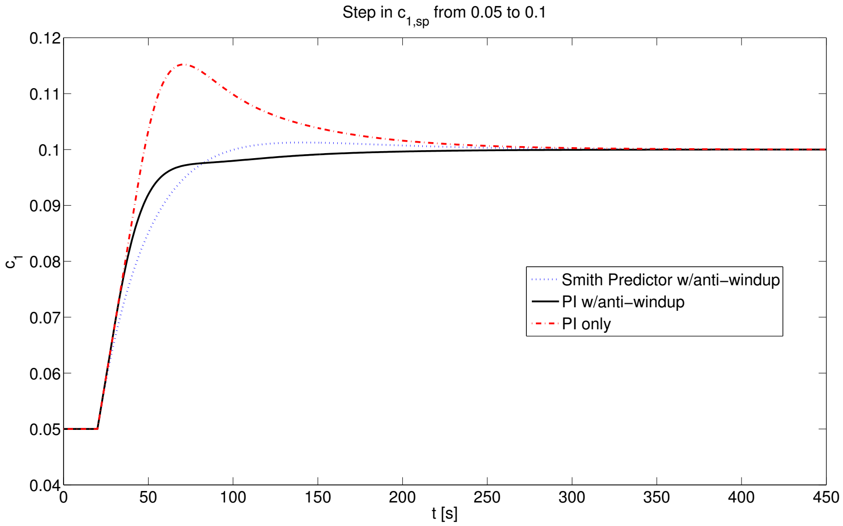

Figure 2.4: This plot compares the performance of the controllers when a step change in the setpoint is made.

Chapter 3

Determining the optimal setpoints

3.1 Overview of the procedure

This is an iterative procedure, where we start out with a set of arbitrarily selected setpoints, examine the results, and repeat the whole procedure with a new set of setpoints.

The first step of the procedure is to find the worst-case disturbance combinations for each tank. We will consider the worst-case scenario where all disturbances are occurring at the same time, and because each disturbance can be either positive or negative, we need to find the combination of signs that gives the largest deviation from the setpoint for the outlet concentration of the tank. We use the linear model derived in Section 2.2 to calculate the deviation from the setpoint for all disturbance combinations, using a Matlab script. The maximum deviation from the setpoint in Tank 1 is used as the concentration disturbance to Tank 2, and equivalently from Tank 2 to Tank 3. When we have found the worst-case disturbance combination for each tank, we will apply these in a linear Simulink model to see how the reagent flow rate is affected. The critical issue here is that we never can have negative reagent flow rate. We will also use the linear Simulink model to obtain more correct values for the maximum deviations from the setpoints. In the Matlab script we have modeled a Smith predictor with a simple PI controller, while in the linear Simulink model, we have included saturation and anti-windup in the PI controller. This improves the controller performance, as shown in Section 2.3.4. Finally, we will also do a nonlinear simulation in Simulink to compare the results from the linear simulation.

We are considering step disturbances. The concentration disturbances to Tank 2 and Tank 3, which are equal to the deviation from the setpoint for the outlet concentration from the previous tank, will certainly not appear as steps. However, we will determine the maximum of the concentration profile from the outlet stream from Tank 1 and Tank 2 and use this as the magnitude of a step disturbance in the inlet concentration to Tank 2 and Tank 3. This simplification increases the impact of the disturbance, as it is harder to keep the setpoint when you have a sudden disturbance like a step, than when the disturbance arises gradually.

Summarized, we can say that the procedure consists of the following three steps:

1. Find the worst-case disturbance combination using a linear Matlab model.
2. Invoking these disturbances in a linear Simulink model to obtain the maximum deviation from the setpoints in each tank and examine how the reagent flow rate is affected.
3. Repeating step 2 with a nonlinear Simulink model to compare the results.

3.2 Finding the worst-case disturbance combination

Inserting the closed-loop output from the Smith Predictor (Eq. 2.24) into the Laplace transformed linearized model (Eq. 2.8), we obtain the closed-loop response for Tank i (See Appendix F for a more detailed derivation):

$$c_i'(s) = S_i(s)G_{d,i}(s)d_i(s)e^{-\theta s} \tag{3.1}$$

Where $c_i'(s)$ is the deviation from the setpoint in Tank i, $G_{d,i}$ is the vector of disturbance transfer functions, $d_i(s)$ is the disturbance vector, θ is the overall effective time delay, and S_i is the sensitivity:

$$S_i(s) = \frac{1}{1 + \frac{G_i(s)G_{C,PI,i}(s)e^{-\theta s}}{1+G_{C,PI,i}\widetilde{G}_i(s)(1-e^{-\widetilde{\theta}s})}} \tag{3.2}$$

Since the disturbances can be both positive and negative, we need to find the combination of signs that gives the largest deviation from the setpoint. Because we are working with a linear model, we can simply calculate the deviation from the setpoint for the disturbances one by one and add them up to see which combination of signs that yields the largest deviation from the setpoint.

Single disturbances:
To find c_{max}' for a certain disturbance, say d_{cin1}, we have to find the maximum value $S_1(s)G_{dcin}(s)$ can take and multiply it with d_{cin1}. Finding the maximum value is the same as computing the norm in Matlab.

Matlab can compute two different norms:

1. The H_2 norm: The root-mean-square of the impulse response of the system.

2. The H_∞ norm: The peak gain of the frequency response

We have to make use of the H_∞ norm, as we need to consider the maximum output.

Let the norm of $S_i(s)G_{d,i}(s)$ be denoted N to simplify the notation:

$$N_{ij} = norm(S_i G_{d,j}) \tag{3.3}$$

Where i is the tank number and j is the selected disturbance, e.g.:

$$N_{cin1} = norm(S_1 G_{d,cin1}) \tag{3.4}$$

Such that the maximum output when $d = d_{cin1} = c'_{in1}$ is:

$$c'_{1max} = N_{cin1} d_{cin1} \tag{3.5}$$

Multiple disturbances:
As mentioned earlier, we can simply add up c'_{max} for each of the disturbances to find the worst-case deviation from the setpoint. In Tank 1, there are four disturbances, $c_{in,1}$, $q_{in,1}$, c_{r1} and q_{r1}, such that we need to consider the following combinations:

$$C_1 = \begin{bmatrix} c'_{max,cin1} + c'_{max,qin1} + c'_{max,cr1} + c'_{max,qr1} \\ c'_{max,cin1} + c'_{max,qin1} + c'_{max,cr1} - c'_{max,qr1} \\ c'_{max,cin1} + c'_{max,qin1} - c'_{max,cr1} + c'_{max,qr1} \\ c'_{max,cin1} + c'_{max,qin1} - c'_{max,cr1} - c'_{max,qr1} \\ c'_{max,cin1} - c'_{max,qin1} + c'_{max,cr1} + c'_{max,qr1} \\ c'_{max,cin1} - c'_{max,qin1} + c'_{max,cr1} - c'_{max,qr1} \\ c'_{max,cin1} - c'_{max,qin1} - c'_{max,cr1} + c'_{max,qr1} \\ c'_{max,cin1} - c'_{max,qin1} - c'_{max,cr1} - c'_{max,qr1} \end{bmatrix} \tag{3.6}$$

Note that due to the linearity of the model, we have that:

$$c'_{max,cin1} - c'_{max,qin1} + c'_{max,cr1} + c'_{max,qr1} = -[-c'_{max,cin1} + c'_{max,qin1} - c'_{max,cr1} - c'_{max,qr1}] \tag{3.7}$$

And so on, such that we for the first tank only have to calculate 8 values and not 16.

The maximum deviation from the setpoint in the first tank is finally found as the maximum value of the vector C_1:

$$c'_{1,max} = max\{abs(C_1)\} \tag{3.8}$$

For Tank 2 and 3, changes in the flowrate are neglected, such that there only are three disturbances, where the concentration disturbance is equal to the maximum deviation from the setpoint of the previous tank. We thus have that:

$$d_{cin2} = c'_{1,max} \tag{3.9}$$

$$C_2 = \begin{bmatrix} c'_{max,cin2} + c'_{max,cr2} + c'_{max,qr2} \\ c'_{max,cin2} - c'_{max,cr2} + c'_{max,qr2} \\ c'_{max,cin2} + c'_{max,cr2} - c'_{max,qr2} \\ c'_{max,cin2} - c'_{max,cr2} - c'_{max,qr2} \end{bmatrix} \tag{3.10}$$

Equivalently as for Tank 2, we have that the concentration disturbance to Tank 3 is:

$$d_{cin3} = c'_{2,max} = max\{abs(C_2)\} \tag{3.11}$$

$$C_3 = \begin{bmatrix} c'_{max,cin3} + c'_{max,cr3} + c'_{max,qr3} \\ c'_{max,cin3} - c'_{max,cr3} + c'_{max,qr3} \\ c'_{max,cin3} + c'_{max,cr3} - c'_{max,qr3} \\ c'_{max,cin3} - c'_{max,cr3} - c'_{max,qr3} \end{bmatrix} \tag{3.12}$$

And finally, the maximum deviation from the setpoint in the last tank is found:

$$c'_{3,max} = max\{abs(C_3)\} \tag{3.13}$$

Where the requirement for $c'_{3,max}$ is:

$$c'_{3,max} < 10^{-6} \tag{3.14}$$

This method was modeled in Matlab, and the code is attached in Appendix G.1.

However, as mentioned in the overview of the procedure, we have modeled the controller without a saturation block and anti-windup. This gives poorer controller performance, such that when we do the same simulation with the Simulink models (which have controllers with a saturation block and anti-windup), we might satisfy the requirement due to better controller performance.

We should now have found which combination of signs that yields the maximum deviation from the setpoint in each tank, such that we can invoke these disturbances in the linear Simulink model to see how this affects the reagent flow rate and to obtain more precise values for the maximum deviation from the setpoint in each tank.

3.2.1 Linear simulation

The linearized model was used to model the system in Simulink, see Fig. 3.1. A description of the modifications that have to be made to the model for every case and tank is attached in Appendix H.1. The tanks were modeled separately, such that we can simulate the tanks one by one. We first apply the worst-case disturbance combination for Tank 1 found in Section 3.2 to find the maximum deviation from the setpoint ($c_{1,max}$) and to examine the reagent flow rate. We then use $c_{1,max}$ as the concentration disturbance to Tank 2 together with the other two disturbances in the worst-case combination. We find $c_{2,max}$ and examine the reagent flow rate, and finally we repeat this procedure for Tank 3.

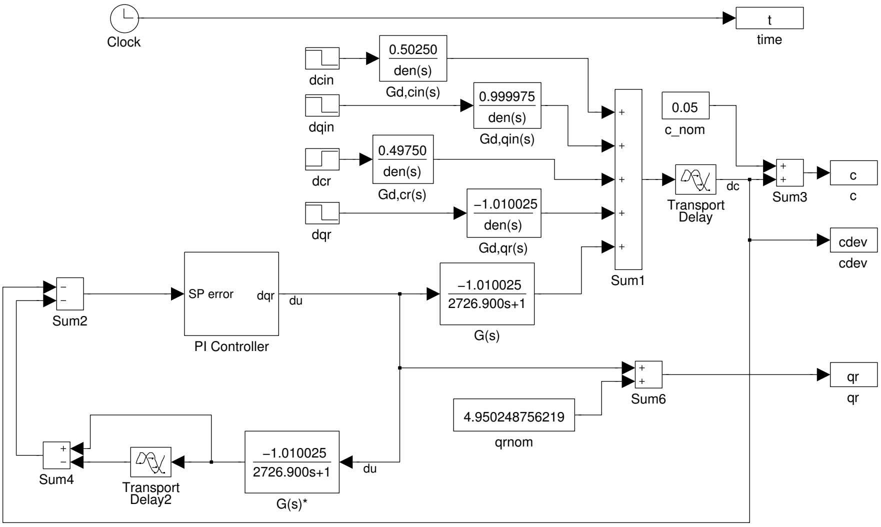

Figure 3.1: The linear Simulink model

3.2.2 Nonlinear simulation / Verification of the results

To verify the results from the linear simulation, we will repeat the procedure in the previous section (3.2.1) with a nonlinear simulation. The nonlinear component and total mass balances (Eq. 2.3 and 2.4) were used to model the tank in Matlab, see Appendix G.5. The Simulink interface to the nonlinear Matlab model is attached in Appendix G.6, and the Simulink model is shown in Figure 3.2 below. A description of the modifications that have to be made to the Simulink model for every case and tank is attached in Appendix H.2.

We will repeat this procedure with different sets of setpoints until we have satisfied Equation 3.14.

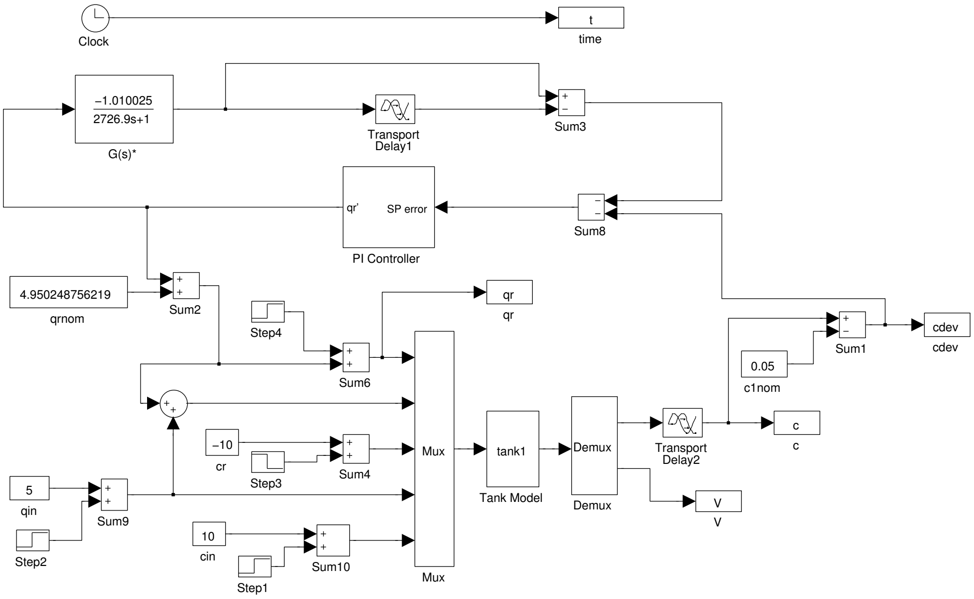

Figure 3.2: The nonlinear Simulink model

Chapter 4

Results

The cases with different sets of setpoints that were tested are presented in Table 4.1. Some selected nominal values are listed in Table 4.2.

Table 4.1: The selected setpoints for the outlet concentration in each tank.

	Tank 1	Tank 2	Tank 3
Case 1	0.05	0.0001	0
Case 2	0.001	0.00001	0
Case 3	0.5	0.01	0
Case 4	0.1	0.01	0
Case 5	0.5	0.05	0
Case 6	0.5	0.005	0
Case 7	1.0	0.01	0

Table 4.2: Nominal values

	Tank 1			Tank 2			Tank 3		
Case	q_r^*	q_{in}^*	τ	q_r^*	q_{in}^*	τ	q_r^*	q_{in}^*	τ
1	4.950	5.0	2726.9	0.0497	9.9503	2713.4	$9.9999{\cdot}10^{-5}$	9.99990	2713.3
2	4.999	5.0	2713.6	0.00099	9.9990	2713.3	$9.9999{\cdot}10^{-6}$	9.99999	2713.3
3	4.524	5.0	2849.0	0.46620	9.5238	2716.1	0.009990	9.99001	2713.3
4	4.901	5.0	2740.5	0.08902	9.9010	2716.0	0.009990	9.99001	2713.3
5	4.524	5.0	2849.0	0.42644	9.5238	2726.9	0.049751	9.95025	2713.3
6	4.524	5.0	2849.0	0.47119	9.5238	2714.7	0.0049975	9.99503	2713.3
7	4.091	5.0	2984.7	0.8991	9.0909	2716.0	0.00999	9.99000	2713.3

4.1 Worst-case disturbance combinations

For all cases, it was found that the combination of disturbances that gave the maximum deviation from the setpoint were the following:

Tank 1:

$$d_1 = \begin{bmatrix} c'_{in1} \\ q'_{in1} \\ -c'_{r1} \\ q'_{r1,d} \end{bmatrix} \tag{4.1}$$

Tank 2:

$$d_2 = \begin{bmatrix} c'_{in2} = -c'_{1,max} \\ -c'_{r2} \\ q'_{r2,d} \end{bmatrix} \tag{4.2}$$

Tank 3:

$$d_3 = \begin{bmatrix} c'_{in3} = -c'_{2,max} \\ -c'_{r3} \\ q'_{r3,d} \end{bmatrix} \tag{4.3}$$

As explained in Section 3.2 (See Eq. 3.7), we have that disturbance combinations with opposite signs yield the same absolute value of the deviation from the setpoint for the outlet concentration. Since the critical issue here is that we never can have negative reagent flow, we have to consider the combination of signs that require $q_{r,u,i}$ to decrease.

In practice this means that we, for example for Tank 1, have to check whether it is the combination where c'_{r1} is positive or negative that yields a decreasing reagent flow rate.

Tank 1: c'_{in1}, q'_{in1}, $-c'_{r1}$, $q'_{r1,d}$ is equivalent to $-c'_{in1}$, $-q'_{in1}$, c'_{r1}, $-q'_{r1,d}$

Tank 2: c'_{in2}, $-c'_{r2}$, $q'_{r2,d}$ is equivalent to $-c'_{in2}$, c'_{r2}, $-q'_{r2,d}$

Tank 3: c'_{in3}, $-c'_{r3}$, $q'_{r3,d}$ is equivalent to $-c'_{in3}$, c'_{r3}, $-q'_{r3,d}$

In Table 4.3, the maximum deviations from the setpoint that were caused by these disturbance combinations are listed. Recall that this model has a controller with poorer performance, which results in a larger deviation from the setpoints than what we will obtain with the Simulink models.

Table 4.3: Maximum deviation from the setpoints for the outlet concentration in each tank, obtained by the linear Matlab model (with poorer controller performance).

	Tank 1	Tank 2	Tank 3
Case 1	$\pm\ 4.44{\cdot}10^{-2}$	$\pm\ 4.04{\cdot}10^{-4}$	$\pm\ 3.16{\cdot}10^{-6}$
Case 2	$\pm\ 4.48{\cdot}10^{-2}$	$\pm\ 3.36{\cdot}10^{-4}$	$\pm\ 2.52{\cdot}10^{-6}$
Case 3	$\pm\ 4.13{\cdot}10^{-2}$	$\pm\ 9.89{\cdot}10^{-4}$	$\pm\ 2.23{\cdot}10^{-5}$
Case 4	$\pm\ 4.40{\cdot}10^{-2}$	$\pm\ 4.58{\cdot}10^{-4}$	$\pm\ 1.82{\cdot}10^{-5}$
Case 5	$\pm\ 4.13{\cdot}10^{-2}$	$\pm\ 9.27{\cdot}10^{-4}$	$\pm\ 8.11{\cdot}10^{-5}$
Case 6	$\pm\ 4.13{\cdot}10^{-2}$	$\pm\ 9.96{\cdot}10^{-4}$	$\pm\ 1.49{\cdot}10^{-5}$
Case 7	$\pm\ 3.81{\cdot}10^{-2}$	$\pm\ 2.68{\cdot}10^{-3}$	$\pm\ 2.68{\cdot}10^{-5}$

4.2 Maximum deviation from setpoints

Running the linear and nonlinear Simulink tank models one by one, we obtain the maximum deviation from the setpoints for the outlet concentration in each tank, and use the maximum deviation from the setpoint in one tank as the concentration disturbance to the next tank. The results are listed in Table 4.4 and Table 4.5. A typical concentration profile in a tank where q_r does not saturate is shown in Figure 4.1, and a typical concentration profile where q_r *does* saturate is shown in Figure 4.2.

In some of the cases, $q_{r,u,i}$ saturated, and these results are listed in inside a parenthesis. Most of the simulations showed that the disturbance combination with a positive $c'_{r,i}$ required $q_{r,u,i}$ to decrease. The simulations where $c'_{r,i}$ is negative is denoted with an *.

After simulating the four first cases both linearly and nonlinearly, it became clear that the nonlinear model gave slightly better results, so to simplify the procedure, the linear simulation was excluded in the remaining cases.

Table 4.4: **Linear model:** Max. deviation from the setpoints for the outlet concentration in each tank.

	Tank 1	Tank 2	Tank 3
Case 1	- $3.83{\cdot}10^{-2}$	- $2.69{\cdot}10^{-4}$	(- $1.49{\cdot}10^{-5}$)
Case 2	(- $1.92{\cdot}10^{-1}$)	($1.90{\cdot}10^{-1}$)	(- $1.90{\cdot}10^{-1}$)
Case 3	- $3.70{\cdot}10^{-2}$	- $5.82{\cdot}10^{-4}*$	$-1.85{\cdot}10^{-5}*$
Case 4	- $3.83{\cdot}10^{-2}$	$-1.95{\cdot}10^{-4}*$	- $2.35{\cdot}10^{-5}*$

Table 4.5: **Nonlinear model:** Max. deviation from the setpoints for the outlet concentration in each tank.

	Tank 1	Tank 2	Tank 3
Case 1	- $2.76{\cdot}10^{-2}$	- $1.78{\cdot}10^{-4}$	(- $6.54{\cdot}10^{-5}$)
Case 2	- $2.78{\cdot}10^{-2}$	(- $2.53{\cdot}10^{-2}$)	($-2.39{\cdot}10^{-2}$)
Case 3	- $2.65{\cdot}10^{-2}$	- $6.17{\cdot}10^{-4}*$	- $2.19{\cdot}10^{-6}*$
Case 4	- $2.75{\cdot}10^{-2}$	- $1.02{\cdot}10^{-4}$	$-1.69{\cdot}10^{-5}*$
Case 5	- $2.65{\cdot}10^{-2}$	- $5.47{\cdot}10^{-4}*$	$-8.40{\cdot}10^{-5}*$
Case 6	- $2.65{\cdot}10^{-2}$	- $6.26{\cdot}10^{-4}*$	$3.19\ {\cdot}10^{-6}$
Case 7	- $2.52{\cdot}10^{-2}$	- $1.41{\cdot}10^{-3}*$	$4.85{\cdot}10^{-6}$

*Opposite disturbance combination (e.g. c'_{in}, $-c'_r$, $q'_{r,d}$)

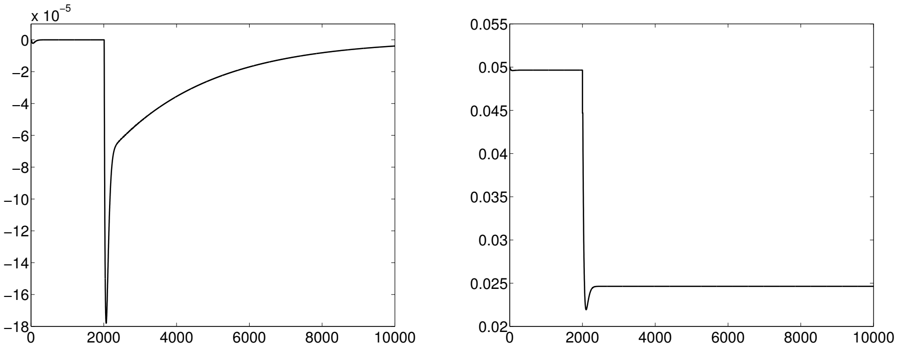

Figure 4.1: **Case 1, Tank 2, Nonlinear Model.** *Left:* Deviation from setpoint when the worst-case combination of all four disturbances is applied at the same time. Maximum deviation: $c'_{2,max} = -1.78 \cdot 10^{-5}$. *Right:* Reagent flow rate, $q_{r,u}$ (does not saturate)

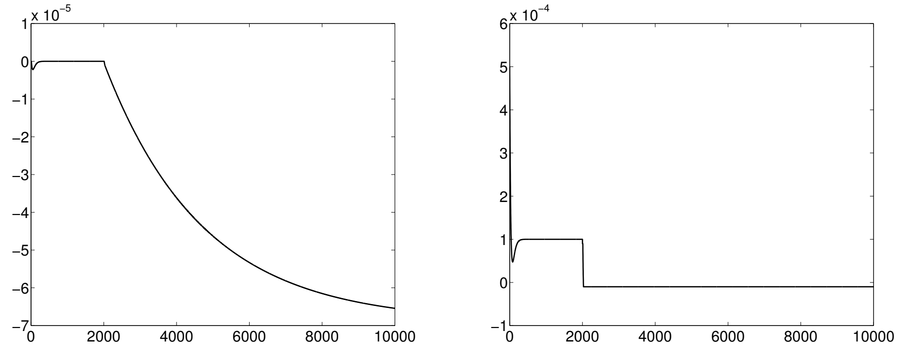

Figure 4.2: **Case 1, Tank 3, Nonlinear Model.** *Left:* Deviation from setpoint when the worst-case combination of all four disturbances is applied at the same time. Maximum deviation: $c'_{3,max} = -6.54 \cdot 10^{-5}$. *Right:* Reagent flow rate, $q_{r,u}$ (saturates).

Chapter 5

Discussion

5.1 Optimal setpoints

We were not able to find a combination of setpoints that ensured that the outlet concentration in Tank 3 always would satisfy the specification. However, we saw that a very low setpoint in Tank 2 lead to saturation of q_r. In Case 1, where $c_{sp2} = 0.0001$, q_r saturated in Tank 3, and in Case 2, where $c_{sp2} = 0.00001$, q_r saturated in both Tank 2 and Tank 3. For the cases where c_{sp2} ranged from 0.005 to 0.01, q_r did not saturate in neither Tank 2 or Tank 3.

The best result we were able to obtain, was a maximum deviation of $2.19 \cdot 10^{-6}$ in Tank 3. This is not very far from the specification, which is 10^{-6}. When we also keep in mind that we have implied step disturbances in the inlet concentration to each tank, which are stronger disturbances than the actual gradual disturbances, it might be that we in practice will satisfy the specification. At the same time we have neglected disturbances in the inlet flow rate to to Tank 2 and Tank 3.

5.2 The Method

This is a very time consuming method for determining the optimal setpoints. It is a very "wild guess" approach, and even after looking at several different cases, it was hard to see any pattern that could make the process more iterative.

As the disturbance combination that gave the maximum deviation from the setpoint was the same in all the cases, it is probably not necessary to run the Matlab script more than once. The Matlab script could possibly been used to indicate which setpoint combinations that give a small deviation from the setpoint of the last tank, but as long as it does not tell us whether q_r saturates or not, we have to perform a simulation in Simulink anyway.

To simplify the method the linear simulation in Simulink can be omitted, such that we only perform a nonlinear simulation. However, the nonlinear simulation is still time consuming, as we have to obtain the process transfer function and nominal values for every case. This was done running a separate Matlab script and inserting the values manually in the Simulink model. Other modifications, like changing the lower limit of the saturation block, the value of the step disturbance in the reagent flow rate, the inlet concentration, the initial value of the transport delay block, the initial concentration in the tank model in Matlab, also were time consuming. By doing all these changes manually, there is also a risk of forgetting to change some of the parameters or to make typing mistakes.

5.3 Method Improvements

To improve the method, the Simulink model could have been built such that the only parameters that had to be changed were the setpoints of each tank. Then a Matlab script could have been used to to run the tank models in Simulink with different selections of setpoints until a setpoint combination that satisfied the requirement in the last tank was found. We would also use the Matlab script to ensure that the reagent flow rate does not saturate.

Chapter 6

Conclusions and further work

Even though we were not able to obtain a combination of setpoints that would ensure that the outlet concentration in Tank 3 always would satisfy the specification, we obtained results that were quite close. The method was very time consuming and there was not observed any significant pattern that could help in the selection of setpoints for new cases. The only conclusion we can draw, is that the setpoint in Tank 2 should not be too low in order to avoid saturation of the reagent flow rate in Tank 3.

Further work on this subject could include to develop a Simulink model where the modifications for each case should not be performed manually. A Matlab script can be used to guess a number of setpoint combinations and run the Simulink model until it has found a setpoint combination that ensures that the specification in Tank 3 is satisfied and that the reagent flow rate does not saturate.

Bibliography

[1] Faanes, A., Skogestad, S., 2004: pH-neutralization: integrated process and control design. In: Computers and Chemical Engineering 28

[2] Walsh, S.: Integrated design of chemical waste water treatment systems. UK: Imperial College (1993)

[3] Seeborg, D., Edgar, T., Mellichamp, D., Doyle, F., 2011: Process Dynamics and Control, John Wiley and Sons, Inc, page 49 (linearization) and page 128 (anti-windup).

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

[5] Mathworks: `http://www.mathworks.com/matlabcentral/fileexchange/23965-anti-windup-pid-controller-simulink-block` (24.05.2013)

[6] Mathworks: `http://www.mathworks.se/products/simulink/examples.html?file=/products/demos/shipping/simulink/sldemo_antiwindup.html` (24.05.2013)

Chapter 7

Appendices

A Representing in terms of excess H^+

$$pH = -logc_{H^+} \tag{A.1}$$

$$c_{H^+} = 10^{-pH} \tag{A.2}$$

$$c_{OH^-} = \frac{K_W}{c_{H^+}} = \frac{10^{-14}}{10^{-pH}} = 10^{-14+pH} \tag{A.3}$$

We let c denote excess H^+ concentration:

$$c = c_{H^+} - c_{OH^-} = 10^{-pH} - 10^{-14+pH} \tag{A.4}$$

We then have for the inlet flow to the first tank:

$$c_{in1} = 10^{-(-1)} - \frac{10^{-14}}{10^{-(-1)}} = 10 - 10^{-15} = 10 \text{ mol/l} \tag{A.5}$$

For the reagent flow to each tank:

$$c_{r,i} = 10^{-15} - \frac{10^{-14}}{10^{-15}} = 10^{-15} - 10 = -10 \text{ mol/l} \tag{A.6}$$

And for the outlet flow from the last tank:

$$c_{out3} = 10^{-7} - \frac{10^{-14}}{10^{-7}} = 10^{-7} - 10^{-7} = 0 \text{ mol/l} \tag{A.7}$$

The specification for the outlet stream is $pH = 7 \pm 1$, which corresponds to the following upper and lower limits:

$$c_{out3,max} = 10^{-6} - \frac{10^{-14}}{10^{-6}} = 10^{-6} - 10^{-8} = 10^{-6} \text{ mol/l} \tag{A.8}$$

$$c_{out3,min} = 10^{-8} - \frac{10^{-14}}{10^{-8}} = 10^{-8} - 10^{-6} = -10^{-6} \text{ mol/l} \tag{A.9}$$

B Tank model

B.1 Component mass balance

For one tank, the general component mass balance can be written:

$$\frac{d}{dt}n_j = F_{j,in} + F_{j,r} - F_j + rV \tag{B.1}$$

Where F is the molar flow rate of component j in [mol/s].

As F = qV and n = cV, the mass balances for H^+ and OH^- can be written:

$$\frac{d}{dt}(c_{H^+}V) = c_{H^+in}q_{in} + c_{H^+r}q_r - c_{H^+}q + rV \tag{B.2}$$

$$\frac{d}{dt}(c_{OH-}V) = c_{OH^-in}q_{in} + c_{OH^-r}q_r - c_{OH^-}q + rV \tag{B.3}$$

By letting c denote the excess H^+ concentration c = c_{H^+} - c_{OH^-}, and subtracting the mass balance for OH^- from the mass balance for H^+, the component balance becomes:

$$\frac{d}{dt}(cV) = c_{in}q_{in} + c_r q_r - cq \tag{B.4}$$

B.2 Total mass balance

Total mass balance for one tank:

$$\frac{dV}{dt} = q_{in} + q_r - q \tag{B.5}$$

Chain rule:

$$\frac{d}{dt}(cV) = c\frac{dV}{dt} + V\frac{dc}{dt} \tag{B.6}$$

Rearranging:

$$\frac{dc}{dt} = \frac{1}{V}\left[\frac{d}{dt}(cV) - c\frac{dV}{dt}\right] \tag{B.7}$$

Inserting total mass balance and component mass balance:

$$\frac{dc}{dt} = \frac{1}{V}[c_{in}q_{in} + c_r q_r - cq - (q_{in} + q_r - q)] \tag{B.8}$$

Rearranging:

$$\frac{dc}{dt} = \frac{1}{V}[(c_{in} - c)q_{in} + (c_r - c)q_r] \tag{B.9}$$

Each tank can thus be described by Eq. B.4 and Eq. B.9, and more generally the equations can be written as follows (where the subscript i now denotes the tank number):

$$\frac{d}{dt}(c_i V_i) = c_{in,i}q_{in,i} + c_{r,i}q_{r,i} - c_i q_i \tag{B.10}$$

$$\frac{dc_i}{dt} = \frac{1}{V_i}[(c_{in,i} - c_i)q_{in,i} + (c_{r,i} - c_i)q_{r,i}] \tag{B.11}$$

C Linearization

As we are considering tanks with constant volumes, the component mass balance (Eq. B.9) for one tank can be written:

$$V\frac{dc}{dt} = (c_{in} - c)q_{in} + (c_r - c)q_r \tag{C.1}$$

This is a nonlinear equation on the form $\frac{dc}{dt} = f(c, u)$, and a linear approximation can be obtained using a Taylor series expansion and truncating after the first order terms [3].

$$V \cdot \frac{dc}{dt} = V \cdot f(c, \mathbf{u}) \cong f(c^*, \mathbf{u}^*) + \left.\frac{\partial f}{\partial c}\right|_{ss}(c - c^*) + \sum_{i=1}^{4} \left.\frac{\partial f}{\partial u_i}\right|_{ss}(u_i - u_i^*) \tag{C.2}$$

Where the steady-state operation corresponds to $f(c^*, \mathbf{u}^*) = 0$. $(c - c^*)$ and $(u_i - u_i^*)$ are replaced by the deviation variables c' and $\mathbf{u}'$:

$$V \cdot \frac{dc}{dt} \cong 0 + \left.\frac{\partial f}{\partial c}\right|_{ss} c' + \left.\frac{\partial c}{\partial c_{in}}\right|_{ss} c'_{in} + \left.\frac{\partial c}{\partial q_{in}}\right|_{ss} q'_{in} + \left.\frac{\partial c}{\partial c_r}\right|_{ss} c'_r + \left.\frac{\partial c}{\partial q_r}\right|_{ss} q'_r \tag{C.3}$$

$$V \cdot \frac{dc}{dt} = (-q^*_{in} - q^*_r)c' + q^*_{in}c'_{in} + (c^*_{in} - c^*)q'_{in} + q^*_r c'_r + (c^*_r - c^*)q'_r \tag{C.4}$$

Because c^* is a constant, it follows that:

$$\frac{dc}{dt} = \frac{d(c - c^*)}{dt} = \frac{dc'}{dt} \tag{C.5}$$

Where $q^* = q^*_{in} + q^*_r$, such that Equation C.4 can be written:

$$V \cdot \frac{dc}{dt} = -q^*c' + q^*_{in}c'_{in} + (c^*_{in} - c^*)q'_{in} + q^*_r c'_r + (c^*_r - c^*)q'_r \tag{C.6}$$

Laplace transform of Equation C.6 yields:

$$Vs[c'(s) - c'(0)] = -q^* c'(s) + q_{in}^* c_{in}'(s) + (c_{in}^* - c^*) q_{in}'(s) + q_r^* c_r'(s) + (c_r^* - c^*) q_r'(s) \tag{C.7}$$

Inserting the initial condition $c'(0) = 0$ and rearranging:

$$c'(s) = \frac{1}{(\frac{V^*}{q^*} s + 1)} \left[\frac{q_{in}^*}{q^*} c_{in}'(s) + \frac{c_{in}^* - c^*}{q^*} q_{in}'(s) + \frac{q_r^*}{q^*} c_r'(s) + \frac{c_r^* - c^*}{q^*} q_r'(s) \right] \tag{C.8}$$

Where

$$q_r'(s) = q_{r,d}'(s) + q_{r,u}'(s) \tag{C.9}$$

D Calculation of nominal values

The component balance derived in Appendix B is used to calculate nominal values for q_r:

$$V\frac{dc}{dt} = (c_{in} - c)q_{in} + (c_r - c)q_r \tag{D.1}$$

At steady-state, $\frac{dc}{dt} = 0$, such that Equation D.1 can be rearranged to get an expression for q_r^*:

$$q_r^* = \frac{(c_{in}^* - c^*)}{(c^* - c_r^*)} \cdot q_{in}^* \tag{D.2}$$

The nominal outflow rates from each tank:

$$q^* = q_{in}^* + q_r^* \tag{D.3}$$

E Tuning of PI controllers

The linear model of Tank 1 with $c_{1,sp} = 0.05$ was used to obtain the tuning parameters using the SIMC Tuning Rules[4]. The procedure is described in Section 2.3.1.

Making a -1% step change in the input $q_{r,1}$ and plotting the response of the output c_1:

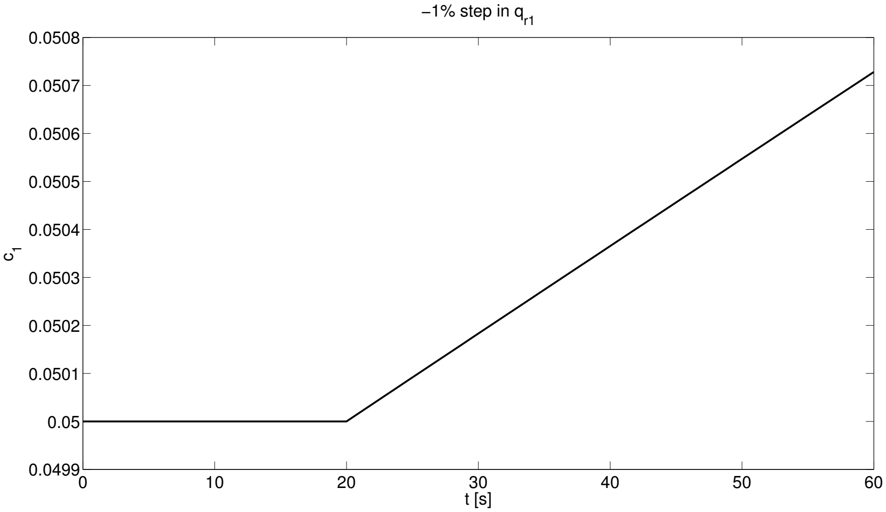

Figure E1: Step response, Case 1, Tank 3, where $c_{1,sp} = 0.05$.

Calculating the tuning parameters, integrating process:

$$k' = \frac{\Delta y}{\Delta t \cdot \Delta u} = \frac{0.05073 - 0.05}{40 \cdot (-0.01 \cdot 4.95)} = -3.76 \cdot 10^{-4} \tag{E.1}$$

Setting $\tau_c = \theta = 10$ s:

$$K_c = \frac{1}{k'} \frac{1}{(\theta + \tau_c)} = \frac{1}{-3.76 \cdot 10^{-4}} \frac{1}{(10 + 10)} = -133.0 \tag{E.2}$$

To verify that we can use the same tuning parameters for all tanks, independent of setpoint, we also obtain k' for Tank 3 from Case 1 (where $c_{3,sp} = 0$):

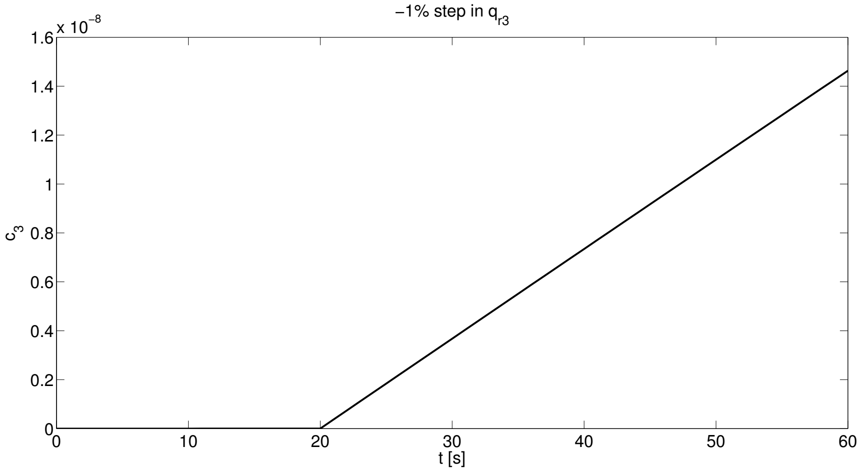

Figure E2: Step response, Case 1, Tank 3, where $c_{3,sp} = 0$.

Integrating process, calculating k':

$$k' = \frac{\Delta y}{\Delta t \cdot \Delta u} = \frac{1.099 \cdot 10^{-8} - 0}{30 \cdot (-0.01 \cdot 9.9999 \cdot 10^{-5})} = -3.74 \cdot 10^{-4} \qquad \text{(E.3)}$$

Which is practically equal to k' for the first tank, and we have verified that the same controller settings can be applied for all tanks and cases.

F Closed-loop response for the linear model

A Laplace transformed linearized model for the process was derived in Appendix C:

$$c_i(s) = \left[G_i(s)u_i(s) + G_{d,i}(s)d_i(s)\right] e^{-\theta s} \tag{F.1}$$

Recall that $c_i'(s)$ and $u_i'(s)$ are deviation variables.

Controller:

$$u_i'(s) = G_{C,PI,i}(s)\left[e_i'(s) - \widetilde{G}_i(s)(1 - e^{-\widetilde{\theta}s})u_i'(s)\right] \tag{F.2}$$

Rearranging and inserting for $e_i' = 0 - y_i' = -y_i' = -c_i'$:

$$u_i'(s) = \frac{G_{C,PI,i}(s)(-c_i')}{1 + G_{C,PI,i}\widetilde{G}_i(s)(1 - e^{-\widetilde{\theta}s})} \tag{F.3}$$

Inserting $u_i'(s)$ into Equation F.1:

$$c_i'(s) = \left[G_i(s)\frac{G_{C,PI,i}(s)(-c_i')}{1 + G_{C,PI,i}\widetilde{G}_i(s)(1 - e^{-\widetilde{\theta}s})} + G_{d,i}(s)d_i(s)\right] e^{-\theta s} \tag{F.4}$$

Rearranging:

$$c_i'(s) = \frac{1}{1 + \frac{G_i(s)G_{C,PI,i}(s)e^{-\theta s}}{1+G_{C,PI,i}(s)\widetilde{G}_i(s)(1-e^{-\widetilde{\theta}s})}}G_{d,i}(s)d_i(s)e^{-\theta s} = S_i(s)G_{d,i}(s)d_i(s)e^{-\theta s} \tag{F.5}$$

G Matlab scripts

G.1 Finding the worst-case disturbance combinations

This script calls the functions nomcalc.m (Section G.3), kcalc.m (Section G.4) and sgd.m (Section G.2).

```
% This script calculates the maximum deviation from the selected setpoints
% for the outlet concentration from each tank for a pH neutralization
% process with three tanks in series.

% The disturbances are the following:
%       Tank 1:     cin1,        qin1, cr1, qr1
%       Tank 2:     cin2 = c1max,      cr2, qr2
%       Tank 3:     cin3 = c2max,      cr3, qr3
% Changes in the flowrate q are neglected after the first tank.

% It passes the parameters to the function nomcalc, which
% calculates the nominal values for the inlet and outlet streams to each of
% the three tanks, e.g. flowrate of base and acid, concentration of acid,
% flowrate out of the tanks and the residence time tau in each tank.
% The function then uses the nominal values returned by nomcalc to
% calculate the steady-state gains for different disturbances and the
% control input gains for each tank.

clear all
clc

% Setpoints for the outlet concentration from each tank (excess
% H+ [mol/l] (Selected arbitrarily):

sp = [0.05 0.0001 0];   % Tank 1, Tank 2, Tank 3

% Time delay in each tank [s]:
theta = 10;

% Volume of each tank [l]
V = 2*40.7/3*1000;

% Controller settings:
```

```
Kc = -133;
tau_i = 80;

% Disturbances :
d_cin1 = 0.5;       % First tank (50%)
d_qin1 = 0.5;       % First tank (50%)
d_cr  = 0.1;        % All tanks  (10%)
d_qr  = 0.1;        % All tanks  (10%)

% Nominal values::
qin1_nom = 5;       % Nominal Flow rate of acid to Tank 1, [l/s]
pH_acid = -1;       % pH of feed stream (pH=-1 --> c = 10 mol/l excess H+)
pH_base = 15;       % pH of all reagent streams (pH=15 --> c = -10 mol/l excess H+)

% Concentration (excess H+) of acid stream to Tank 1:
cin1_nom = 10^(-pH_acid)-10^(-14+pH_acid);
% Concentration (excess H+) of base added to each tank:
cr_nom = 10^(-pH_base)-10^(-14+pH_base);

% Collecting the parameters in an array:
par_nom = [V cin1_nom qin1_nom cr_nom];

% Calculating nominal values
N = nomcalc(sp,par_nom);  % Returns nom = [qr_nom ; q_nom ; cin_nom ; qin_nom ; tau];

qr_nom = N(1,:);
q_nom = N(2,:);
cin_nom = N(3,:);
qin_nom = N(4,:);
tau = N(5,:);

% Calculating the steady state gains for different disturbances and the gain from
% the control input:
K = kcalc(N,cr_nom,sp);

% Extracting the arrays from the matrix K:
ktank1 = K(1,:);
ktank2 = K(2,:);
ktank3 = K(3,:);

% Process gains
```

```
Kp = [ktank1(5) ; ktank2(5) ; ktank3(5)];

% Defining s as the Laplace variable s
s = tf('s');

% Controller TF:
Gc = Kc*(1+1/(tau_i*s));

% Process model TF and sensitivity for each tank:
for i=1:3

    % Process model TF
    G(i) = Kp(i)/(tau(i)*s+1);

    % Sensitivity;    y(i) = sens(i)*gd(i)*d(i)
    sens(i) = 1/(1 + G(i)*Gc*exp(-theta*s)/(1+G(i)*Gc*(1-exp(-theta*s))));

end

% SGd transfer functions for each disturbance in Tank 1:
SGd_tank1_c  = sgd(ktank1(1),sens(1),tau(1));
SGd_tank1_q  = sgd(ktank1(2),sens(1),tau(1));
SGd_tank1_cr = sgd(ktank1(3),sens(1),tau(1));
SGd_tank1_qr = sgd(ktank1(4),sens(1),tau(1));

% SGd transfer functions for each disturbance in Tank 2:
SGd_tank2_c  = sgd(ktank2(1),sens(2),tau(2));
SGd_tank2_q  = sgd(ktank2(2),sens(2),tau(2));
SGd_tank2_cr = sgd(ktank2(3),sens(2),tau(2));
SGd_tank2_qr = sgd(ktank2(4),sens(2),tau(2));

% SGd transfer functions for each disturbance in Tank 3:
SGd_tank3_c  = sgd(ktank3(1),sens(3),tau(3));
SGd_tank3_q  = sgd(ktank3(2),sens(3),tau(3));
SGd_tank3_cr = sgd(ktank3(3),sens(3),tau(3));
SGd_tank3_qr = sgd(ktank3(4),sens(3),tau(3));

% Multiplying the norm(max values) of sens(i)*Gd(i) with the maximum expected
% disturbance to obtain the maximum deviation from the setpoint for the
% outlet concentration from each tank:
```

```
%__________________________________________________________________________
%/////////////////////////////////TANK 1: /////////////////////////////////

% Disturbances, Tank 1:
dcin1  = d_cin1*cin1_nom;
dqin1  = d_qin1*qin1_nom;
dcr1 = d_cr*cr_nom;
dqr1 = d_qr*qr_nom(1);


% _________________________Single disturbances:____________________________

% When d = c_in:
 c1(1) = norm(SGd_tank1_c,inf)*dcin1;

% When d = q_in:
 c1(2) = norm(SGd_tank1_q,inf)*dqin1;

% When d = cr1:
 c1(3) = norm(SGd_tank1_cr,inf)*dcr1;

% When d = +qr1:
 c1(4) = norm(SGd_tank1_qr,inf)*dqr1;

%__________________________Multiple disturbances:__________________________

% Four disturbances at a time:

c1m(1) = + c1(1) + c1(2) + c1(3) + c1(4);
c1m(2) = + c1(1) + c1(2) + c1(3) - c1(4);
c1m(3) = + c1(1) + c1(2) - c1(3) + c1(4);
c1m(4) = + c1(1) + c1(2) - c1(3) - c1(4);
c1m(5) = + c1(1) - c1(2) + c1(3) + c1(4);
c1m(6) = + c1(1) - c1(2) + c1(3) - c1(4);
c1m(7) = + c1(1) - c1(2) - c1(3) + c1(4);
c1m(8) = + c1(1) - c1(2) - c1(3) - c1(4);

%________________________Max. output from Tank 1:__________________________

c1max = max(abs(c1m));
```

```
%__________________________________________________________________________
%/////////////////////////////////TANK 2: /////////////////////////////////

dcin2  = c1max;
dqin2  = 0;
dcr2 = dcr1;
dqr2 = d_qr*qr_nom(2);

% _________________________Single disturbances:____________________________
% When d = c1max:
 c2(1) = norm(SGd_tank2_c,inf)*dcin2;

% When d = cr2:
 c2(2) = norm(SGd_tank2_cr,inf)*dcr2;

% When d = qr2:
 c2(3) = norm(SGd_tank2_qr,inf)*dqr2;

%_________________________Multiple disturbances:___________________________

c2m(1) = + c2(1) + c2(2) + c2(3);
c2m(2) = + c2(1) + c2(2) - c2(3);
c2m(3) = + c2(1) - c2(2) + c2(3);
c2m(4) = + c2(1) - c2(2) - c2(2);

%________________________Max. output from Tank 2:__________________________

c2max = max(abs(c2m));

%__________________________________________________________________________
%///////////////////////////////// TANK 3: ////////////////////////////////

dcin3  = c2max;
dqin3  = 0;
dcr3 = dcr1;
dqr3 = d_qr*qr_nom(3);

% _________________________Single disturbances:____________________________
% When d = c2max:
 c3(1) = norm(SGd_tank3_c,inf)*dcin3;
```

```
% When d = cr2:
 c3(2) = norm(SGd_tank3_cr,inf)*dcr3;

% When d = qr2:
 c3(3) = norm(SGd_tank3_qr,inf)*dqr3;

%________________________Multiple disturbances:____________________________

c3m(1) = c3(1) + c3(2) + c3(3);
c3m(2) = c3(1) + c3(2) - c3(3);
c3m(3) = c3(1) - c3(2) + c3(3);
c3m(4) = c3(1) - c3(2) - c3(3);

%________________________Max. output from Tank 3:__________________________

c3max = max(abs(c3m));


%__________________________________________________________________________

% Finding the indices of the worst-case combinations:

index1 = find(c1m==(max((max(c1m)))));
index2 = find(c2m==(max((max(c2m)))));
index3 = find(c3m==(max((max(c3m)))));

% -------------------Displaying the results:-------------------------------

format shortEng

if index1 == 1
    comb1 = 'cin1, qin1, cr1, qr1';
elseif index1 == 2
    comb1 = 'cin1, qin1, cr1, -qr1';
elseif index1 == 3
    comb1 = 'cin1, qin1, -cr1, qr1';
elseif index1 == 4
    comb1 = 'cin1, qin1, -cr1, -qr1';
elseif index1 == 5
    comb1 = 'cin1, -qin1, cr1, qr1';
```

```
elseif index1 == 6
    comb1 = ’cin1, -qin1, cr1, -qr1’;
elseif index1 == 7
    comb1 = ’cin1, -qin1, -cr1, qr1’;
else
    comb1 = ’cin1, -qin1, -cr1, -qr1’;
end


if index2 == 1
    comb2 = ’cin2, cr2, qr2’;
elseif index2 == 2
    comb2 = ’cin2, cr2, -qr2’;
elseif index2 == 3
    comb2 = ’cin2, -cr2, qr2’;
else
    comb2 = ’cin2, -cr2, -qr2’;
end


if index3 == 1
    comb3 = ’cin3, cr3, qr3’;
elseif index2 == 2
    comb3 = ’cin3, cr3, -qr3’;
elseif index2 == 3
    comb3 = ’cin3, -cr3, qr3’;
else
    comb3 = ’cin3, -cr3, -qr3’;
end


text1 = ’Tank 1: ’;
text2 = ’Tank 2: ’;
text3 = ’Tank 3: ’;


disp(’Maximum allowed dev. from sp, Tank 3: +-1e-06’)

disp(’ ’)

disp(’Selected setpoints:’)
disp([text1, num2str(sp(1))])
disp([text2, num2str(sp(2))])
disp([text3, num2str(sp(3))])

disp(' ')

disp('Calculated max. deviation from setpoint:')
disp([text1, num2str(c1max)])
disp([text2, num2str(c2max)])
disp([text3, num2str(c3max)])

disp(' ')

disp('Worst-case disturbance combinations:')
disp([text1, comb1])
disp([text2, comb2])
disp([text3, comb3])
```

G.2 Calculating SG_d (sgd.m)

```
% This function takes in, for one disturbance at a time, the disturbance gain
% and the sensitivity and tau for the tank, and creates the SGd transfer
% function

function SGd = sgd(kd,sens,tau)

% Defining s as the Laplace variable s
s = tf('s');

 gd = kd/(tau*s+1);           % Disturbance TF
 sgd = sens*gd;               % Output y (outlet concentration)
 [A,B,C,D]=ssdata(sgd);       % Extracting the matrix data from the state-space model
 SGd=tf(ss(A,B,C,D));         % Creating a TF model of the state-space model
```

G.3 Calculating nominal values (nomcalc.m)

```
% This function takes in an array of parameters, calculates the
% nominal values for the 3 tanks: flow rate and concentration of acid, flow
% rate of base, flow rate of outlet stream and residence times. These
% values are returned in arrays collected in a matrix nom.

function nom = nomcalc(setpoints,par_nom)

% Exctracting the parameters:
cin_nom(1) = par_nom(2);
qin_nom(1) = par_nom(3);
cr_nom = par_nom(4);
c_nom(1) = setpoints(1);
c_nom(2) = setpoints(2);
c_nom(3) = setpoints(3);
V = par_nom(1);

% Calculating nominal values and for Tank 1: flow rate of base, outlet flow
% rate, and the residence time tau.
    qr_nom(1) = (cin_nom(1) - c_nom(1))/(c_nom(1)-cr_nom)*qin_nom(1);

    q_nom(1) = qin_nom(1)+qr_nom(1); %Outlet flow rate=acid+base flow rates

    tau(1) = V/q_nom(1);

% Calculating nominal values for Tank 2 and 3: flow rate of base and acid,
% acid concentration and outlet flow rate, and the residence time.
for  i=2:3
    cin_nom(i) = c_nom(i-1);    %Acid conc.=conc. of outlet stream from prev.tank
    qin_nom(i) = q_nom(i-1);    %Acid flow=outlet flow rate from prev.tank

    qr_nom(i) = (cin_nom(i) - c_nom(i))/(c_nom(i)-cr_nom)*qin_nom(i);

    q_nom(i) = qin_nom(i)+qr_nom(i);%Outlet flow rate=acid+base flow rates

    tau(i) = V/q_nom(i);

end

% Creating a matrix nom with the arrays of nominal values.
nom = [qr_nom ; q_nom ; cin_nom ; qin_nom ; tau];
```

G.4 Calculation process and disturbance gains (kcalc.m)

```
% This function takes in an array of parameters
% nomval=[qr_nom, q_nom, cin_nom, qin_nom, c_nom],
% cr_nom and the setpoints.
%
% It calculates the unscaled steady-state gains for different disturbances
% and the control input gains for each tank, which is returned in the
% matrix kval.

function kval = kcalc(nomval,cr_nom,sp)



% Nominal values:
qr_nom = nomval(1,:);
q_nom = nomval(2,:);
cin_nom = nomval(3,:);
qin_nom = nomval(4,:);
c_nom = [sp(1) sp(2) sp(3)];



% Calculating steady-state gains for different disturbances and the gain
% from the control input for Tank 1-3, collecting the values in 1 by 3
% arrays:

for i=1:3

kd_cin(i) = qin_nom(i)/q_nom(i);               % When d = c_in
kd_qin(i) = (cin_nom(i)-c_nom(i))/q_nom(i);    % When d = q_in
kd_cr(i) = qr_nom(i)/q_nom(i);                 % When d = c_r
kd_qr(i) = (cr_nom-c_nom(i))/q_nom(i);         % When d = q_r

k(i) = (cr_nom-c_nom(i))/q_nom(i);             % Controller

end

k_tank1 = [kd_cin(1)  kd_qin(1)  kd_cr(1)  kd_qr(1)  k(1)];
k_tank2 = [kd_cin(2)  kd_qin(2)  kd_cr(2)  kd_qr(2)  k(2)];
k_tank3 = [kd_cin(3)  kd_qin(3)  kd_cr(3)  kd_qr(3)  k(3)];


% Creating a matrix kval with the vectors of disturbance gains, control
% input gains and tau for each tank. This is what the function returns.
```

```
kval = [k_tank1 ; k_tank2 ; k_tank3];
```

G.5 Nonlinear Matlab model

```
function xprime=ph_model(t,X,U)
%
% ph_model - This is a model of a single pH neutralization tank, where an
%            acid (influent) is mixed with a base (reagent) to yield a
%            product with pH = 7. Concentrations are in terms of excess H+,
%            e.g. c = c_H+ - c_OH-
%
% Inputs:    t      - Time [s].
%            X      - States, X(1) = c, X(2) = V
%            U(1)   - reagent flow, qr [l/s]
%            U(2)   - Excess H+ in reagent, cr [mol/l]
%            U(3)   - Influent flow rate, qi [l/s]
%            U(4)   - Excess H+ in influent, ci [mol/l]
%
% Outputs:   xprime - vector with time derivative of the states
%                     (states: X(1) = c, X(2) = V)




% Splitting the states
c=X(1);                              % Excess H+ in effluent
V=X(2);                              % Volume of tank

% One manipulated input:
qr = U(1);                            % Reagent flow rate
q = U(2);

% Three disturbances:
cr = U(3);                            % Excess H+ in reagent, cr [mol/l]
qi = U(4);                            % Influent flow rate
ci = U(5);                            % Excess H+ in influent, ci [mol/l]


% Total mass balance:
```

```
dVdt = qi + qr - q;


% Component mass balance (in terms of excess H+):
%dcVdt = ci*qi + cr*qr - c*q;

dcdt = (1/V)*((ci-c)*qi + (cr-c)*qr);


% Compute the derivative for the concentration from d(cV) = c dV + V dc
%dcdt = (dcVdt - c*dVdt)/V;

% Output
xprime = [dcdt; dVdt];
```

G.6 Simulink interface to the nonlinear model

This is the Simulink interface to the nonlinear tank model in Appendix G.5. It is important to note that the initial value of c have to be changed to the actual setpoint for the tank, such that $x0(1) = c_{sp}$.

```
function [sys,x0] = tank1(t,x,u,flag)
%
% Simulink interface to ph_model.m
%
% Inputs:    t    - time [s]
%            X    - State, X(1) = c, X(2) = V
%
%            U(1) - reagent flow, qr [l/s]
%            U(2) - Excess H+ in reagent, cr [mol/l]
%            U(3) - Influent flow rate, qi [l/s]
%            U(4) - Excess H+ in influent, ci [mol/l]
%
% Outputs:   x0:    Column vector of initial conditions
%            sys:   The main vector of results requested by Simulink.
%                   Will hold different information, depending on the flag.
%
%                   When flag=0:
%                       x0 contains initial conditions
%                       sys = [a, b, c, d, e, f, g]
%                       a = # of continuous time states
%                       b = # of discrete time states
%                       c = # of outputs
%                       d = # of inputs
%                       e = required to be 0
%                       f = 0(no) or 1(yes) for direct algebraic feed
%                           through of input to output.
%                       g = # of sample times (for continuous process, we
%                           set this equal to 1)
%
%                   When flag = 1:
%                       sys = column vector of the derivatives
%
%                   When flag = 3:
%                       sys  = column vector of the output variables:
%                       y(1): effluent concentration c in terms of excess H+
%                       y(2): tank volume V
%
%                   Else: sys is a null vector


if abs(flag) == 1
  % Return state derivatives.
  sys = ph_model(t,x,u);

elseif abs(flag) == 3
  % Return system outputs.
  sys(1,1) = x(1);          % Outlet concentration, c [mol/l]
  sys(2,1) = x(2);          % Tank volume, V [l]

elseif flag == 0
  % Initialize the system
  x0 = [ 0.05 ; 1000*2*40.7/3];         % col.vector of init.values of c and V
  sys = [2, 0, 2, 5, 0, 0];

else
  sys = [];

end
```

H Simulink models

H.1 Linear Simulink model

For every case and tank, the following modifications of the linear Simulink model (Fig. 3.1) has to be done:

1. Disturbance gains (k_d), process gain (k) and τ.
2. $c_{1,nom}$ = setpoint
3. $q_{r,nom}$
4. Disturbance in q_r
5. Lower limit of saturation block in PI controller $= -q_{r,nom}$

H.2 Nonlinear Simulink model

For every case and tank, the following modifications of the nonlinear Simulink model (Fig. 3.2) has to be done:

1. Process model in Smith predictor (k and τ).
2. $c_{1,nom}$ = setpoint
3. $q_{r,nom}$
4. Disturbance in q_r
5. Lower limit of saturation block in PI controller $= -q_{r,nom}$
6. $q_{in,nom}$
7. $c_{in,nom}$ = setpoint from previous tank
8. Initial value of transport delay $= c_{sp}$
9. In the tank model in Matlab (tank1.m/tank2.m/tank3.m), the initial value of c must be changed: $x0(1) = c_{sp}$