NTNU
Norwegian University of
Science and Technology

Model predictive control of a Kaibel distillation column

Martin Krister Kvernland

Master of Science in Engineering Cybernetics

Submission date:	July 2009
Supervisor:	Ole Morten Aamo, ITK
Co-supervisor:	Ivar Halvorsen, SINTEF
	Sigurd Skogestad, IKP

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Background:

A Kaibel distillation column separates a feed into four products with significant lower energy consumption than a conventional sequence of binary columns. Optimal operation and control of such systems is an important task in order to obtain the potential energy savings.

A laboratory column has been built at NTNU, Department of Chemical Engineering. At the time of the diploma work the laboratory column has unfortunately not been available for MPC experiments.

In practical operation a control structure based on temperature measurements is chosen for the given case. This structure gives a four-by-four multivariable system. The candidate shall base his work on a model developed by Jens Strandberg.

Tasks:
1. Describe the model and extend it to include an efficiency parameter describing insufficient mixing at stages
2. Describe a general linear MPC approach for the system
3. Analyze sensitivity of model errors
4. Evaluate alternative MPC approaches
5. Implement the MPC in MATLAB and illustrate the performance by simulations
6. Prepare a setup for connecting the MPC to the actual laboratory column

Assignment given: 12. January 2009
Supervisor: Ole Morten Aamo, ITK

”I’ll tell you this –
No eternal reward will forgive us now for wasting the dawn.”

Jim Morrison

Abstract

Model predictive control (MPC) of a Kaibel distillation column is the main focus of this thesis. A model description together with a model extension is also considered.

The motivation of using a Kaibel distillation column is primarily its energy saving potential. There is no reason for using such an energy saving column if the product purities are below some acceptable values. These purities can be kept above these acceptable values by sufficient control of the column.

The simulation model of the Kaibel column was extended to include an efficiency parameter which describes an insufficient vapor mixing effect that occurs in distillation columns. This insufficient mixing leads to increased impurity flows in open-loop, but this is counteracted by increased reflux flow in closed-loop. The column can obtain sufficient purities for increasing insufficient mixing until the reflux flow reaches its maximum value.

A single layer MPC and a supervisory MPC approach have been described, implemented and tested on the simulation model of the column. These MPCs show improved dynamic responses compared to the existing decentralized control approach. These MPCs have also been compared in a sensitivity analysis part. The sensitivity analysis shows a clear improvement in the robustness properties for a supervisory MPC compared to decentralized control in terms of input uncertainty. The supervisory MPC has also shown to be more robust than the single layer MPC.

A brief qualitative discussion regarding alternative MPC approaches has been done, where inferential control is suggested as an alternative MPC approach for the Kaibel column.

An MPC implementation has been done for use at the existing laboratory column at Department of Chemical Engineering. Also here, based on implementation issues, the supervisory MPC is preferable compared to a single layer MPC. A parameter adjusted version of the developed simulation model is recommended as the MPC's internal prediction model.

PREFACE

This is the master's thesis describing my work in the last semester at the Norwegian University of Science and Technology (NTNU). The work is carried out at Department of Engineering Cybernetics. The assignment is given by Department of Engineering Cybernetics in cooperation with Department of Chemical Engineering and SINTEF.

Ivar Halvorsen at SINTEF has been very helpful during my work. He has given me many useful comments regarding my work and has always been available for a meeting. Sigurd Skogestad also gave me some useful theoretical tips during my work. Thanks also to Terje Mugaas at SINTEF for giving me some introduction to LabVIEW and OPC software.

Finally, I want to thank the guys at the office; Bjarne Grimstad, Henrik Tuvnes, Jørgen Syvertsen and Morten Bakke for useful discussions regarding the diploma work and for having a great time together.

Martin Kvernland
July 2009

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DMC	Dynamic matrix control / controller
EMPC	Explicit model predictive control / controller
LQG	Linear quadratic gaussian regulator
LQR	Linear quadratic regulator
MPC	Model predictive control / controller
NMPC	Nonlinear model predictive control / controller
OPC	Object linking and embedding for process control
PI(D)	Proportional, integral (and dertivative)
QDMC	Quadratic dynamic matrix control / controller
QP	Quadratic programming
SIMC	Simple / Skogestad internal model control (tuning rules)
VLE	Vapor-liquid equilibrium

Chapter 1

Introduction

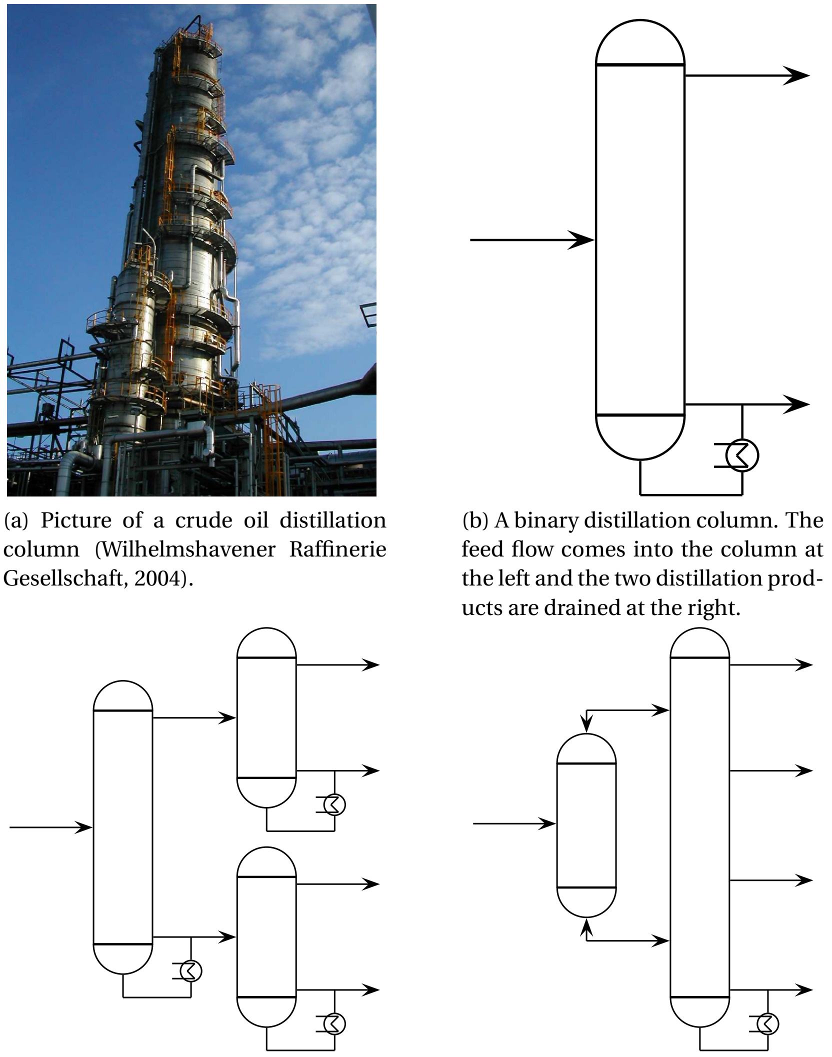
Distillation is the most widely used fluid separation technology in the process industry today. The distillation process is a highly energy-consuming process — enormous amounts of heat is required to separate e.g. crude oil into different fuels in an oil refinery (see Figure 1.1a). This gives extra motivation for new ideas that can reduce the energy-usage of distillation columns.

1.1 Motivation

The worlds focus on reducing CO_2 emissions has increased in the latest years. The industry stands for a large amount of these emissions; therefore only small energy savings in industry can make a big difference to the total emission. As mentioned above, distillation arrangements consume a lot of energy. E.g. distillation columns alone in the US consume 3% of the total energy consumption in the country (Ognisty, 1995).

The *Kaibel distillation column* was introduced in 1987 (Kaibel, 1987) in order to make distillation in the process industry more energy efficient. Halvorsen & Skogestad (2006) presents a theoretical example that compares the energy usage of a Kaibel distillation column with a conventional column setup. The energy calculation shows an energy reduction of 33% for the Kaibel column. According to Ž. Olujić et al. (2009) such a column arrangement generally reduces the energy consumption by around 30% compared with a conventional setup.

There are also other advantages of using a Kaibel distillation column than the energy savings. The distillation products can be separated in a single column shell, which leads to less investment costs compared to a conventional setup. Since it can consist of a single column shell, the space required in a process plant is also smaller.



(a) Picture of a crude oil distillation column (Wilhelmshavener Raffinerie Gesellschaft, 2004).

(b) A binary distillation column. The feed flow comes into the column at the left and the two distillation products are drained at the right.

(c) A conventional column setup when separating four products. This setup requires three reboilers.

(d) Kaibel column setup. This setup require only one reboiler.

Figure 1.1: Different distillation columns.

To summarize; the motivating factors for using a Kaibel distillation column:

- Less energy consumption
- Less investment costs
- Less physical space required in a process plant

1.2 The Kaibel distillation column

The Kaibel distillation column is a *thermally coupled column* which separates a feed flow into *four* distillation products in the same column shell.

Let us first explain the difference between the arrangement that is referred to as a *conventional setup* and a Kaibel column. A conventional setup that separates a feed into four products consists of three binary columns. A binary column (shown in Figure 1.1b is the most used distillation column around the world. When binary columns are used for separation of four products, the column arrangement can be like shown in Figure 1.1c, which is referred to as a conventional setup. More generally, if you want to separate n products using a conventional setup, you will need $n-1$ binary columns.

The Kaibel column is shown in Figure 1.1d. In this setup the binary columns are thermally connected with each other such that they exchange heat and become more energy efficient. As we can see in Figure 1.1c and 1.1d, a conventional setup would require three reboilers in the bottom (necessary to perform the distillation process), and the Kaibel column only require a single reboiler.

1.3 Model predictive control

One of the main challenges that come with use of the Kaibel distillation column is in the field of *control*. The energy savings are not achieved if the distillation column does not operate around its optimal operating points. A common objective where these optimal operating points are resolved from for a distillation column is to maximize the purities, or equivalently, *minimize the impurities*. This objective is used in this thesis.

The column is a multivariable process that gives motivation for use of a *model predictive controller* (MPC). An MPC finds the optimal inputs to the process by use of mathematical optimization based on a mathematical model of the process. The greatest number of MPC applications can be found in petrochemical industries which generally have shown to be very successful (Camacho & Bordons, 2004).

To clarify the MPC strategy, an MPC analogy taken from Camacho & Bordons (2004) is presented in Figure 1.2. Suppose the driver of the car uses conventional PID feedback control for driving, hence the driver's control actions are based on past errors, i.e. the driver can only see in the mirror. An MPC predicts future outputs for a finite horizon N into the future, based on these predicted outputs it calculates the optimal inputs for N time steps ahead such that the output can follow a specified trajectory. Thus, the driver can now see through the front window to control the car.

The cars condition determines its constraints, e.g. maximum speed, acceleration, brakes, steering characteristics etc. An MPC would take these constraints into consideration when it computes the optimal inputs.

It is important to have a prediction model that is close or equal to the physical

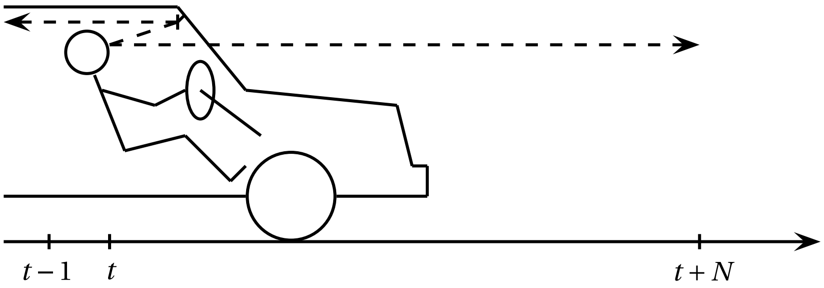


Figure 1.2: MPC analogy (Camacho & Bordons, 2004). The MPC predicts the process' outputs N time steps into the future and calculates the optimal inputs such that the process follows a specified trajectory.

model in order to make the control optimal. To continue on the presented MPC analogy, suppose the driver thinks he drives a Porsche, but in reality he is driving a Lada. When he brakes down before a sharp turn, he overestimates the Ladas breaking qualities because he thinks he has the brakes of a Porsche, which causes a crash. The same thing (not a car crash...) can happen in a real process plant if the prediction model is to poor.

1.4 Project scope

Therefore, an MPC require a good mathematical model of the process in order to predict the future outputs. As a part of earlier work a mathematical model has been made for computer simulations of the Kaibel distillation column. It is desirable to make this model be as close to the real physical process as possible, hence we want to extend the model to include physical effects that occurs in an actual implementation of the distillation column.

The problem description of the thesis is to do the following tasks:

- Extend the existing model to include an efficiency parameter that describes insufficient mixing at column stages.
- Describe a general linear MPC approach for the process.
- Implement the MPC in MATLAB and illustrate the performance by simulations.
- Analyze sensitivity of model errors.
- Evaluate alternative MPC approaches.
- Prepare a setup for connecting the MPC to an actual laboratory column.

1.5 Thesis outline

Some background information is presented in the Chapter 2 to clarify what work that has been done earlier regarding the Kaibel column. The rest of the thesis can roughly be divided into four main parts:

- **Theory:** Introduction to distillation and the Kaibel distillation column.
- **Problem statement:** A deeper presentation of the problem description, which is meant as a ”bridge” between the introductory part and the main work.
- **Modeling:** Extension of the existing distillation model.
- **Control:** Model predictive control of the distillation column.

Chapter 3 starts with the part on the theory of distillation, which presents the main principles in distillation together with the most important equations. The Kaibel distillation column model is explained next, followed by a section which explains how the column has been controlled in earlier work.

Chapter 4 contains the problem statement itself in detail, and more motivation for why it is important to investigate the presented problems.

Presentation of the main work starts with Chapter 5 which is about the model extension part of the work. Chapter 6 provides an introduction to model predictive control. This chapter presents the first optimal controller, the linear quadratic regulator, then the first model predictive controller, the dynamic matrix controller, and thereafter the state-space based MPC.

Implementation and simulation of the MPC is presented in Chapter 7. The sensitivity analysis is also given in this chapter. A brief discussion on alternative MPC approaches is given in Chapter 8.

Chapter 9 deals with an MPC implementation on a laboratory distillation column.

Chapter 10 contains a discussion on the different results obtained in the work. The final chapter, Chapter 11, provides the concluding remarks, contributions and suggestions for further work.

1.6 Source code

The ZIP-file attached to this report contains the source code that has been used in simulations. All code from earlier work has been re-written to make it more understandable to the author and for those who want to use this code later.

The ZIP-file also contains files that have been used for a practical MPC implementation on the laboratory column.

CHAPTER 2

BACKGROUND

2.1 Research at BASF

BASF[1] is a German company that is the largest chemical company in the world. It was Dr. Gerd Kaibel at BASF that in 1987 introduced the Kaibel distillation column (Kaibel, 1987). The Kaibel column is developed from the dividing wall column (DWC) (Wright, 1949) and the Petlyuk column (Petlyuk et al., 1965). A dividing wall column and a Petlyuk column uses the principle of thermally coupled distillation columns in order to save energy. The Kaibel distillation column is able to separate a mixture into four products in the same column shell, in difference to a corresponding Petlyuk arrangement that would be much more complex in its structure and more difficult to control (Strandberg & Skogestad, 2006).

Ž. Olujić et al. (2009) reports that there are over 70 DWCs operated by BASF worldwide. In 2004 Kaibel et al. (2004) reported the first Kaibel column in operation.

2.2 Work by PhD fellow Jens Strandberg

PhD fellow Jens Strandberg at Department of Chemical Engineering is currently writing a doctoral thesis with the title ”Optimal operation of Petlyuk and Kaibel columns” (Strandberg, 2009). The motivation for his work is to gain deeper understanding of the operation of *Petlyuk and Kaibel* column arrangements because of their energy saving potentials. The control strategy that has been used in his work is *self-optimizing control*, which has the following definition (Skogestad & Postlethwaite, 2005):

[1] BASF originally stood for ”Badische Anilin- und Soda-Fabrik”.

> *Self-optimizing control is when we can achieve an acceptable loss with constant setpoints values for the controlled variables without the need to reoptimize when disturbances occurs.*

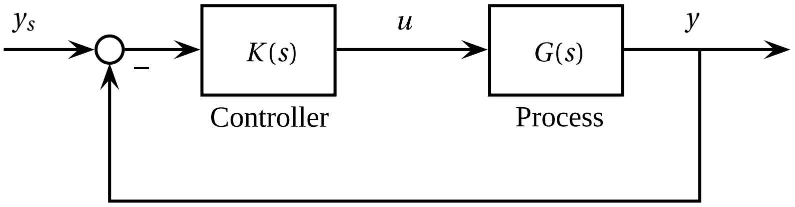
In the work of Strandberg (2009) optimal operating points are found by mathematical optimization. These operating points are also used in this thesis. There are these operating points that are used by the self-optimizing control.

Strandberg has used a *decentralized control structure* in his work which is the simplest way to design a multivariable controller. Suppose the multivariable plant is a linear process described by a transfer function matrix $G(s)$ with input u and output y. A decentralized controller is described by a transfer function matrix $K(s)$ with diagonal elements only;

$$K(s) = \begin{bmatrix} k_{11}(s) & & & \\ & \ddots & & \\ & & k_{ii}(s) & \\ & & & \ddots \end{bmatrix}. \tag{2.1}$$

The diagonal elements $k_{ii}(s)$ can for example be PI controllers[2]. The open loop transfer function matrix will be $G(s)K(s)$. Decentralized control will only work good if the process $G(s)$ is close to diagonal, i.e. it has little *interaction* (Skogestad & Postlethwaite, 2005). It is not required that the process is linear.

Suppose the controlled process has a reference setpoint y_s which can be such that the process is operated optimal when this reference is maintained. The block diagram of this control configuration can be seen in the following figure:



Among the things that Strandberg have worked with is the MATLAB / Simulink model of a Kaibel distillation column. This simulation model is further developed and used in the work of this thesis.

Pilot plant at Department of Chemical Engineering

PhD fellow Jens Strandberg has built a laboratory pilot plant in the Experimental Hall at Department of Chemical Engineering in order to do real experiments on

[2] PI controllers are commonly used in process control, its transfer function is $K_{\text{PI}}(s) = K_c\left(1+\frac{1}{\tau_I s}\right)$ (parallel form). Suitable values for K_c and τ_I can be found using the Simple / Skogestad internal model control (SIMC) tuning rules (Skogestad, 2003b).

operation and control of a Kaibel distillation column. This laboratory column is fed with the alcohols *butanol*, *ethanol*, *methanol* and *propanol*. The column is controlled by holding some temperatures constant; therefore it has totally 24 temperature sensors installed along the about 8 m high column. The process inputs are physical valves on the column that can be controlled by pulse-width modulation. The column is operated by a computer interface which is connected to an I/O system from National Instruments (Strandberg, 2009). Additional information on this column is presented later in Chapter 9.

2.3 Final year project work fall 2008

The author wrote a final year project report with the title "Control and modeling of Kaibel distillation column" (Kvernland, 2008). The simulation model was in this work extended to include heat loss in order to make the model more physical. The report contained a short discussion on the issue of adjusting the model parameters to fit simulation data with real laboratory measurements. Including the heat loss and the discussion on the system identification issue was the modeling part of this project work.

In the control part of the project work, decentralized control was presented together with model predictive control (MPC). The decentralized controller consisted of several PI controllers controlling an output variable each, i.e. the transfer functions $k_{ii}(s)$ were transfer functions of PI controllers. Since decentralized control requires the process to have little interaction some interaction analysis was done in the project work. The decentralized controller was tuned in work using Simple / Skogestad internal model control (SIMC) tuning rules (Skogestad, 2003b).

The MPC-part of the project was however limited to simulation use only because the disturbances that occurred during the simulations were assumed to be known. In a real control implementation this approach would of course not be sufficiently. A linear MPC was used and this required linearization of the initially nonlinear distillation model. To simplify the optimization problem for the MPC, this linearized model was reduced to a lower order still keeping dynamic behavior for relevant frequencies. Hence, the project work also dealt with model reduction.

Project work conclusions

Some conclusions were made from the project work and computer simulations which are briefly summarized here.

The model extension with heat loss made the top product more pure since less was vaporized upwards in the column, but the bottom product and the side products became less pure.

The discussion on model identification concluded that a further model extension was preferable in order to obtain an easier identification problem. The proposed model extension was an efficiency parameter that describes insufficient vapor mixing in the column, which is one of the main tasks in this thesis.

The interaction analysis that was done did not indicate heavy interactions, which means that decentralized control can be used as a controller for the Kaibel column. The MPC that was made for the simulation model gave good disturbance rejection, but since the disturbances were assumed to be known, it can not be used in practical situations.

CHAPTER 3

INTRODUCTION TO DISTILLATION AND THE KAIBEL DISTILLATION COLUMN

This chapter does the necessary introduction to the theory behind distillation, the Kaibel distillation column model and presents the control approach that has been used in earlier work. Some simulation results are presented to show some dynamic behavior of the controlled distillation column.

The bibliographic source of the distillation theory section is Kvernland (2008). This section is added for the completeness of the thesis.

3.1 Distillation theory

Distillation is a physical process that separates boiling liquid mixtures based on the difference in volatility. Volatility is a measure of the tendency of a substance to vaporize (Chang, 2006). Figure 3.1 shows a binary distillation column with feed input and top and bottom product, D (distillate) and B respectively. A column has two heat exchangers, one *reboiler* at the bottom that generates vapor V going upwards. The other one is a *condenser* at the top that cools vapor and generate top product and reflux L. Reflux is condensated liquid that goes back to the column in the top, and make the top product more pure.

The equilibrium-stage concept

A distillation column is often vertically divided in a number of stages, and at each stage it is assumed vapor-liquid equilibrium (VLE). The general VLE relation is given in the equation below. N_c is the number of components to be separated in the column. $y_1, y_2, \ldots, y_{N_c-1}$ are mole fractions in the vapor phase for $N_c - 1$ components, similar for the liquid phase; $x_1, x_2, \ldots, x_{N_c-1}$. T and P denotes tempera-

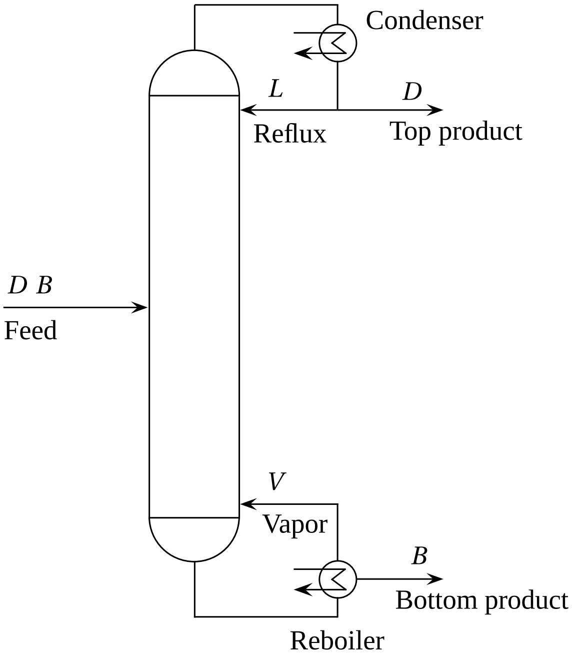


Figure 3.1: Distillation column.

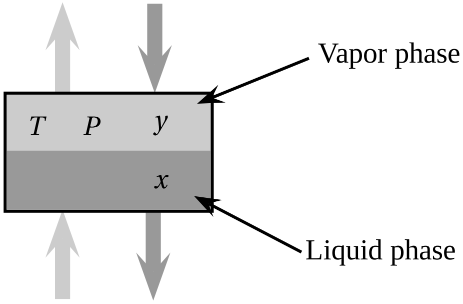


Figure 3.2: Equilibrium-stage concept.

ture and pressure respectively (Halvorsen, 2001). For a Kaibel column, N_c equals four.

$$[y_1, y_2, \dots, y_{N_c-1}, T] = f(P, x_1, x_2, \dots, x_{N_c-1}). \tag{3.1}$$

The concept of equilibrium-stage is shown in Figure 3.2.

Material balance on a stage

It is easy to set up a material (mass) balance for each stage in a distillation column. Let $N_{i,k}$ denote the number of moles of component i on stage k, L_k denotes liquid molar flow from stage k to $k-1$ and V_k is the vapor molar flow from stage k to

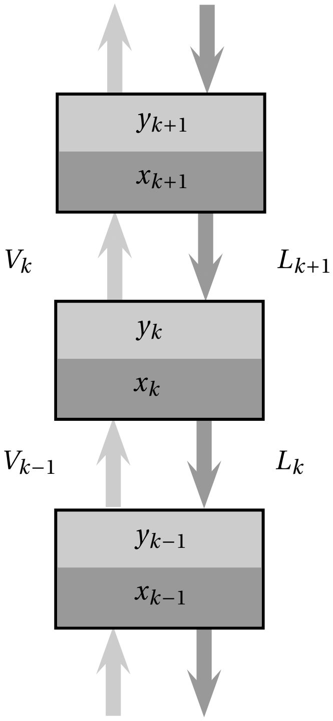


Figure 3.3: Three stages of a distillation column.

$k+1$. This is illustrated in Figure 3.3. The material balance will then be

$$\frac{\mathrm{d}N_{i,k}}{\mathrm{d}t} = (L_{k+1}x_{i,k+1} - V_k y_{i,k}) - (L_k x_{i,k} - V_{k-1} y_{i,k-1}). \tag{3.2}$$

The number of moles of a component on a stage is equal to $M_k x_{i,k}$, where M_k is the total number of moles at stage k. Using this relation a differential equation for the liquid mole fraction can be derived from the material balance,

$$\frac{\mathrm{d}N_{i,k}}{\mathrm{d}t} = \frac{\mathrm{d}(M_k x_{i,k})}{\mathrm{d}t} = M_k \frac{\mathrm{d}x_{i,k}}{\mathrm{d}t} + \frac{\mathrm{d}M_k}{\mathrm{d}t} x_{i,k}. \tag{3.3}$$

The time derivative of M_k is

$$\frac{\mathrm{d}M_k}{\mathrm{d}t} = L_{k+1} - L_k + V_{k-1} - V_k. \tag{3.4}$$

The time derivative of $x_{i,k}$ can then be found,

$$\begin{aligned} \frac{\mathrm{d}x_{i,k}}{\mathrm{d}t} &= \frac{1}{M_k}\left(\frac{\mathrm{d}N_{i,k}}{\mathrm{d}t} - \frac{\mathrm{d}M_k}{\mathrm{d}t} x_{i,k}\right) && \text{(3.5a)} \\ &= \frac{1}{M_k}\left(L_{k+1}(x_{i,k+1} - x_{i,k}) - V_k(y_{i,k} - x_{i,k}) + V_{k-1}(y_{i,k-1} - x_{i,k})\right). && \text{(3.5b)} \end{aligned}$$

3.2 Equations used in distillation

For ideal mixtures *Raoult's law* can be used (given below), where the partial pressure p_i of component i equals the product of the liquid mole fraction x_i and vapor pressure $p_i^0(T)$.

$$p_i = x_i p_i^0(T). \tag{3.6}$$

To determine the vapor pressure $p_i^0(T)$ the logarithmic equation given below is commonly used, where the constants a, b, c, d, e and f are found in chemical handbooks.

$$\ln p^0(T) \approx a + \frac{b}{c+T} + d\ln(T) + eT^f. \tag{3.7}$$

$d = e = 0$ gives the *Antoine equation. Dalton's law* gives the relation between the partial pressure p_i and vapor mole fraction y_i,

$$p_i = y_i P. \tag{3.8}$$

P denotes the total pressure in the vapor phase, the same as in Figure 3.2 (Halvorsen, 2001).

3.3 The Kaibel distillation column

The Kaibel column shown in Figure 3.4 has a separate column first which is the *prefractionator* (left) where the two lightest and the two heaviest products are supposed to be separated. In the *main column* (right) the products are separated further and drained.

The products are called distillate, side stream 1, side stream 2 and bottom product and the flows are denoted D, S_1, S_2 and B respectively. The distillate is drained at the top of the main column. The product *compositions* (mole fractions) are denoted x_D, x_{S_1}, x_{S_2} and x_B respectively.

The input mixture (feed) flow is denoted with F. The feed has a *liquid fraction q* and a composition z_F. The composition z_F is a vector of three elements;

$$z_F = \begin{bmatrix} z_D & z_{S_1} & z_{S_2} \end{bmatrix}^\top. \tag{3.9}$$

The fourth composition can be computed from the three first ones since the composition always sums up to unity.

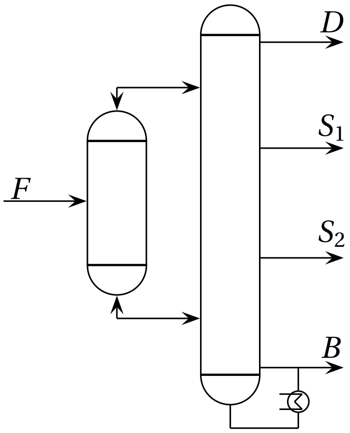


Figure 3.4: The Kaibel distillation column.

Column model

The simulation model has adjustable number of column stages, thus, it is possible to scale it up and down to a desirable size. The simulations performed in this work have used a model where the total number of stages is 64. The size of a physical column which has the equivalent number of stages is larger than the laboratory column at Department of Chemical Engineering. The model's equivalent size is more like an industrial column.

The simulation column is shown in Figure 3.5. It consists of seven *column sections* (numbered) which each consists of a specific number of *stages* (numbered from 1 to 64). The simulation column uses the same alcohols as the laboratory column as its feed, butanol, ethanol, metanol and propanol. The lightest alcohol used is methanol and is drained at the top of the column (D). Side stream 1 (S_1) is ethanol, side stream 2 (S_2) is propanol and the bottom product (B) is butanol. The column has eight degrees of freedom; the four product streams, *reflux*, *vapor boilup* and *liquid and vapor split.* The liquid split (R_L) decide how much liquid that goes into the top of the prefractionator from the main column and similar for the vapor split (R_V), how much vapor that goes into the bottom of the prefractionator from the main column.

3.4 Distillation modeling

The nonlinear model used in MATLAB / Simulink is basically built from three different differential equations. These differential equations are *mass* and *mole fraction* balances, together with an update law for the *temperature* inside the distillation column.

The mass balance was presented in Equation (3.4), and the mole fraction balance was given in Equation (3.5b).

To use the mole fraction balance in the simulation model the vapor mole fraction $y_{i,k}$ needs to be computed. This vapor mole fraction is determined by using an extended version of *Raoult's law* (Skogestad, 2003a);

$$P_k y_{i,k} = x_{i,k}\gamma_{i,k}p^0_{i,k}(T_k). \tag{3.10}$$

$p^0_{i,k}(T_k)$ is the *vapor pressure* and is found using the *Antoine equation* (Equation (3.7)). γ_i is the acitivity coefficient and is found using *Wilson's model* (Gmehling & Onken, 1977). This model uses *Wilson coefficients* to find the activity coefficient, and the Wilson coefficients are found by table look-up for the four alcohols (butanol, ethanol, methanol and propanol) which is used as the feed. Wilson's model is a nonlinear model and therefore the complete distillation model for the Kaibel column is a *nonlinear model.* By summing over all components in Equation (3.10) the pressure at each stage can be calculated,

$$P_k(x_{i,k}, T_k) = \sum_{i=1}^{N_c} x_{i,k}\gamma_{i,k}p^0_{i,k}(T_k). \tag{3.11}$$

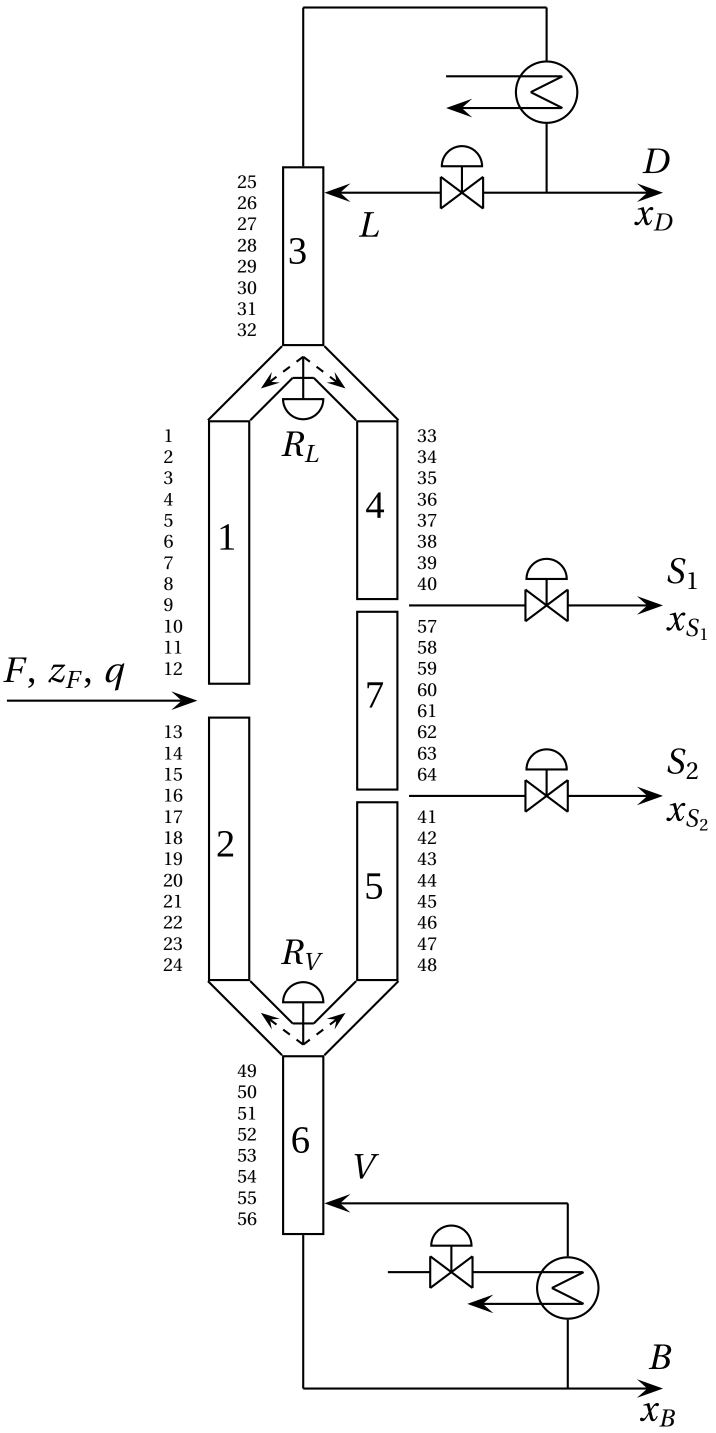


Figure 3.5: The Kaibel distillation column model with stage numbers. The seven column sections are also numbered.

A single simulation step in the simulation model

1	Compute liquid and vapor flows for each stage k in the column, i.e. determine: $L_k, \forall k$, where the liquid input of stage 25 is given (see Figure 3.5) $V_k, \forall k$, where the vapor input of stage 56 is given (see Figure 3.5)
2	Compute change in mass for each stage k using the differential equation for mass balance: $\frac{\mathrm{d}M_k}{\mathrm{d}t} = L_{k+1} - L_k + V_{k-1} - V_k$
3	Compute vapor mole fraction for each component i at each stage k: $y_{i,k} = \frac{x_{i,k}\gamma_{i,k}p^0_{i,k}(T_k)}{P_k}$, where the activity coefficient $\gamma_{i,k}$ is found by Wilson's model and the pressure at stage k is $P_k(x_{i,k}, T_k) = \sum_{i=1}^{N_c} x_{i,k}\gamma_{i,k}p^0_{i,k}(T_k)$ (liquid mole fraction $x_{i,k}$ and temperature T_k used in these equations comes from the previous simulation step)
4	Compute liquid mole fraction for each component i at each stage k: $\frac{\mathrm{d}x_{i,k}}{\mathrm{d}t} = \frac{1}{M_k}\left(L_{k+1}(x_{i,k+1} - x_{i,k}) - V_k(y_{i,k} - x_{i,k}) + V_{k-1}(y_{i,k-1} - x_{i,k})\right)$
5	Update temperature at each stage k: $\frac{\mathrm{d}T_k}{\mathrm{d}t} = \mu\left(P_0 - P(x_{i,k}, T_k)\right)$

Table 3.1: Distillation model summarization.

The third differential equation that completes the distillation model is an update law for the temperature inside the column. The column is modeled with an open vent, which means that the pressure inside the column will tend to the atmospheric pressure P_0. To make sure that the pressure given by Equation (3.11) converge to P_0, the temperature is adjusted. This give rise to the last differential equation that involves the temperature. This is just a numerical update law that does not really have any physical meaning, but it assure that the pressure computed using Raoult's law and Wilson's model converges to the correct pressure P_0. The differential equation is as follows

$$\frac{\mathrm{d}T_k}{\mathrm{d}t} = \mu\left(P_0 - P(x_{i,k}, T_k)\right). \tag{3.12}$$

μ is just a positive constant that decides the speed of convergence.

The simulation model is summarized in Table 3.1.

By using the differential equations presented the states are defined; mass, mole fractions and temperature for each stage in the distillation column. The model has 13 inputs which are shown in Table 3.2. The table also show nominal (steady state) values for each input when the column model is in optimal operation.

Variable	Explanation	Nominal value	Unit
L	Reflux	2.8492	mol/min
V	Vapor boil-up	3.0000	mol/min
S_1	Sidestream 1	0.2494	mol/min
S_2	Sidestream 2	0.2497	mol/min
R_L	Liquid split	0.2572	<ratio>
R_V	Vapor split	0.3770	<ratio>
F	Feed stream	1.0000	mol/min
z_D	Mole fraction of D in feed stream	0.2500	<ratio>
z_{S_1}	Mole fraction of S_1 in feed stream	0.2500	<ratio>
z_{S_2}	Mole fraction of S_2 in feed stream	0.2500	<ratio>
q	Liquid fraction of feed stream	0.9000	<ratio>
D	Top product	0.2508	mol/min
B	Bottom product	0.2503	mol/min

Table 3.2: Inputs for column model, see Figure 3.5 for further explanation. The output flows are actually scaled with respect to the feed, but since the feed equals 1.000 mol/min, the same unit is used for the scaled flows. The table is copied from Kvernland (2008).

Some of the inputs presented in Table 3.2 must be considered as disturbances since they can not be manipulated on a real plant. The inputs that can be manipulated from a controller can then be defined as u and the disturbances as d;

$$u = \begin{bmatrix} L & S_1 & S_2 & R_L & D & B \end{bmatrix}^\top, \tag{3.13a}$$

$$d = \begin{bmatrix} V & R_V & F & z_D & z_{S_1} & z_{S_2} & q \end{bmatrix}^\top. \tag{3.13b}$$

Note that the vapor split (R_V) is considered as a disturbance and not a manipulated input. It was initially thought as a manipulated input, but it has shown to be very difficult to control the vapor flows in practice. This is a challenge where no good solution is found yet, therefore the vapor split is not used as a input from the controller. The vapor boil-up (V) is also a disturbance in this model because it is set to constant during operation of the column. Thus, the vapor boil-up is not manipulated.

The model output can be chosen freely. It is very common in distillation control to measure temperatures along the column, and then use the controller to hold certain temperatures at given setpoints to hold the temperature profile along the column constant. This is the way of thinking that have been used to control the column in previous work (Strandberg & Skogestad, 2006). The temperatures at each stage have good correlation with the respective compositions. The compositions can also be measured by a *gas chromatograph*, but it is very expensive and the measurements will arrive with a considerable time delay. Therefore keeping the temperature profile at an optimal place is preferred in practice.

The full state-space model can be written as

$$\dot{x} = f(x,u,d), \tag{3.14a}$$

$$y = g(x). \tag{3.14b}$$

Note that the measurements (y) depends only on the states (x).

3.5 Linearized and reduced model

The nonlinear model given above in Equation (3.14) can be linearized by perturbing the nonlinear model around the steady state values and approximate the responses to linear responses. This can be done in MATLAB using the command `linmod`. Linearizing the model gives the following linear model

$$\frac{\mathrm{d}}{\mathrm{d}t}\Delta x = A_l \Delta x + B_l \Delta u + B_{l,d}\Delta d, \tag{3.15a}$$

$$\Delta y = C_l \Delta x. \tag{3.15b}$$

The Δ-variables are deviations from the steady state values.

When the Kaibel distillation model has 64 stages, the number of states becomes 341, which is a quite large state-space system. Kvernland (2008) showed that the model can be reduced significant without affecting the dynamics behavior for relevant frequencies to much. The model reduction method itself was presented in Kvernland (2008) and is therefore not presented in greater depth here. Model reduction make the optimal control problem solved in an MPC much easier and less time-consuming.

The model reduction algorithm is performed on the linearized model given by Equation (3.15) and together with a specified order (number of states in the reduced model) it creates a reduced model,

$$\frac{\mathrm{d}}{\mathrm{d}t} = A_r \Delta x_r + B_r \Delta u + B_{r,d}\Delta d, \tag{3.16a}$$

$$\Delta y = C_r \Delta x_r. \tag{3.16b}$$

Subscript r denotes the reduced constants and variables. This reduced model can be discretized (the Δ-notation is now dropped),

$$x_{k+1} = A x_k + B u_k + B_d d_k, \tag{3.17a}$$

$$y_k = C x_k. \tag{3.17b}$$

This discretized model can now be used in an MPC implementation.

3.6 Control of the distillation column

As mentioned earlier it is most common that distillation columns are controlled by use of temperatures as measurements. The temperature at a stage in a distillation column is a good indication of its composition. Skogestad (2007) presents some benefits of using temperature loops for controlling the composition:

1. *Stabilizes the column composition profile along the column*
2. *Gives indirect level control: Reduces the need of level control*
3. *Gives indirect composition control: Strongly reduces disturbance sensitivity*
4. *Makes the remaining composition problem less interactive and thus makes it possible to have good two-point composition control*
5. *Makes the column behave more linearly*

According to Skogestad (2007) temperature control is a simple way of stabilizing the column profile, i.e. in terms of composition.

The modeled column has initially eight degrees of freedom, but as we saw in one of the previous sections vapor boil-up and the vapor split were not considered as manipulated inputs, and we are left with six degrees of freedom for control of the column. These degrees of freedom are the reflux, liquid split and the four product flows.

Optimal operation

In earlier work (Strandberg & Skogestad, 2006), these six degrees of freedom are manipulated by six different PI controllers. Two level controllers are used for controlling the level in the condenser and reboiler. These levels are just kept constant by controlling the product flows in the top (D) and bottom (B). The other manipulated inputs (L, R_L, S_1 and S_2) are used for *temperature control*. Four temperatures are here chosen to be controlled to specified setpoints that automatically leads to *optimal steady state* column operation, that means they minimize a specified objective function. It is important to notify that these setpoints are predetermined using offline optimization beforehand. The objective function used to obtain these setpoints weighs the *total impurity flow*;

$$J = D(1 - x_D) + S_1(1 - x_{S_1}) + S_2(1 - x_{S_2}) + B(1 - x_B). \tag{3.18}$$

Figure 3.6 shows how the Kaibel column is controlled in this work. PI controllers are used for level control of condenser and reboiler in top and bottom of the column respectively. Some results from the simple approach of using PI controllers for controlling the compositions are presented in this chapter, but the main focus in this thesis is the use of MPC for composition control. An MPC can also be putted on top of the PI controllers, controlling their setpoints.

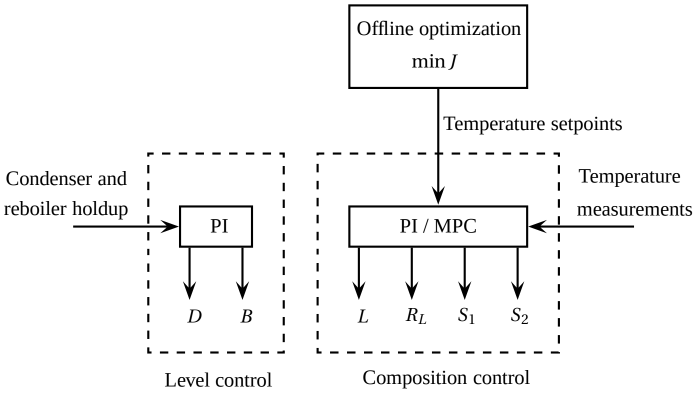


Figure 3.6: Control structure of the Kaibel distillation column.

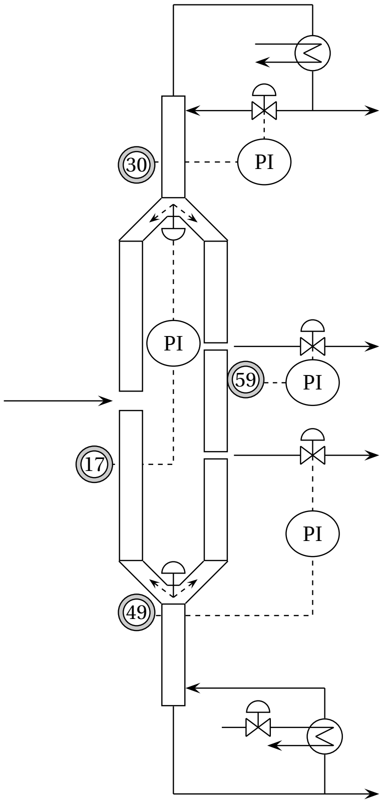


Figure 3.7: Decentralized control of the Kaibel column. The figure shows the four temperature loops.

Disturbance variable	Change from nominal value
Feed flow (F)	+ 10 %
Ethanol composition in feed (z_{S_1})	+ 20 %
Vapor split (R_V)	+ 10 % and + 50 %

Table 3.3: Disturbance tests (Kvernland, 2008).

In Kvernland (2008) the temperatures controlled were T_{17}, T_{30}, T_{59} and T_{49} (the subscript indicates which stage the temperature measurement is). These stage temperatures were controlled by a PI controller each which were tuned using Simple / Skogestad internal model control (SIMC) tuning rules (Skogestad, 2003b).

The reflux (L) is the manipulated variable for T_{30}, side stream 1 (S_1) for T_{59}, side stream 2 (S_2) for T_{49} and the liquid split (R_L) for T_{17} (Strandberg & Skogestad, 2006). The temperature setpoints are

$$y_{\text{ref}} = \begin{bmatrix} T_{17,\text{ref}} \\ T_{30,\text{ref}} \\ T_{59,\text{ref}} \\ T_{49,\text{ref}} \end{bmatrix} = \begin{bmatrix} 368.29\,\text{K} \\ 341.97\,\text{K} \\ 355.83\,\text{K} \\ 379.64\,\text{K} \end{bmatrix}. \tag{3.19}$$

This control strategy by using several PI controllers on a multivariable plant is called *decentralized control*, as we introduced already in Chapter 2. The decentralized control structure is shown in Figure 3.7 for the Kaibel column model.

If the model simulates using the nominal values for inputs as given in Table 3.2 the nominal values for the four product compositions will be

$$\begin{bmatrix} x_D \\ x_{S_1} \\ x_{S_2} \\ x_B \end{bmatrix} = \begin{bmatrix} 0.9703 \\ 0.9361 \\ 0.9589 \\ 0.9949 \end{bmatrix}. \tag{3.20}$$

Disturbance tests

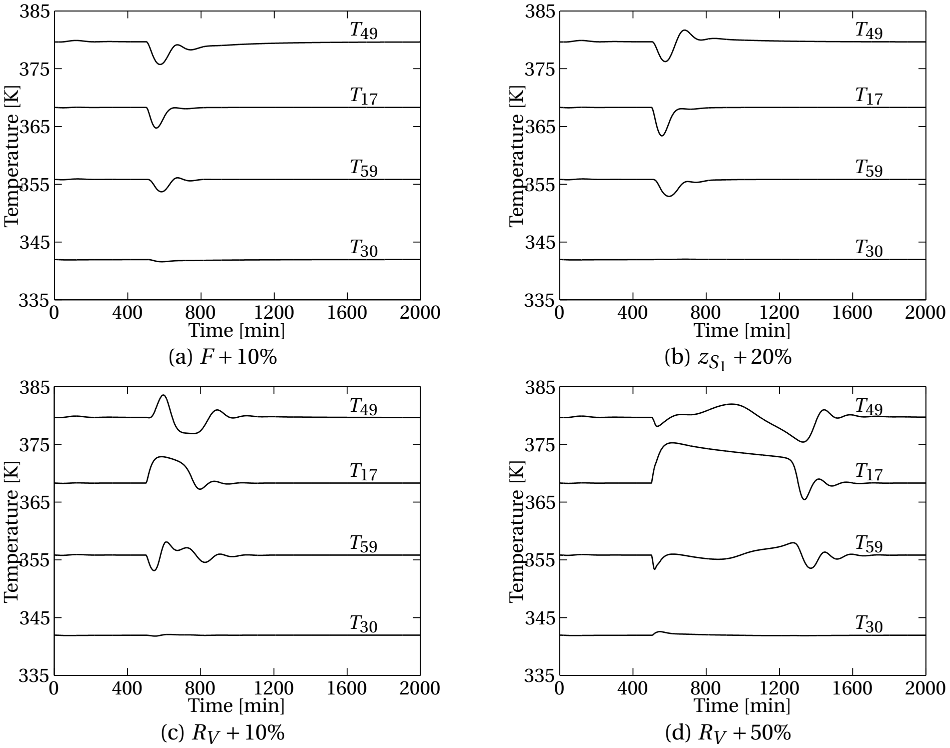
Different disturbance tests are performed to test the control of the column. These tests are shown in Table 3.3. It is important that the controlled column is able to handle changes in the feed flow since this flow determined by other parts of the process plant where this column is in operation. It is very unlikely that the feed is constant all the time. The feed composition also changes and should be tested in the simulations.

The vapor split is a degree of freedom with great uncertainty and is therefore included in the disturbance tests. The uncertainty arises from the difficulty of controlling the vapor; it is not equally of controlling liquid since the vapor behavior is much more complex.

Responses using decentralized control on the distillation model

To test the decentralized control strategy the disturbances shown in Table 3.3 are applied at time 500 min on the simulation model in MATLAB / Simulink.

Figure 3.8 show how the controlled temperatures is affected by the disturbances and how well the controllers manages to control them to their respective setpoints. Figure 3.9 and 3.10 show the product compositions and the total impurity flow respectively. From the product composition plots it is seen that the products goes to around their initial purity some time after the occuring disturbance. The reason for some less purity after the disturbance is that the column is temperature controlled; *the pre-computed optimal temperature setpoints are changed some after the constant disturbance has occured.* This is verified by the rise of the total impurity flow in Figure 3.10b and 3.10d. The feed increase disturbance (Figure 3.10a) has a natural rise in the total impurity flow because of increased product flows.



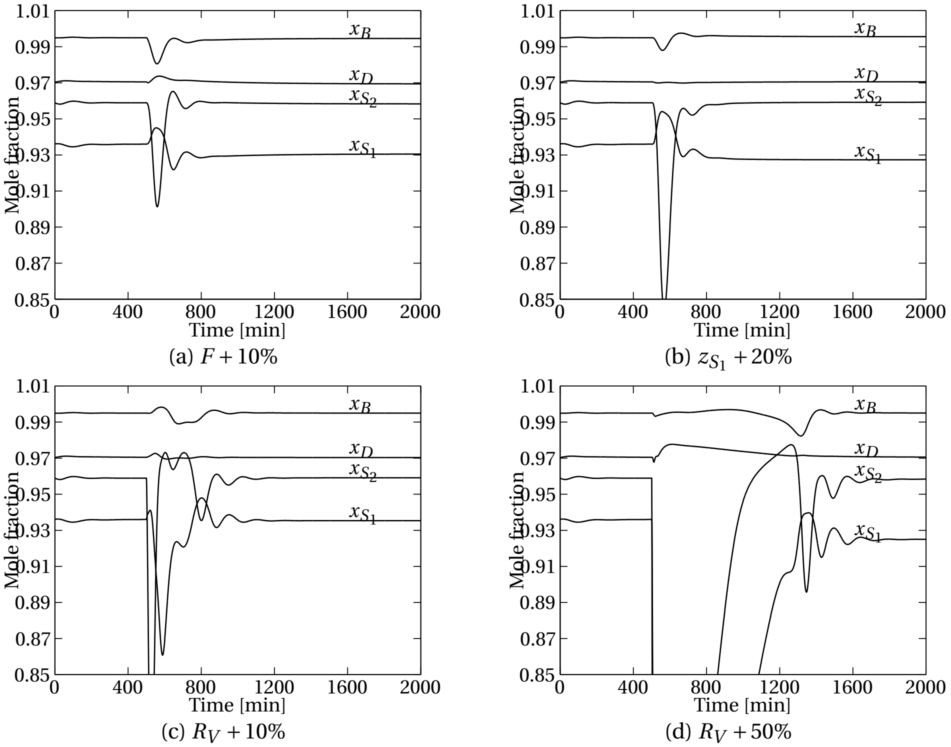
(a) $F + 10\%$

(b) $z_{S_1} + 20\%$

(c) $R_V + 10\%$

(d) $R_V + 50\%$

Figure 3.8: Disturbances affecting the controlled temperatures.



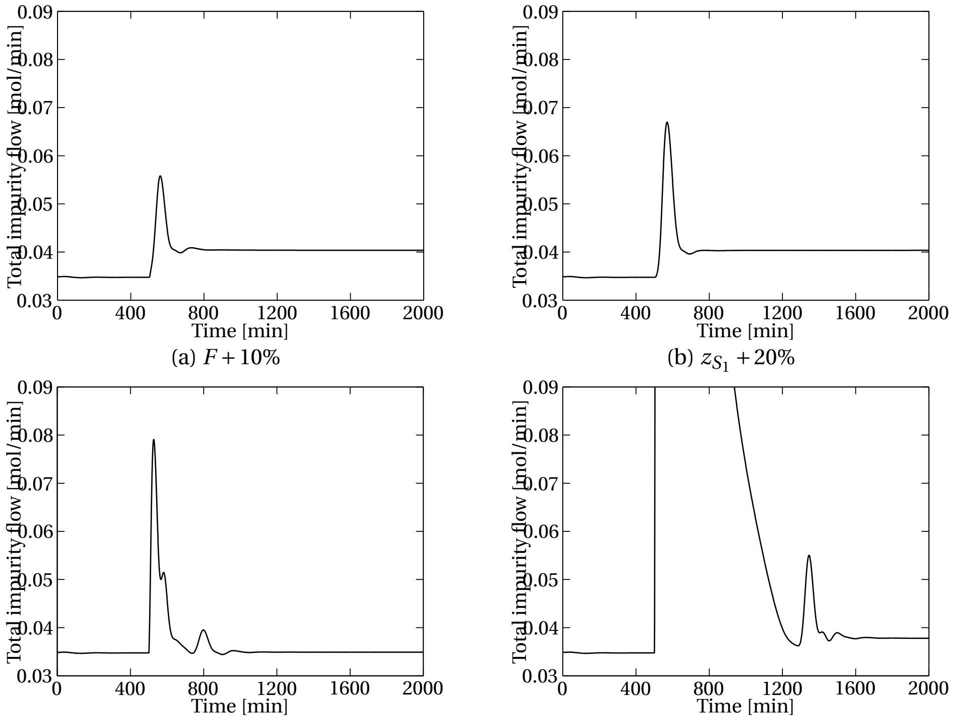
(a) $F + 10\%$

(b) $z_{S_1} + 20\%$

(c) $R_V + 10\%$

(d) $R_V + 50\%$

Figure 3.9: Disturbances affecting the product compositions.



(a) $F + 10\%$

(b) $z_{S_1} + 20\%$

(c) $R_V + 10\%$

(d) $R_V + 50\%$

Figure 3.10: Disturbances affecting the total impurity flow, which is the objective function for finding optimal setpoints. The plot in Figure 3.10d does not show the maximum value because of the chosen axes. However, the maximum value is 0.3587 mol/min at $t = 517$ min, which shows that such a disturbance is quite large relative to the other disturbances tested.

CHAPTER 4

PROBLEM STATEMENT

One of the main tasks is to improve the simulation model to make it more equal to a physical implementation of the process. It is mainly two reasons for improving the simulation model. Most relevant for this thesis is that an MPC needs a correct model to predict future outputs for optimal control. The other reason is to obtain a more precise model in order to do simulations instead of building expensive test plants. Even though a laboratory column has been built at Department of Chemical Engineering, it is only comparable with an industrial implementation to some extent.

It is essential that the column operates around some specified operating points in order to obtain acceptable product purities in spite of disturbances that occur. If acceptable product purities are not achieved, there is no reason for using an energy saving Kaibel distillation column. Using an MPC this would be achieved only if the model error is small or if the MPC is robust to model errors.

4.1 Model extension

The modeling part of the work is to include an *efficiency parameter* that describes insufficient vapor mixing at the column stages. In a real distillation column is it unlikely that the vapor flow from a stage only mixes with the stage above. It is more likely that some vapor would mix with several stages above. This effect is called *vapor bypassing* and is described briefly in Honeywell (2008). When the vapor mixes with more than one stage, the steady state operation is quite similar to a column with fewer stages without vapor bypassing. Therefore a parameter that adjusts the bypassed vapor is referred to as an efficiency parameter.

4.2 Model predictive control

The Kaibel distillation column is a multivariable process as presented in Chapter 3. It has four manipulated variables that control the product compositions, and four controlled variables that are temperatures. It is possible to control the process with decentralized PI controllers as shown in Chapter 3, but it can result in unacceptable product purities for some disturbances.

The Kaibel column is an interacted process, i.e. an input affects all outputs. Thus, a decentralized controller would give worse performance than an optimal multivariable controller.

The author wants to implement, test and compare *three* different MPC designs for control of the Kaibel column model. The first MPC design should be an MPC designed from scratch in order to use the MPC theory to be presented in later chapters and in addition obtain deeper knowledge on MPC design issues. The three MPCs can be summarized:

- **First design:** Use the presented theory to design a single layer MPC from scratch.

- **Second design:** Design a single layer MPC with use of MPC Toolbox in MATLAB.

- **Third design:** Design an MPC combined with the existing PI-control (supervisory MPC) using MPC Toolbox.

Hence, the first design should be done for the author's own advantage of learning basic problems regarding design of an MPC. The two last designs should be made for further comparison since they are made using the same design tool; MPC Toolbox in MATLAB. Results from simulations of a single layer MPC compared to a supervisory based MPC should confirm theoretical assertions regarding these two approaches. The phrase *supervisory MPC* is explained later in Chapter 6.

An appropriate way to include the comparison of these MPCs is in the sensitivity analysis part that is one of the main tasks of this thesis. It is interesting to compare these MPCs with respect to different model errors, i.e. input uncertainty, time delay and for different values of the efficiency parameter that is supposed to be included in the model extension part in the thesis. Because of the lack of a specific performance requirement for the Kaibel column, the author want to mainly focus on comparison between these two MPCs, and also do some comparison with the existing decentralized controller in the sensitivity analysis section.

As pointed out earlier it is important to have a proper mathematical model of the controlled process in order to fulfill performance criteria when MPC is used. However, an exact and probable nonlinear model can hardly be used in an optimal controller because it leads to a very difficult optimization problem, typically non-convex[1]. Therefore a simpler MPC which e.g. is based on a linear model must be robust to model errors. Hence, it is important to know if such an MPC implementation would result in a satisfactory control performance, and recommend an MPC design that gives the best performance.

Unfortunately, it was not possible to test the MPC on the laboratory facilities caused by mechanical problems with the column. However, it is very important to describe how an MPC implementation should be done at the laboratory for further work on this. A later experimental test would confirm if model predictive control of the column is preferable in difference to simple PI-control for future industrial use.

4.3 Evaluation of alternative MPC approaches

There exist several MPC approaches; therefore a brief discussion should be done in order to evaluate these alternatives for control of the Kaibel distillation column. The author want to give only a qualitative evaluation of some alternative approaches and spend more of the time on the actual MPC implementations and testing of these.

[1] When an unconstrained objective function is convex, it is easy to find the global optimum because it is located the place where the first derivative of the objective function is zero. For a nonconvex optimization problem things becomes harder, because the first derivative of the objective function is zero several places and the global optimum is much harder to find. A non-convex optimization problem is also much harder than a convex problem when constraints are added to the problem.

CHAPTER 5

MODEL EXTENSION

As mentioned, it is important to make the mathematical model of the Kaibel distillation column more physical. In the work of Kvernland (2008) *heat loss* was included to the model to make it more physical. Heat loss made the overall product composition to decrease, i.e. increased impurity flow. Since the column became colder as a result of the heat loss, less was vaporized and the top product became more pure, but the other product compositions decreased.

A short summarization of the heat loss modeling is added here because of later use. As mentioned, heat loss make some vapor at each stage be condensed into liquid in the column. The amount of vapor that is condensed into liquid at a column stage k is written as

$$\Delta V_k = \frac{\Delta Q_{k,\text{loss}}}{\Delta_{\text{vap}} H(T_k)}, \tag{5.1}$$

where $\Delta_{\text{vap}} H(T_k)$ is the molar enthalpy of vaporization and T_k is the temperature at the current stage. The molar enthalpy of vaporization is a constant that can be found in chemical handbooks for each component. $\Delta Q_{k,\text{loss}}$ denotes the heat loss itself at stage k and is calculated by applying *Newton's law of cooling*;

$$\Delta Q_{k,\text{loss}} = U_k A_k (T_k - T_s), \tag{5.2}$$

where U_k is the heat transfer coefficient, A_k is the surface area and T_s is the temperature in the surroundings.

5.1 Vapor bypassing

A more physical model was also a motivation for including *vapor bypassing* (Honeywell, 2008) to the model.

Including the vapor bypassing keeps the number of column stages to the same, but introduces an efficiency parameter (α) that describes insufficient vapor mixing at the column stages. The new vapor model is written as

$$V_k y_{i,k} = (1-\alpha) V_{k-1} y_{i,k-1} + \alpha V_{k-2} y_{i,k-2}. \tag{5.3}$$

Summation over all components i shows that the vapor flow from stage k is distributed to the two stages above; $k-1$ and $k-2$,

$$V_k = (1-\alpha) V_{k-1} + \alpha V_{k-2}. \tag{5.4}$$

This principle is shown in Figure 5.1. If this efficiency parameter (also referred to as the bypass factor) equals zero, we are left with the original model. Since some vapor skips a step (decided by the bypass factor α) on its way upwards in the distillation column it is quite intuitive that the products above the bottom product become less pure as the bypass factor increases.

5.2 Simulations with vapor bypassing

The distillation column is simulated using the *constant* nominal values (from Table 3.2) for the inputs and Figure 5.2 show how the product compositions are affected by different bypass factors. The simulations show that the product compositions changes quite much even for small values of the bypass factor. The bottom product composition is not affected much compared to the three others. Figure 5.3 shows from the same simulation how the temperature profile along the distillation column moves from a nominal position when the vapor bypassing is added to the model.

It is maybe more interesting to have a look at some simulation results when the controllers are switched on. This is to see what the controller does to counteract the vapor bypassing in order to achieve the temperature setpoints. Plots of the product compositions, the controlled temperatures and the inputs from this simulation are shown in Figure 5.4. The simulation starts with nominal values. The liquid split ratio and the two side streams remain at around the same values, but the reflux increases (Figure 5.4d).

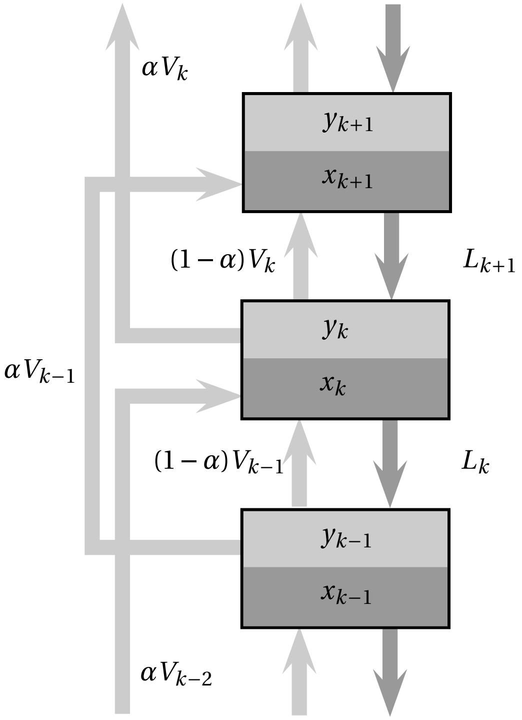
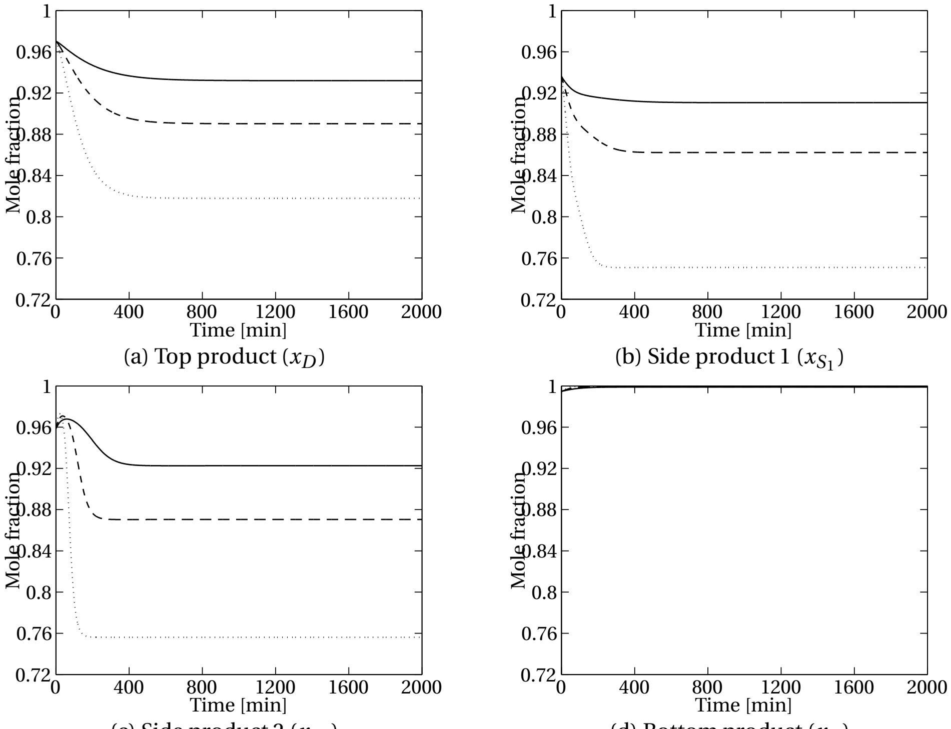


Figure 5.1: Bypassing of vapor flow.



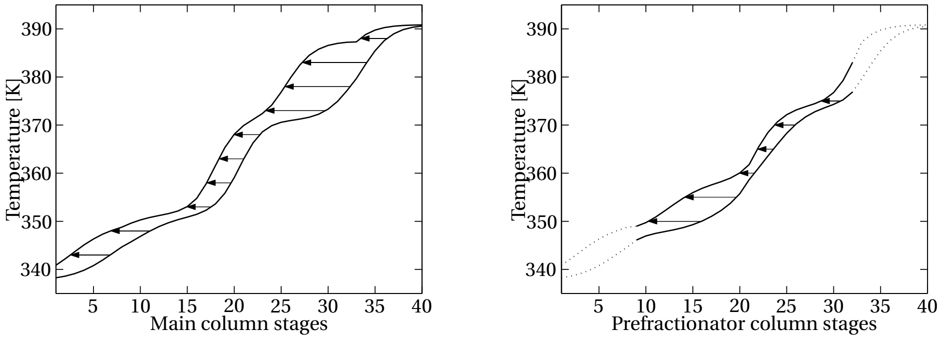
(a) Top product (x_D)

(b) Side product 1 (x_{S_1})

(c) Side product 2 (x_{S_2})

(d) Bottom product (x_B)

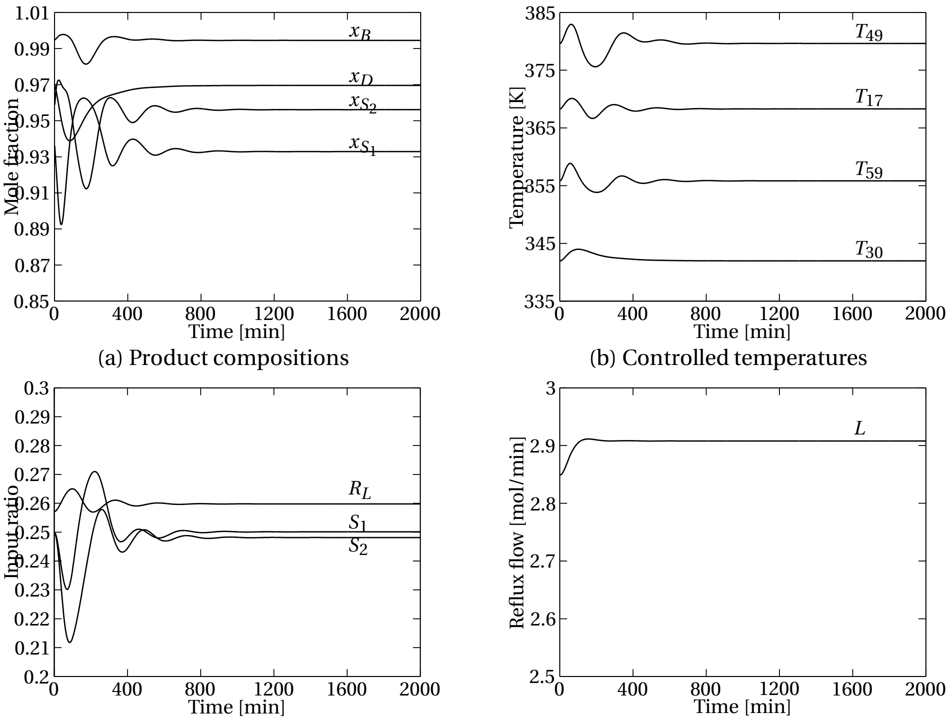
Figure 5.2: Effect of vapor bypassing with constant inputs (open-loop). $\alpha = 0.005$ (solid), $\alpha = 0.01$ (dashed), $\alpha = 0.02$ (dotted).



(a) Temperatures along main column.

(b) Temperatures along prefractionator column.

Figure 5.3: Change of temperature profile along column, the vapor bypassing factor was 0.02 in this open-loop simulation. The nominal temperature profile is to the right in the two plots. When the profile moves to the left it seems that the column gets much warmer from an ordinary plot of the temperature versus time. Stage 1 is the bottom stage and stage 40 is the top stage (see Figure 3.5).



(a) Product compositions

(b) Controlled temperatures

(c) Liquid split ratio and side streams

(d) Reflux

Figure 5.4: Effect of vapor bypassing in closed-loop with decentralized control. The vapor bypassing factor was 0.02 in this simulation. Increased vapor bypassing is counteracted by increased reflux.

CHAPTER 6

INTRODUCTION TO MODEL PREDICTIVE CONTROL

Model predictive control (MPC) optimizes the future behavior of a plant based on a multivariable model of this plant. It has mainly been used in the petrochemical industry (Maciejowski, 2002), but have gradually gained interest in other sectors of control engineering, e.g. in control of vehicles. Markus & Lee (1967) define optimal control as

> *One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time interval, after which a new measurement of the function is computed for this new measurement. The procedure is then repeated.*

This chapter presents the *linear quadratic regulator* that was the first optimal controller to be developed. *Dynamic matrix control* is also presented to show the development towards modern MPCs. The MPC that is based on a state-space model together with related issues are presented in more detail afterwards. It is *this* controller that is referred to as an *MPC* in this thesis, even though dynamic matrix controllers and other optimal controllers can be considered to be in the MPC family. A comparison between the introduced linear controllers is done as a short summary after the MPC section. Finally, some other variants of model predictive controllers are presented, followed by a discussion on model identification for MPCs and last; use of control layers.

6.1 Linear quadratic regulator

The linear quadratic regulator (LQR) proposed by Kalman (Kalman, 1960a,b) minimizes a quadratic objective function J that penalizes input and state deviations for each time step k. An objective function for optimal control can be written as[1]

$$J = \sum_{k=0}^{\infty} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right). \tag{6.1}$$

x_k is a vector of the future states values at time step k, similar for the inputs; u_k. Their desired values at each time step are given by $x_{k,\text{ref}}$ and $u_{k,\text{ref}}$. These values need to be computed on the basis of the desired output $y_{k,\text{ref}}$ and the process model. x_0 is the given initial condition. Q and R are weight matrices for state and input deviations respectively. The objective function from Equation (6.1) is an infinite sum; hence the corresponding optimal controller has an *infinite horizon*. The LQR can be applied on the discrete-time, linear state-space model

$$x_{k+1} = Ax_k + Bu_k, \tag{6.2a}$$

$$y_k = Cx_k. \tag{6.2b}$$

The system dimension is defined as

$$x_k \in \mathbb{R}^{n_x}, \tag{6.3a}$$

$$u_k \in \mathbb{R}^{n_u}, \tag{6.3b}$$

$$y_k \in \mathbb{R}^{n_y}. \tag{6.3c}$$

If $x_{k,\text{ref}} = 0$ and $u_{k,\text{ref}} = 0$, the solution to the minimization problem is shown to be a proportional controller (Foss, 2004);

$$u_k = -Kx_k, \tag{6.4}$$

where the matrix K is computed from the *Ricatti equation* (Qin & Badgwell, 1997). If the reference values are nonzero, the feedback term in Equation (6.4) must in addition include a feed-forward term from the reference values (Foss, 2004).

The linear quadratic gaussian controller (LQG) is an LQR combined with a Kalman filter. The Kalman filter is used to estimate the states, since it is rarely that all states are measured in a process. This gives the similar proportional controller;

$$u_k = -K\hat{x}_k, \tag{6.5}$$

where $\hat{x}_k$ is the state estimate for time step k.

[1] The term $||x_k - x_{k,\text{ref}}||_Q^2$ means $(x_k - x_{k,\text{ref}})^\top Q(x_k - x_{k,\text{ref}})$

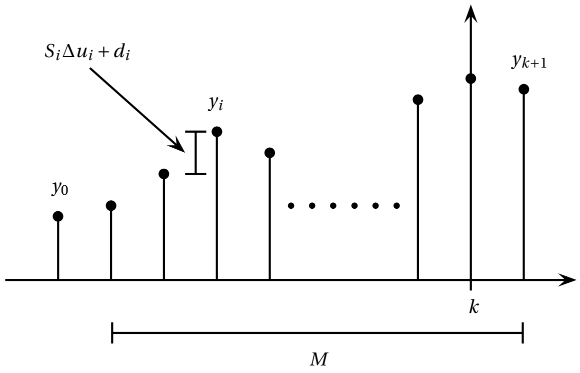


Figure 6.1: How the output y_{k+1} is written as a sum of past inputs multiplied with some specified unit step response coefficients, plus an initial condition for the output and disturbance.

6.2 Dynamic matrix control

One of the first MPC technologies was presented by Cutler & Ramaker (1979) and was called dynamic matrix controller (DMC) (Qin & Badgwell, 1997). Unlike LQR and LQG, the DMC optimizes at every time step. DMC uses a linear step response model for the process. The output is the measurements (y) that we want to control. Since it uses a linear step response model the output can be written as summation of earlier inputs multiplied with their respective step response coefficients plus initial output and un-modeled effects (Garcia & Morshedi, 1986),

$$\begin{aligned} y_{k+1} &= S_1\Delta u_k + S_2\Delta u_{k-1} + \cdots + S_M\Delta u_{k-M+1} + y_0 + d_{k+1} \\ &= \sum_{i=1}^{M} S_i\Delta u_{k-i+1} + y_0 + d_{k+1}, \end{aligned} \tag{6.6}$$

where Δu_k denotes the change in input at time step k. This is illustrated in Figure 6.1. S_i is a matrix of unit step response coefficients and d_k represents the un-modeled factors that affect the output. M is the number of time steps that is required to transfer the process into steady state. The output at a future time $k+l$ ($l>0$) can be written as sum of effects by future and past moves,

$$\begin{aligned} y_{k+l} = &\underbrace{\sum_{i=l+1}^{M} S_i\Delta u_{k+l-i} + y_0}_{\text{effect of past moves}} + \underbrace{\sum_{i=1}^{l} S_i\Delta u_{k+l-i}}_{\text{effect of future moves}} \\ &+ \underbrace{d_{k+l}}_{\text{estimated disturbance at time } k+l} . \end{aligned} \tag{6.7}$$

Define the new variable y^{p}_{k+l}, which is the effect of past moves,

$$y^{\mathrm{p}}_{k+l} = \sum_{i=l+1}^{M} S_i \Delta u_{k+l-i} + y_0. \tag{6.8}$$

The future outputs up to a future time step N can then be written in matrix form,

$$\begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+N} \end{bmatrix} = \begin{bmatrix} y^{\mathrm{p}}_{k+1} \\ \vdots \\ y^{\mathrm{p}}_{k+N} \end{bmatrix} + \Lambda \begin{bmatrix} \Delta u_k \\ \vdots \\ \Delta u_{k+N} \end{bmatrix} + \begin{bmatrix} d_{k+1} \\ \vdots \\ d_{k+N} \end{bmatrix}. \tag{6.9}$$

Λ is called the *dynamic matrix* and consist of the unit step response coefficients, derivation of this matrix can be viewed in Garcia & Morshedi (1986). It is desirable to have the future output values equal to some target value, thus by setting these future outputs (left side of Equation (6.9)) equal to these target values given by y_{ref}, and then solve for the inputs gives

$$\begin{aligned} \begin{bmatrix} y_{\mathrm{ref}} - y^{\mathrm{p}}_{k+1} - d_{k+1} \\ \vdots \\ y_{\mathrm{ref}} - y^{\mathrm{p}}_{k+N} - d_{k+N} \end{bmatrix} &= \Lambda \begin{bmatrix} \Delta u_k \\ \vdots \\ \Delta u_{k+N} \end{bmatrix} \\ \Rightarrow \begin{bmatrix} \Delta u_k \\ \vdots \\ \Delta u_{k+N} \end{bmatrix} &= \Lambda^{\dagger} \begin{bmatrix} y_{\mathrm{ref}} - y^{\mathrm{p}}_{k+1} - d_{k+1} \\ \vdots \\ y_{\mathrm{ref}} - y^{\mathrm{p}}_{k+N} - d_{k+N} \end{bmatrix}, \end{aligned} \tag{6.10}$$

where $\Lambda^{\dagger}$ is called the *pseudoinverse* of Λ and is equivalent with the least-square solution of Equation (6.9) (Strang, 2006). The pseudoinverse of a matrix A is defined as (Strang, 2006),

$$A^{\dagger} = \left(A^{\top} A\right)^{-1} A^{\top}. \tag{6.11}$$

DMC require estimation of the un-modeled effects, d_k. These un-modeled effects can be seen as a constant difference between the measured output and the computed output based on earlier inputs, hence

$$d_k = y^{\mathrm{m}}_k - y_k, \tag{6.12}$$

where y^{m}_k is the measured output at time step k. A common way to estimate future un-modeled effects is to assume that they are constant, i.e.

$$d_{k+l} = d_k, \ \forall \ l = 1, \ldots, N. \tag{6.13}$$

If the multivariable process is unconstrained, a DMC would work very well. Unconstrained processes are rarely, so it is desirable to have an algorithm that can handle limitations on inputs and outputs. By looking from an economically point of view it is often optimal to operate the plant as close as possible to some specified constraints. Quadratic dynamic matrix control (QDMC) was first introduced by Cutler et al. (1983) and it is closer described in Garcia & Morshedi (1986) and Camacho & Bordons (2004). QDMC uses a quadratic objective function and handle the input and output constraints explicitly (Qin & Badgwell, 1997). The quadratic objective function used in a QDMC can be written as (Ying & Joseph, 1999),

$$J = \sum_{i=1}^{N} \left(||y_{k+i} - y_{\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right). \tag{6.14}$$

This objective function is minimized with respect to the future inputs. Penalization of the inputs is often added to the objective function. Constraints on inputs and outputs are added to the optimization problem, which is solved by a quadratic programming (QP) solver.

6.3 Model predictive control

The major advantage of today's MPC technology compared to an ordinary DMC is the ability to handle constraints on inputs and states. This is an important property since we will always have limited inputs to the process and that it is often desirable to keep the measurements or the states within specified bounds.

There are many variants of model predictive controllers. E.g. there are several ways of defining the objective function, or whether the MPC uses a linear state-space model or a nonlinear model as its prediction model.

The following section first deals with the objective function formulation and the constraints. Afterwards the issue of adding integral action to an MPC is presented. Smoother control actions are achieved by defining a smooth reference trajectory towards the real reference value, which is presented next. Thereafter, a brief discussion on how to solve the optimization problem is given, and finally some comments on stability of an MPC are made.

Objective function

There are several ways to define the optimization problem that must be solved in the MPC. First, the objective function can have different formulations. The most common formulation is a quadratic programming (QP) formulation. In a QP formulation the objective function is quadratic and the constraints are linear. An alternative to the QP formulation is a linear programming (LP) formulation, where the objective function and the constraints are linear. According to Hovd (2008) a QP formulation gives smoother control actions and the MPC will have more intuitive tuning parameters. The QP formulation is used in this thesis.

The objective function given by Equation (6.1) is an *infinite horizon* objective function since it takes account for infinite many time steps into the future. This infinite horizon formulation is possible when there are no constraints in the optimization problem, e.g. for an LQR controller. An MPC uses constraints in the optimization problem, and a finite horizon formulation must be used. To deal with this it is possible to divide the infinite sum into two parts,

$$J = \sum_{k=0}^{N-1} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right) + \sum_{k=N}^{\infty} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right). \tag{6.15}$$

N is the control and prediction horizon introduced in Section 6.2. After this horizon the input can be set to a constant or proportional to the state (e.g. use the LQR gain). In addition, if the closed-loop system is stable after the horizon (something else would not be of interest), the second term at the right side of Equation (6.15) is a finite sum, i.e.

$$\sum_{k=N}^{\infty} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right) = ||x_N - x_{N,\text{ref}}||_P^2. \tag{6.16}$$

P is referred to as a *terminal state penalty matrix* (Muske & Rawlings, 1993). This matrix depends of the stability of the process to be controlled and the input behavior after the horizon. This gives a finite horizon objective function that can be used in an MPC;

$$J = \sum_{k=0}^{N-1} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||u_k - u_{k,\text{ref}}||_R^2 \right) + ||x_N - x_{N,\text{ref}}||_P^2. \tag{6.17}$$

As mentioned in the first section the matrices Q and R are weight matrices. These can be chosen freely, but it is required that $Q \geq 0$ and $R > 0$ (Imsland, 2007) such that the objective function becomes convex. These matrices can be used to tune the MPC performance.

Constraints

Input and output constraints can be written in the following form

$$u_{\min} \leq u_{k+i} \leq u_{\max}, \quad i = 0, 1, \dots, N-1, \tag{6.18a}$$

$$\Delta u_{\min} \leq \Delta u_{k+i} \leq \Delta u_{\max}, \quad i = 1, 2, \dots, N-1, \tag{6.18b}$$

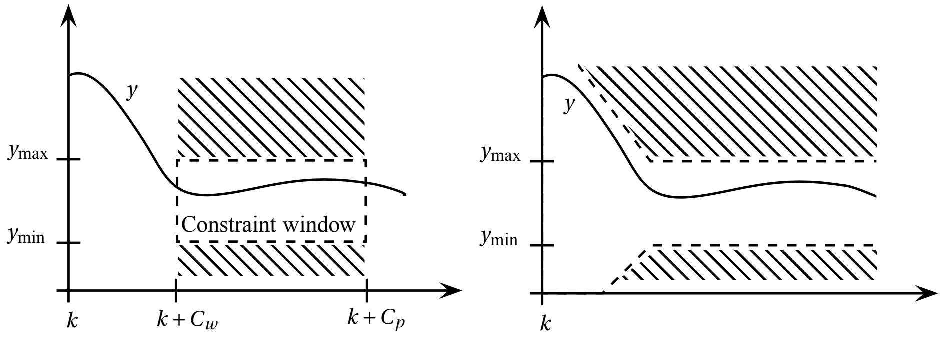
$$y_{\min} \leq C x_{k+i} \leq y_{\max}, \quad i = 0, 1, \dots, N. \tag{6.18c}$$

When it comes to constraints for the optimization problem, there are also several representations. All inequality constraints can be gathered like presented in Morari & Lee (1999),

$$D_x x + D_u u \leq \psi. \tag{6.19}$$

D_x and D_u are matrices that must be constructed and ψ will be a vector which length is equal to the number of inequality constraints. This inequality must include specified minimum and maximum constraints for inputs and outputs.

It is necessary to relax the constraints if the optimization problem becomes infeasible. This infeasibility can arise from huge disturbances on the process. It is therefore important that the MPC is able to handle such situations. Muske & Rawlings (1993) deals with such situations by using a *constraint window*. The principle is shown in Figure 6.2a. The figure show an example with a single output y with specified lower and upper bounds, $y_{\min}$ and $y_{\max}$. Suppose that a large disturbance occurs at time k such that the controlled output comes above its maximum limit. A specified number of steps, C_w, decides how long into the horizon that the constraints should take effect from and C_p is how long the constraints must be fulfilled, typically the same as the control horizon, N. By using this window the output will be in a feasible area for some time after the disturbance so that the MPC will be able to control it inside its bounds. A similar way to deal with this constraint problem is to create a *funnel*, with a specified slope such that the process variable to be controlled is able to come inside the bounds. This principle is shown in Figure 6.2b. Another approach to avoid infeasibility problems is to use *soft constraints* instead of *hard constraints*. Ordinary inequality constraints are hard constraints, since they *must* not be exceeded. Soft constraints can be exceeded but it is preferable that the constrained variables are inside the original feasible region. It is relatively easy to add soft constraints to the optimization problem by use of a *slack variable*, ϵ. This positive slack variable is added at the left side of the inequality constraint and in the objective function. Suppose that the original objective function is denoted by J, the optimization variable is θ and the inequality constraint is $\Omega\theta \leq \omega$. By use of soft constraints, the optimization problem will be



(a) Constraint window.

(b) Funnel.

Figure 6.2: Constraint relaxing.

$$\min_{\theta,\epsilon} J + \rho||\epsilon||^2 \tag{6.20a}$$

subject to

$$\Omega\theta \leq \omega + \epsilon, \tag{6.20b}$$

$$\epsilon \geq 0, \tag{6.20c}$$

where ρ is a positive constant.

Integral action

Integral action is not straight forward even for an MPC. Hovd (2008) presents two approaches to add integral action to an MPC. The most common approach is to use the *input changes* as free variables in the optimization, instead of the input itself. Therefore, instead of Equation (6.17), the objective function will be

$$J = \sum_{k=0}^{N-1} \left(||x_k - x_{k,\text{ref}}||_Q^2 + ||\Delta u_k||_R^2 \right) + ||x_N - x_{N,\text{ref}}||_P^2. \tag{6.21}$$

Using this objective function formulation we are not required to compute $u_{k,\text{ref}}$, which is an advantage.

The other approach to add integral action described is to include a disturbance estimate to the model and use this estimate as a feed-forward to the MPC. If the model is linear, the model given by Equation (6.2) must be modified to include disturbances. E.g. the model can be like

$$x_{k+1} = Ax_k + Bu_k + B_d d_k, \tag{6.22a}$$

$$y_k = Cx_k + p_k. \tag{6.22b}$$

d_k and p_k are disturbances affecting state and output respectively.

Reference trajectory

The objective function given in Equation (6.21) can be modified to include a *reference trajectory*, which defines a smooth trajectory as the reference from the current measurement y_k towards the real reference value, y_{ref}. The objective function is then modified to

$$J = \sum_{k=0}^{N-1} \left(||\hat{y}_{k+1} - w_{k+1}||_Q^2 + ||\Delta u_k||_R^2 \right) + ||x_N - x_{N,\text{ref}}||_P^2, \tag{6.23}$$

where w_k is the reference trajectory. This trajectory can be determined from a first order system as described in Camacho & Bordons (2004),

$$w_{k+i} = \beta w_{k+i-1} + (1-\beta) y_{\text{ref}}. \tag{6.24}$$

This equation must use the initial condition $w_k = y_k$. The parameter β $(0 \leq \beta \leq 1)$ decides the reference trajectory smoothness. The simplest approach is to set $\beta = 0$ and thus have no trajectory at all. The principle of reference trajectory is shown in Figure 6.3.

Solving the optimization problem

By using a QP formulation, the optimization problem will be an ordinary QP problem with linear constraints. There exist several algorithms for solving these kinds of problems. The most common one is the *active set method* (Fletcher, 1987). This

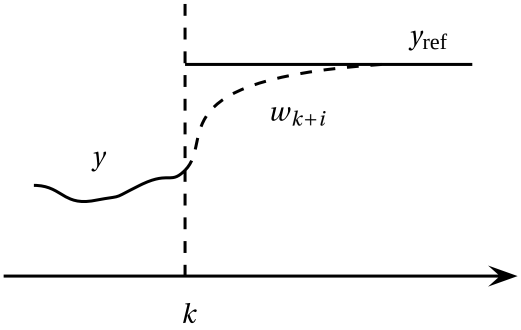


Figure 6.3: Reference trajectory.

method put some of the inequality constraints in a subset called the *active set* and treat them as equality constraints. The method starts with an initial solution and then iteratively solves sub-problems to improve the solution. These sub-problems are QP problems with equality constraints, those in the active set. A QP problem with only equality constraints can be solved analytically using the *Karush-Kuhn-Tucker* (KKT) conditions (Kuhn & Tucker, 1951) and the improved solution is easily found. The inequality constraints that are not added in the active set are not taken in consideration in the sub-problem and the feasibility can be lost when an improved solution is found. If this happens, *line search* is used to go back where the feasibility was lost. Nocedal & Wright (2006) presents the active set method in closer detail, not just a brief introduction like in this chapter. Other QP algorithms that are explained in Nocedal & Wright (2006) are *interior-point methods* and the *gradient projection method.*

Stability

The *regulation problem* is when it is desirable to let $x_k \to 0$ as $k \to \infty$. According to Imsland (2007) it is guarantied that x_k goes to zero if

1. The MPC optimization problem is feasible at $k = 0$,
2. The state must be detectable through the weight matrix Q,
3. The constraints must hold on the infinite horizon.

The first item is obvious since an infeasible optimization problem would not give a solution and can not assure stability. To avoid this problem it may be necessary to slack the constraints as described above. It is also clear that the model must be detectable, i.e. when all the unobservable modes are asymptotically stable[2] (Foss,

[2] An asymptotically mode converge to an asymptotic value when the time goes to infinity (Khalil, 2000)

2004). The third item must also hold because the optimization problem must be feasible after the horizon. To make sure that the constraints holds on the infinite horizon the prediction horizon N must be chosen long enough or it can be assured by adding a *terminal state constraint*. A terminal state constraint is just an equality constraint at the end of the horizon that must be fulfilled. This constraint may be like

$$x_N = 0. \tag{6.25}$$

A common approach in the design of MPCs is to use a *pre-stabilizing* state feedback. This pre-stabilizing feedback make the input be divided into two parts,

$$u_k = -Kx_k + c_k. \tag{6.26}$$

The feedback term $-Kx_k$ is a stabilizing proportional controller, which is the pre-stabilizing feedback. This stabilizing controller K can be chosen freely, but is common to use the LQR gain (Equation (6.4)). Using this divided input term the optimization variable is c_k instead of u_k. For unstable processes, pre-stabilizing state feedback is very common. It is required if terminal constraints mentioned above are not used.

6.4 LQR, (Q)DMC and MPC comparison

The linear optimal controllers introduced in this chapter are compared in Table 6.1.

6.5 Variants of model predictive controllers

There are various variants of MPCs, this section briefly introduce nonlinear MPC (NMPC) and explicit MPC (EMPC).

Nonlinear model predictive control (NMPC) is used when the process to be controlled is highly nonlinear and an ordinary linear MPC cannot be used. An NMPC can be designed as a linear MPC by simply replacing the linear model as the operation point changes (Maciejowski, 2002). In this approach a set of linear models needs to be linearized from the nonlinear model beforehand. The most common NMPC approach is to re-linearize the nonlinear model during operation around the present operating point (Maciejowski, 2002). An example on a process that can be quite nonlinear is a distillation column (Skogestad, 1997). Therefore it can be preferable to use an NMPC instead of a linear MPC to obtain the desirable product purities from the column. More NMPC theory can be found in Allgöwer et al. (2004).

For explicit model predictive control (EMPC) the optimization problem is solved beforehand. The state-space is divided into *regions* where the optimal control solution has almost the same input. Figure 6.4 show an example of how the state-space with only two states (x_1 and x_2) can be divided. Each region in the figure is

	Re-optimizes at every step	**Handles limitations in inputs and outputs**	**Uses state-space model**	**Nonzero target value**
LQR / LQG	No	No	Yes (if not all states are measured)	Yes (require feed-forward from target values)
DMC	Yes	No	No (only step response coefficients)	Yes
QDMC	Yes	Yes	No (only step response coefficients)	Yes
MPC	Yes	Yes	Yes	Yes

Table 6.1: Comparison between different linear optimal controllers.

associated with two variables, K_i and k_i; where the number i identifies a region. The control law will then be

$$u_k = -K_i x_k + k_i. \tag{6.27}$$

The EMPC is just a look-up table that finds the pre-computed values of K_i and k_i based on the present state. The main advantage of the EMPC is the computational efficiency compared to an ordinary MPC which solves a mathematical optimization problem in each time step. It is also a safety point of view by using an EMPC since it is very easy to validate the implementation and do simulations offline. More on EMPC can be found in Tøndel et al. (2003).

6.6 Identification of model

Model identification or system identification can be defined as (partly taken from Ljung (1999)):

> *Input and output signals from the process are recorded and subjected to data analysis in order to infer a model. Hence, the model is built from experimental data from the process.*

The experimental data is often recorded from a designed identification experiment.

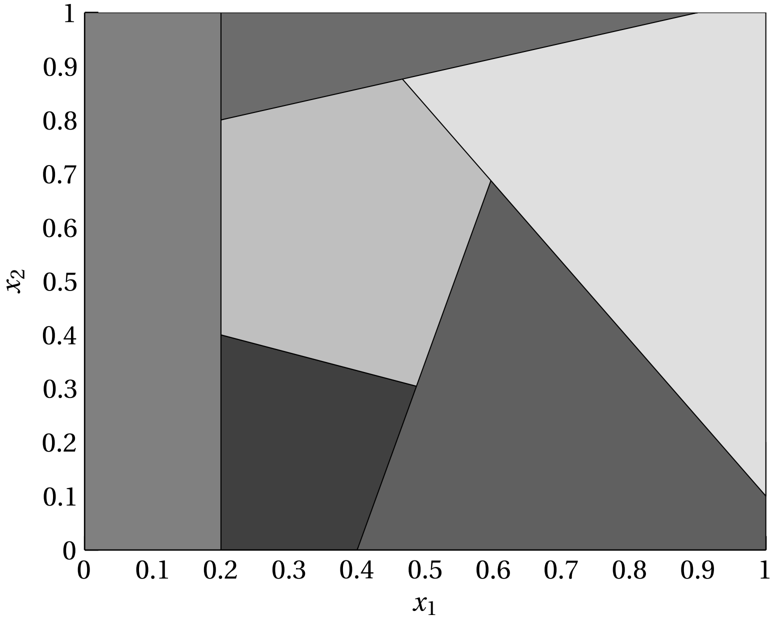


Figure 6.4: Example of a divided state-space for an EMPC with two states.

An MPC require a model because it must predict future behavior to optimize the control of the process. Most common is to use linear models, like those given in Equation (6.2) and (6.22). Using linear models give a *linear MPC*. According to Hovd (2008) the industrial practice is to use *step responses* as the basis for the models. The simplest approach for a multivariable model would be to perturb a small step in the inputs sequentially and record the behavior of each output. Then an approximated transfer function matrix can be constructed from the recorded data and from there a linear model.

If the design of an MPC has required much work on modeling the process, there is no reason for *not* using this model. Often the physical parameters in such a model needs to be identified, this is called *grey box identification*, where the model structure is known, but not the model parameters. These parameters can be identified by various methods; the most common is to use a least square estimation (Ljung, 1999) for estimating these parameters. Suppose the process output at time t is written as $y(t)$ and the model output is $\hat{y}(t,\theta)$, where θ is a vector consisting of the model parameters. Then, by considering a recorded set of data of length N_r, the least square problem is written as

$$\min_{\theta} \sum_{t=1}^{N_r} \left(y(t) - \hat{y}(t,\theta)\right)^2. \tag{6.28}$$

Least square estimation can also be done recursively (Ljung, 1999), such that the model parameters are continuously updated as the plant is in operation.

Identifying models without knowing anything about the process is often referred to as *black box identification*. If we want to obtain a linear model for use in an MPC, the *subspace identification method* can be suitable. The subspace method is among other articles and books described in Ljung (1999). The method computes the state-space matrices A, B, C and D from recorded input and output data. Data from step response experiments performed on the process can be loaded into a subspace method algorithm, and a linear state-space model around the process' operating point is easily found.

6.7 Control layers

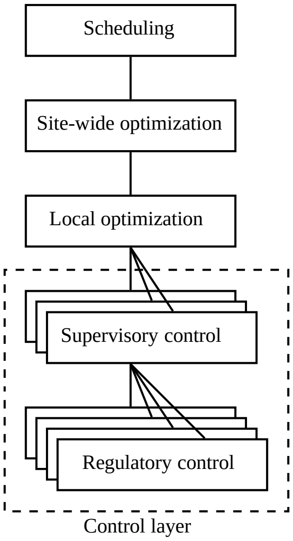


Figure 6.5: Control hierarchy (Skogestad & Postlethwaite, 2005).

It is common to divide the control of a plant into different layers. Figure 6.5 show how such a hierarchy can be like. The *regulatory control* is the bottom level of control and typically consists of PID controllers with given setpoints. These setpoints can come from the *supervisory control* which often is an MPC. These two levels are referred to as the *control layer* (Skogestad & Postlethwaite, 2005).

Above the control layer, higher optimization objectives are solved. These objectives involve the whole plant operation and can be controlled by an overall MPC. It is the ”control layer” that is taken into consideration in this thesis.

It is not required to have the regulatory control level; an MPC is able to control the physical inputs directly instead of a PID controller. This will add some complexity into the MPC and the plant will be left uncontrolled if the MPC goes down. Therefore it can be an advantage to have this basis control level with PID controllers and use the MPC for setpoint adjustments. There are also other advantages of using a *supervisory MPC* compared to a *single layer* MPC as we will come back to later.

The decentralized controllers should be tuned properly in order to obtain a good time scale separation (Skogestad & Postlethwaite, 2005) between the two control layers; MPC layer and PI-control layer. By doing this, there will be several advantages according to Skogestad & Postlethwaite (2005):

- *The stability and performance of a lower (faster) layer is not much influenced by the presence of upper (slow) layers because the frequency of the ”disturbance” from the upper layer is well inside the bandwidth of the lower layer.*
- *With the lower (faster) layer in place, the stability and performance of the upper (slower) layers do not depend much on the specific controller settings used in the lower layers because they only affect high frequencies outside the bandwidth of the upper layers.*

These items emphasize the importance of well tuned controllers in the lowest layer.

CHAPTER 7

IMPLEMENTATION AND SIMULATION OF MPC

This chapter presents the different MPCs that are designed for the Kaibel column model. First, an MPC was constructed using the presented theory. Secondly, MPC Toolbox in MATLAB was used to control the column.

The first MPC was made in order to use the presented theory and obtain deeper knowledge on MPC design issues. As mentioned in the background chapter (Chapter 2), an MPC was also made during the author's final year project, but the occurring disturbances were assumed to be known in the simulations. This is an unrealistic approach for a real implementation, thus, it is desirable to make a disturbance estimator to work properly. It is only the first MPC design that uses a specified disturbance model in its prediction model.

A sensitivity analysis is done to mainly compare two different MPC designs; an MPC that controls the physical inputs directly (single layer MPC) and an MPC that controls the setpoints of the PI controllers used in previous work (supervisory MPC). These two MPC designs were made by use of MPC Toolbox in MATLAB. Results from the decentralized controller is also added to this section.

The prediction model that is used in the MPC implementations is without the vapor bypassing extension that was added to the model in Chapter 5. Hence, the bypassing factor was set to zero in the model when the linear prediction model was made in order to keep the prediction model simple. In the sensitivity analysis part different values of this bypass factor is tested in the simulation model to check the performance of the controllers.

7.1 MPC designs

MPC from constructed theory

Prediction model

The prediction model is the linearized model representing the distillation column that was given in Equation (3.17);

$$x_{k+1} = Ax_k + Bu_k + B_d d_k, \tag{7.1a}$$

$$y_k = Cx_k. \tag{7.1b}$$

The future measurements y_k need to be estimated to be used in the objective function given by Equation (6.23). These measurements are estimated by iterating the linear model from an initial value of the state. If the state vector is not measured in the plant, it needs to be estimated by an observer. An observer gives an estimate $\hat{x}_k$ for use in the MPC.

Optimization problem

The objective function formulation presented in the following section is given in Camacho & Bordons (2004).

The objective function used was given in Equation 6.23;

$$J = \sum_{k=0}^{N-1} \left(||\hat{y}_{k+1} - w_{k+1}||_Q^2 + ||\Delta u_k||_R^2 \right) + ||x_N - x_{N,\text{ref}}||_P^2. \tag{7.2}$$

It is desirable to rewrite the linear state-space model from Equation (7.1) to the following representation

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = M \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + N\Delta u_k + Pd_k, \tag{7.3a}$$

$$y_k = Q \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix}, \tag{7.3b}$$

where the matrices M, N, P and Q are given by

$$M = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \qquad N = \begin{bmatrix} B \\ I \end{bmatrix},$$

$$P = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} C & 0 \end{bmatrix}.$$

Let us now define some vectors,

$$\begin{aligned}
\hat{x} &= \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix}, \\
y &= \begin{bmatrix} \hat{y}_{k+1} \\ \hat{y}_{k+2} \\ \vdots \\ \hat{y}_{k+N} \end{bmatrix} = \begin{bmatrix} QM\hat{x} + QN\Delta u_k \\ QM^2\hat{x} + \sum_{i=0}^{1} QM^{1-i}N\Delta u_{k+i} \\ \vdots \\ QM^N\hat{x} + \sum_{i=0}^{N-1} QM^{1-i}N\Delta u_{k+i} \end{bmatrix}, \\
u &= \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N-1} \end{bmatrix}, \\
d &= \begin{bmatrix} \hat{d}_k \\ \hat{d}_{k+1} \\ \vdots \\ \hat{d}_{k+N-1} \end{bmatrix}.
\end{aligned}$$

The N estimated future outputs can then be written as

$$y = F\hat{x} + Hu + Gd, \tag{7.4}$$

where the matrix F is given by

$$F = \begin{bmatrix} QM \\ QM^2 \\ \vdots \\ QM^N \end{bmatrix}.$$

H and G are block lower triangular matrices defined as

$$\begin{aligned}
H_{ij} &= QM^{i-j}N, \;\; \forall\; i \geq j, \\
G_{ij} &= QM^{i-j}P, \;\; \forall\; i \geq j.
\end{aligned}$$

The objective function can now be rewritten,

$$J = ||y - w||_Q^2 + ||u||_R^2, \tag{7.6}$$

where $w = \begin{bmatrix} w_{k+1}^\top & w_{k+2}^\top & \dots & w_{k+N}^\top \end{bmatrix}^\top$. Q and R are block diagonal matrices.

Equation (7.4) can now be substituted into the objective function given above. By multiplying the parentheses and removing constant terms that does not have any effect on the optimal solution we get the following objective function

$$J = \frac{1}{2}u^\top \left(H^\top QH + R\right) u + \left(2H^\top Q\left(F\hat{x} + Gd - w\right)\right)^\top u. \tag{7.7}$$

The constraints can be added as a single inequality constraint,

$$\Phi u \leq \phi, \tag{7.8}$$

for some constructed matrix Φ and vector ϕ. This constraint can include specified maximum and minimum limits for input, input change and output.

Estimation

From Equation (7.7) we can see that the MPC needs to estimate the current state and future disturbances. Future disturbances are often set to be the same as the first disturbance estimate, hence

$$\hat{d}_{k+i} = \hat{d}_{k+1}, \ \ \forall \ i = 2, \dots, N-1.$$

By using this assumption it is possible to augment the existing state-space model given by Equation (7.1);

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \Omega \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \Psi u_k, \tag{7.9a}$$

$$y_k = Z \begin{bmatrix} x_k \\ d_k \end{bmatrix}, \tag{7.9b}$$

where the matrices Ω, Ψ and Z are given by

$$\Omega = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix}, \qquad \Psi = \begin{bmatrix} B \\ 0 \end{bmatrix},$$
$$Z = \begin{bmatrix} C & 0 \end{bmatrix}.$$

The augmented model in Equation (7.9) can be used for estimation of both states and disturbances. A discrete closed-loop observer for this system is written as (Chen, 1999)

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \Omega \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \Psi u_k + L\left(y_k - \hat{y}_k\right), \tag{7.10a}$$

$$\hat{y}_k = Z \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix}. \tag{7.10b}$$

y_k is the real measurement in the present time step and $\hat{y}_k$ is the estimated output. L is the observer gain and must be chosen such that the error dynamics $(\Omega - LZ)$ is stable.

To make sure that the estimates given by the observer is correct, the observed system need to be *observable* (Chen, 1999). This property can easily be checked by finding the rank of the *observability matrix* which is defined as (Chen, 1999)

$$\mathcal{O} = \begin{bmatrix} Z \\ Z\Omega \\ \vdots \\ Z\Omega^{n_x - 1} \end{bmatrix}. \tag{7.11}$$

If the observability matrix has full column rank the system is observable and thus, the states and disturbances converges to their correct values within finite time.

In the implementation the linearized model was reduced down to an order of 20 states, i.e. the size of the system matrix (A) is 20×20. Using this order together with the four measurements

$$y = \begin{bmatrix} T_{17} & T_{30} & T_{59} & T_{49} \end{bmatrix}^\top, \tag{7.12}$$

does not give observability for the system given in Equation (7.9). Hence, some measurements need to be added. This corresponds with a remark from Borrelli & Morari (2007) which states that the number of measurements must be greater or equal to the number of disturbances that needs to be observed for the system given by Equation (7.9) to be observable. Since the number of disturbances is seven, we need at least three more measurements.

There are only temperature sensors that are available at the laboratory column and therefore we should look for new temperature measurements along the column. The system (A, C) is observable, i.e. the states can be estimated alone, but not the augmented state-space model including the disturbances in addition using only these four measurements. Therefore by looking at which temperatures along the column which the disturbances influence most, it is possible to find good measurements for estimation. This is the same way of thinking as the *maximum gain rule* presented in Skogestad & Postlethwaite (2005). Briefly explained you should select those outputs that are greatest affected by the inputs. To find which temperature measurements that are affected most by the disturbances we want to estimate the, disturbance variables were increased by a step in the nonlinear model, and the temperature with the greatest change in its value was written down. Table 7.1 shows the results from these simulations. The input step sizes were around 10% of their nominal values. Figure 7.1 shows the absolute temperature change for each stage in the column when the feed flow was increased with 10%. From Table 7.1 we can see that there are only two different measurements that are not included in the existing measurement array (Equation (7.12)), T_6 and T_{53}. It is no point in adding the same measurement again (T_{59}) so we need to find an extra measurement that has some significant change in its value, but is not in Table 7.1. It was notified during the computation of Table 7.1 that the temperature T_{24} also had a great change in its value, therefore this temperature is added. The total measurement array then becomes

$$y = \begin{bmatrix} T_{17} & T_{30} & T_{59} & T_{49} & T_6 & T_{53} & T_{24} \end{bmatrix}^\top. \tag{7.13}$$

The temperature measurement locations are showed in Figure 7.2. The measurement array above give an observability matrix of full column rank, thus the disturbances and states can now be estimated. Note that it is only the four first temperature measurements from the array above that are controlled by the MPC.

Disturbance varible	Greatest change
V	T_{59}
R_V	T_6
F	T_{53}
z_D	T_{53}
z_{S_1}	T_{53}
z_{S_2}	T_{53}
q	T_{53}

Table 7.1: Applying maximum gain rule to find new measurements. The disturbance variables were increased by about 10% from their nominal values.

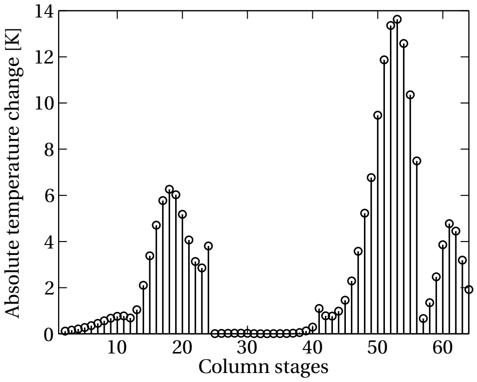


Figure 7.1: Step in feed flow (10%). We can see that T_{53} has the greatest change with almost 14 K.

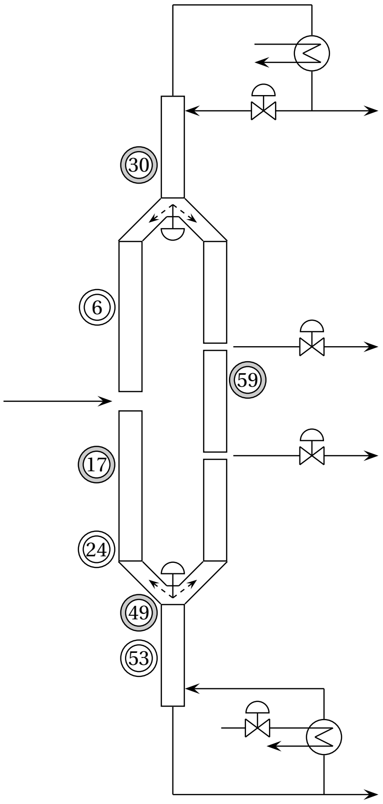


Figure 7.2: Temperature measurement locations. Temperatures that are marked grey are controlled by the MPC, the extra temperature measurements are needed to make disturbances and states observable.

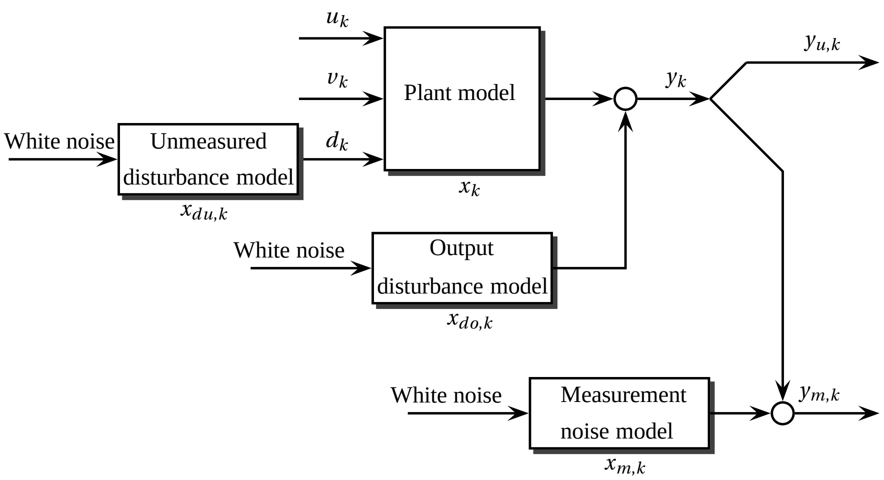


Figure 7.3: Model used in MPC Toolbox (Bemporad et al., 2009).

MPC Toolbox in MATLAB

MPC Toolbox in MATLAB is a quite simple software tool to design a linear MPC. This toolbox is used for the next MPC implementations. Let us first present some theory on how this toolbox is made. The toolbox is closer described in Bemporad et al. (2009).

Prediction model

MPC Toolbox uses the following linear model when predicting future outputs,

$$x_{k+1} = Ax_k + B_u u_k + B_v v_k + B_d d_k, \tag{7.14a}$$
$$y_{m,k} = C_m x_k + D_{vm} v_k + D_{dm} d_k, \tag{7.14b}$$
$$y_{u,k} = C_u x_k + D_{vu} v_k + D_{du} d_k. \tag{7.14c}$$

v_k and d_k is measured and unmeasured disturbances respectively, and affects the states via the matrices B_v and B_d. $y_{m,k}$ denotes measured outputs and $y_{u,k}$ are the outputs that can not be measured. The model is illustrated in Figure 7.3.

Optimization problem

The input changes (Δu's) are computed with a quadratic optimization problem. It penalizes the output error, input change and the input error, Equation (7.15a), (7.15b) and (7.15c) respectively. By input error we mean the deviation from the input value, u_k, to a target value, $u_{k,\text{target}}$. This target value needs to be computed.

$$
\begin{aligned}
\min_{\Delta u(k|k),\ldots,\Delta u(m-1+k|k),\epsilon} \Bigg(\sum_{i=0}^{p-1} \Big(& \\
&\sum_{j=1}^{n_y} \left| w^y_{i+1,j}(y_j(k+i+1|k) - r_j(k+i+1)) \right|^2 + && \text{(7.15a)}\\
&\sum_{j=1}^{n_u} \left| w^{\Delta u}_{i,j} \Delta u_j(k+i|k) \right|^2 + && \text{(7.15b)}\\
&\sum_{j=1}^{n_u} \left| w^u_{i,j}(u_j(k+i|k) - u_{j,\text{target}}(k+i)) \right|^2 \Big) + && \text{(7.15c)}\\
&\rho_\epsilon \epsilon^2 \Bigg). && \text{(7.15d)}
\end{aligned}
$$

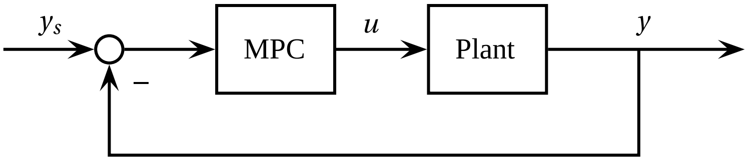
p is the prediction horizon and n_y and n_u is the dimensions of the output and input vectors respectively. The optimization problem minimizes the objective function using the input deviations from the present time step k to $m-1+k$ as optimization variables. m is the *control horizon*, which is not necessarily the same as the prediction horizon as assumed in Chapter 6. The last term in the optimization problem (Equation (7.15d)) arises because the use of soft constraints which was presented in Chapter 6. Hence, the variable ρ_ϵ is just a weighing variable for the slack variable ϵ.

Estimation

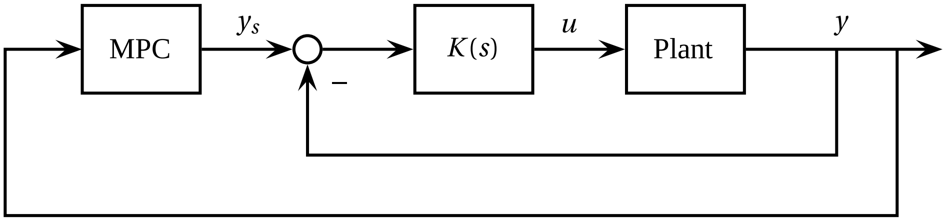
The unknown variables from the model given in Equation (7.14) needs to be estimated in order to obtain good future predictions. MPC Toolbox estimates three different states, the plant states, the disturbance model states and the measurement noise model states, see Figure 7.3. Let the disturbance model states be $x_{d,k} = \begin{bmatrix} x_{du,k} & x_{do,k} \end{bmatrix}^\top$, then the observer can be written as

$$
\begin{bmatrix} \hat{x}_k \\ \hat{x}_{d,k} \\ \hat{x}_{m,k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k-1} \\ \hat{x}_{d,k-1} \\ \hat{x}_{m,k-1} \end{bmatrix} + M(y_{m,k} - \hat{y}_{m,k}). \tag{7.16}
$$

Where the matrix M is the observer gain which is designed using Kalman filter theory.



(a) MPC controlling directly on the physical inputs of the plant.



(b) MPC controlling the setpoints that are given to the PI controllers.

Figure 7.4: Block diagram showing the two MPC approaches. $K(s)$ is a diagonal matrix with the four PI controllers. In Figure a) the setpoints y_s has the same values all the time.

MPC implementation with MPC Toolbox

The state-space model that was used in the implementation for prediction was the following linear distillation model,

$$x_{k+1} = Ax_k + Bu_k, \tag{7.17a}$$

$$y_k = Cx_k. \tag{7.17b}$$

Thus, the prediction model used was without the disturbance dynamics, represented by B_d in Equation (3.17).

For better estimation the same measurement array was used in the implementation as for the previous MPC.

MPC controlling setpoints of PI controllers

There can be several good reasons for letting the MPC adjust the PI controller setpoints instead of directly controlling the physical inputs. Putting the MPC on top of the PI controllers require less work on modeling and tuning of the MPC according to Skogestad (2004). Another reason is that if the MPC for some reason breaks down, the plant will still be controlled by the PI controllers. This was briefly discussed under the topic ”Control layers” in Chapter 6.

Figure 7.4 shows the difference between the previous MPC implementation and the current one. The inner feedback loop in Figure 7.4b reduces the effect of disturbances that occurs inside this loop, such that the MPC sees a more linear plant from the regulator control setpoints compared to the single layer MPC.

MPC Toolbox was chosen for implementation of this MPC as well. Therefore, the optimization problem and the estimation are performed the same way as for the previous MPC implementation. The prediction is however some different and is presented next.

Prediction model

The prediction model now becomes different because it has to include the dynamics of the PI controllers. The discrete state-space model describing the distillation column is written as

$$x_{k+1} = Ax_k + Bu_k, \tag{7.18a}$$

$$y_k = Cx_k. \tag{7.18b}$$

Again, we are considering a state-space model without the disturbance dynamics. The inputs u_k's are controlled by PI controllers. The discrete decentralized PI controller can be written as (Balchen et al., 2003)

$$u_k = u_{k-1} + G_0(y_{s,k} - y_k) + G_1(y_{s,k-1} - y_{k-1}). \tag{7.19}$$

Where G_0 and G_1 are matrices that contain the controller parameters. These matrices are derived in Appendix A. Inserting Equation (7.19) into Equation (7.18a) gives

$$\begin{aligned} x_{k+1} &= Ax_k + B\left(u_{k-1} + G_0(y_{s,k} - y_k) + G_1(y_{s,k-1} - y_{k-1})\right) \\ &= (A - BG_0C)x_k + Bu_{k-1} + BG_1(y_{s,k-1} - y_{k-1}) + BG_0 y_{s,k}. \end{aligned}$$

We want to make a new state-space model that can be used by the MPC. We want to measure u_k such that it is possible to add constraints to these physical inputs. Adds u_{k-1} and σ_k as new states, where σ_k is defined as

$$\sigma_k = y_{s,k-1} - y_{k-1}. \tag{7.20}$$

The new state-space model then becomes

$$\begin{bmatrix} x_{k+1} \\ \sigma_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A - BG_0C & BG_1 & B \\ -C & I & 0 \\ -G_0C & G_1 & I \end{bmatrix} \begin{bmatrix} x_k \\ \sigma_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} BG_0 \\ I \\ G_0 \end{bmatrix} y_{s,k}, \tag{7.21a}$$

$$\begin{bmatrix} y_k \\ u_k \end{bmatrix} = \begin{bmatrix} C & 0 & 0 \\ -G_0C & G_1 & I \end{bmatrix} \begin{bmatrix} x_k \\ \sigma_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ G_0 \end{bmatrix} y_{s,k}. \tag{7.21b}$$

When the physical inputs (L, R_L, S_1 and S_2) are added as measurements it results in *direct feed-through* from the input to the output for the MPC model. Hence, the corresponding transfer function matrix is *semi-proper*[1]. MPC Toolbox that was chosen for the implementation of this controller, does not support

[1] A semi-proper transfer function matrix $G(s)$ satisfy the property; $\lim_{\omega\to\infty} G(j\omega) \neq 0$ (Skogestad & Postlethwaite, 2005).

a prediction model with direct feed-through (see Equation (7.14)). Maciejowski (2002) presents a trick to avoid this problem by defining a new vector of controlled outputs,

$$\tilde{y}_k = y_k - Du_k. \tag{7.22}$$

This new output $\tilde{y}_k$ depends on the states only. This would however lead to quite many changes in the optimization problem solved by the MPC because the outputs are changed. Therefore, a much easier trick is done in order to avoid the problem of direct feed-through. A unit time delay is added to the measurements from the physical inputs. Hence, by measuring u_{k-1} instead of u_k, this direct feed-through is avoided. Thus, the measurements can now be written as

$$\begin{bmatrix} y_k \\ u_{k-1} \end{bmatrix} = \begin{bmatrix} C & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ \sigma_k \\ u_{k-1} \end{bmatrix} \tag{7.23}$$

Adding this small delay does not have any influence on the control performance since the delay is not comparable with the time constants of the column.

7.2 Simulation parameters

The simulation parameters used for simulations of the different MPCs are shown in Table 7.2, 7.3 and 7.4.

Some of the parameters presented in the tables are so-called tuning parameters for the MPCs. According to Maciejowski (2002) tuning parameters for an MPC is the *weighing matrices*, the *prediction horizon* and *the disturbance model and observer dynamics*. The MPC designed from scratch is the only MPC that uses a specified disturbance model in its prediction model.

Some tuning was required to make good responses and to ensure stability when the first MPC design was tested on the simulation model. A large input weighing was needed to obtain smooth responses for the controlled temperatures. By adding a reference trajectory for the output to follow as discussed in Chapter 6, the responses also became smoother. No considerable tuning were done for the other MPCs, only a smaller input weighing was required to obtain smooth responses for the supervisory MPC.

The prediction horizon was determined from the closed-loop responses using decentralized control shown in Chapter 3. For the various disturbances, the temperatures settle around 300 min after the occurring disturbance. Since the disturbance with a 50% change in the vapor split is much larger compared to the other disturbances tested (discussed in the end of Chapter 3), this disturbance response was not considered when the prediction horizon was determined. However, if the response typically settles after 300 min, an appropriate horizon length would be at least of the same length such that the constraints holds on the infinite horizon as pointed out in the stability section of Chapter 6. Therefore, the horizon length was chosen to be 500 min. The sampling time of the MPC was chosen to be 10 min, such that the horizon is 50 discrete time steps into the future.

Prediction model	
Order	20
Inputs	$u = \begin{bmatrix} L & S_1 & S_2 & R_L \end{bmatrix}^\top$
Outputs	$y = \begin{bmatrix} T_{17} & T_{30} & T_{59} & T_{49} & T_6 & T_{53} & T_{24} \end{bmatrix}^\top$
Uses disturbance model	Yes
Tuning parameters	
Prediction horizon	$N = 50$
Input weighing	$\text{diag}(R) = 10^6 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$
Output weighing	$\text{diag}(Q) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$
Reference trajectory specification	$\beta = 0.5$
Constraint specifications	
Minimum input	$u_{\min} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$
Maximum input	$u_{\max} = \begin{bmatrix} 3 & 1 & 1 & 1 \end{bmatrix}^\top$
Minimum change in input	$\Delta u_{\min} = -10 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^\top$
Maximum change in input	$\Delta u_{\max} = 10 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^\top$

Table 7.2: Parameters for the MPC that was constructed from the presented theory. There were no specified constraints for the outputs.

Prediction model	
Order	20
Inputs	$u = \begin{bmatrix} L & S_1 & S_2 & R_L \end{bmatrix}^\top$
Outputs	$y = \begin{bmatrix} T_{17} & T_{30} & T_{59} & T_{49} & T_6 & T_{53} & T_{24} \end{bmatrix}^\top$
Uses disturbance model	No
Tuning parameters	
Prediction horizon	$p = 50$
Control horizon	$m = 50$
Input weighing	$w^u_{i,j} = 0, \ \forall \ i = 0, \dots, p-1, \ j = 1, \dots, 4$
Input change weighing	$w^{\Delta u}_{i,j} = 1, \ \forall \ i = 0, \dots, p-1, \ j = 1, \dots, 4$
Output weighing	$w^y_{i+1,j} = 1, \ \forall \ i = 0, \dots, p-1, \ j = 1, \dots, 4$
	$w^y_{i+1,j} = 0, \ \forall \ i = 0, \dots, p-1, \ j = 5, \dots, 7$
Slack variable weighing	$\rho_\epsilon = 10^5$
Constraint specifications	
Minimum input	$u_{\min} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$
Maximum input	$u_{\max} = \begin{bmatrix} 3 & 1 & 1 & 1 \end{bmatrix}^\top$
Minimum change in input	$\Delta u_{\min} = -10 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^\top$
Maximum change in input	$\Delta u_{\max} = 10 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^\top$

Table 7.3: Parameters for the single layer MPC made with MPC Toolbox.

Prediction model	
Order	28
Inputs	$u = \begin{bmatrix} T_{17,\mathrm{s}} & T_{30,\mathrm{s}} & T_{59,\mathrm{s}} & T_{49,\mathrm{s}} \end{bmatrix}^\top$
Outputs	$y = \begin{bmatrix} T_{17} & T_{30} & T_{59} & T_{49} & L & R_L & S_1 & S_2 \end{bmatrix}^\top$
Uses disturbance model	No
Tuning parameters	
Prediction horizon	$p = 50$
Control horizon	$m = 50$
Input weighing	$w^{u}_{i,j} = 0,\ \forall\ i = 0,\ldots,p-1,\ j = 1,\ldots,4$
Input change weighing	$w^{\Delta u}_{i,j} = 0.01,\ \forall\ i = 0,\ldots,p-1,\ j = 1,\ldots,4$
Output weighing	$w^{y}_{i+1,j} = 1,\ \forall\ i = 0,\ldots,p-1,\ j = 1,\ldots,4$
	$w^{y}_{i+1,j} = 0,\ \forall\ i = 0,\ldots,p-1,\ j = 5,\ldots,8$
Slack variable weighing	$\rho_\epsilon = 10^5$
Constraint specifications	
Minimum output	$y_{\min} = \begin{bmatrix} -\infty & -\infty & -\infty & -\infty & 0 & 0 & 0 & 0 \end{bmatrix}^\top$
Maximum output	$y_{\max} = \begin{bmatrix} \infty & \infty & \infty & \infty & 3 & 1 & 1 & 1 \end{bmatrix}^\top$

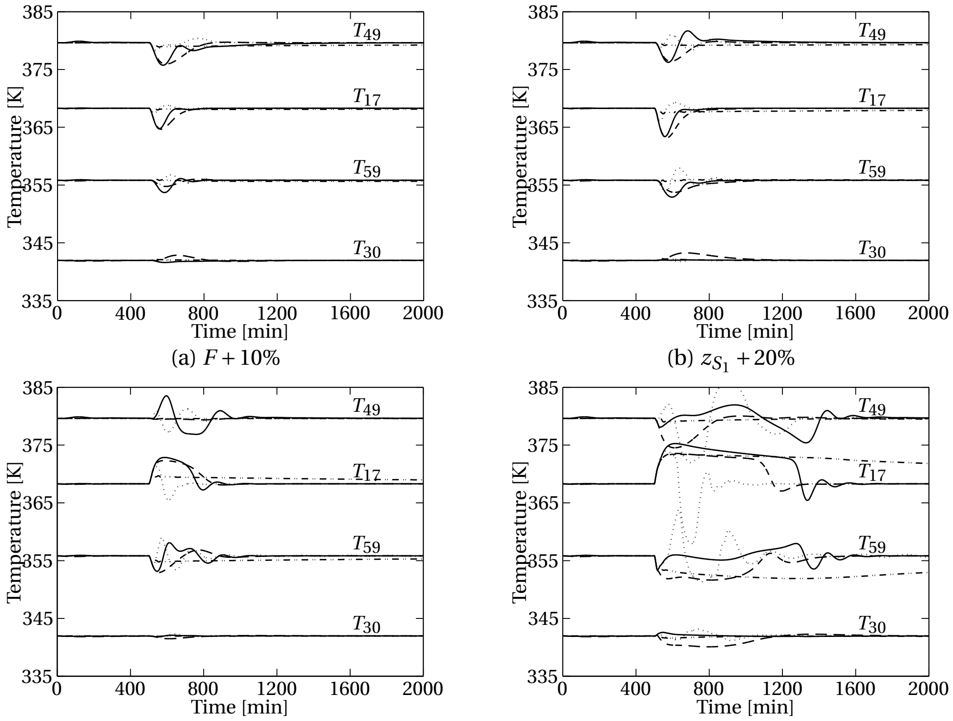
Table 7.4: Parameters for the MPC combined with PI-control, MPC Toolbox was used for this implementation. No constraints were specified for the inputs. $T_{xx,\mathrm{s}}$ means the setpoint for the PI controller controlling temperature xx.

7.3 Simulations

This section provides some dynamic responses when the column is controlled by the different MPCs. First, the introduced disturbances are tested and compared with the decentralized controller. A simulation with a change of the setpoint value for one of the controlled temperatures is presented next, followed by a simulation where vapor bypassing is present.

Disturbance simulations

The different MPCs were tested using the same disturbances as presented in Chapter 3. The corresponding results from decentralized control are also shown in the figures (Figure 7.5 and 7.6).



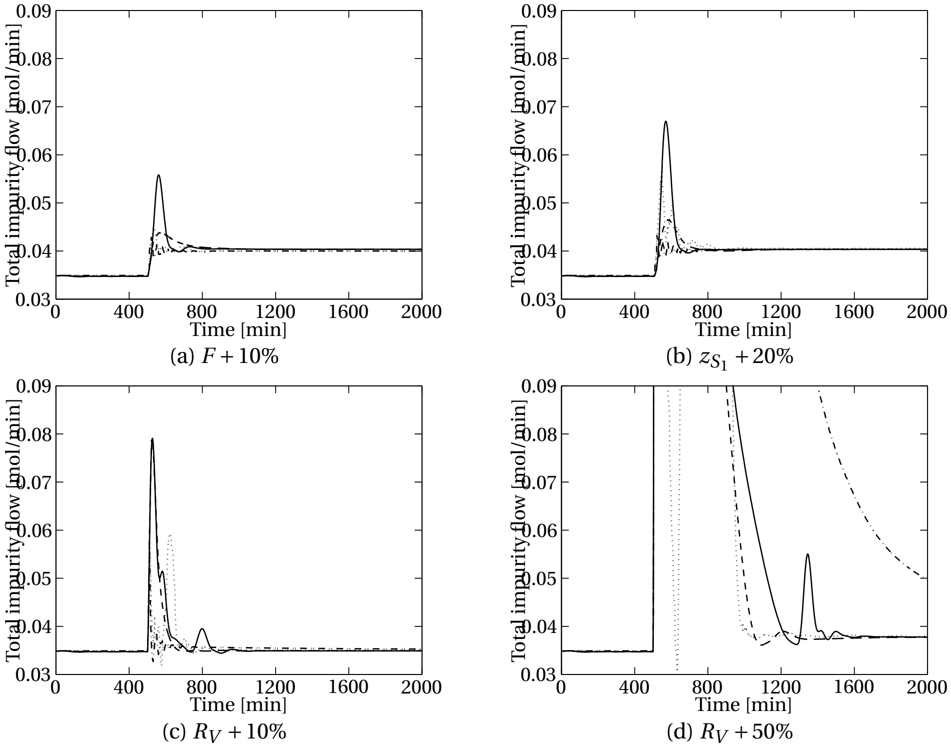
(a) $F + 10\%$

(b) $z_{S_1} + 20\%$

(c) $R_V + 10\%$

(d) $R_V + 50\%$

Figure 7.5: Controlled temperatures. Tuned decentralized PI controllers (solid), single layer MPC constructed from presented theory (dashed), single layer MPC made in MPC Toolbox (dash-dotted), MPC combined with PI-control (dotted).



(a) $F + 10\%$

(b) $z_{S_1} + 20\%$

(c) $R_V + 10\%$

(d) $R_V + 50\%$

Figure 7.6: Total impurity flow. Tuned decentralized PI controllers (solid), single layer MPC constructed from presented theory (dashed), single layer MPC made in MPC Toolbox (dash-dotted), MPC combined with PI-control (dotted).

Setpoint change

Figure 7.7 shows a setpoint change for the controlled temperature T_{17} at time 500 min. The temperature responses are plotted for the three MPCs and the decentralized controller. It can be seen that the MPCs reduce the interactions since they give less response amplitudes for the other temperatures (especially T_{49} and T_{59}). All controllers give quite similar responses for the temperature with setpoint change, T_{17}.

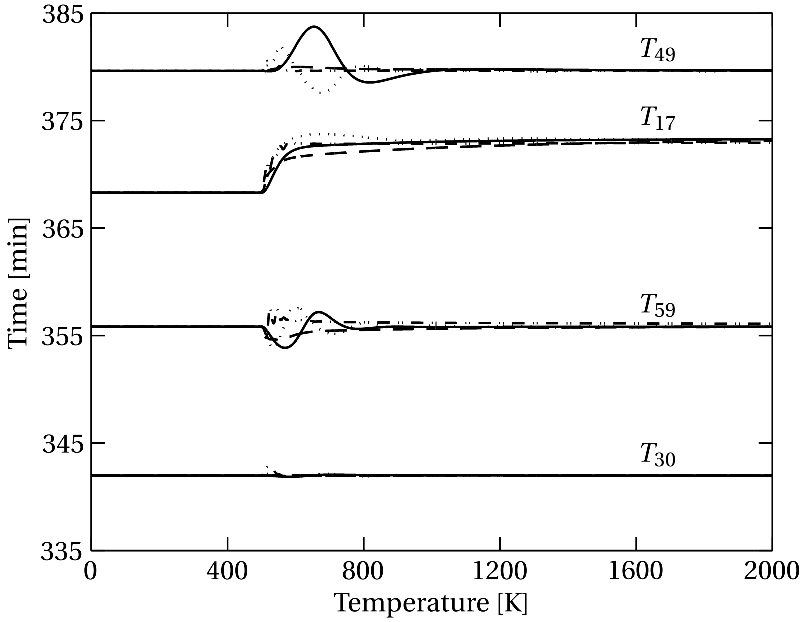
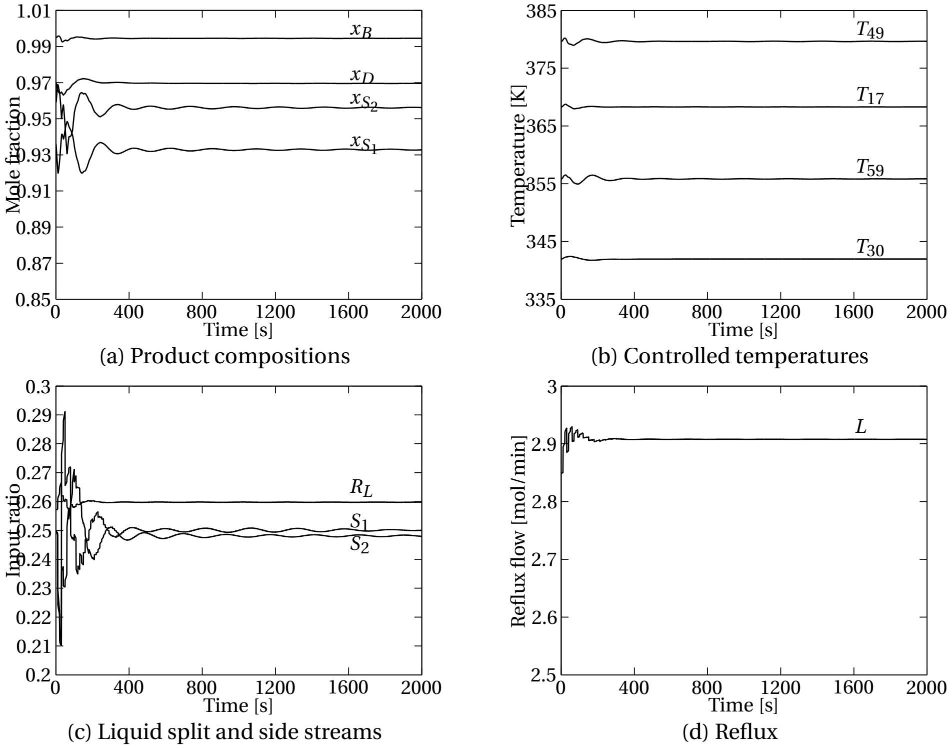


Figure 7.7: Setpoint change for T_{17} (+5 K) at time 500 min. Tuned decentralized PI controllers (solid), single layer MPC constructed from presented theory (dashed), single layer MPC made in MPC Toolbox (dash-dotted), MPC combined with PI-control (dotted).

Simulations with vapor bypassing

The MPCs were tested with vapor bypassing. They gave very similar responses and therefore responses of only one of the MPCs are plotted in Figure 7.8. The simulation is similar to the one shown in Figure 5.4, where the effects of vapor bypassing were tested in closed-loop with decentralized control. If we compare with this similar decentralized controller simulation, we can see that the MPC gives less response amplitudes for the purities and temperatures.



(a) Product compositions

(b) Controlled temperatures

(c) Liquid split and side streams

(d) Reflux

Figure 7.8: Effect of vapor bypassing in closed-loop with supervisory MPC. The vapor bypassing factor was 0.02 in the simulation.

7.4 Sensitivity of model errors

A controller must be *robust*. Skogestad & Postlethwaite (2005) has the following definition:

> *A control system is robust if it is insensitive to differences between the actual system and the model of the system which was used to design the controller.*

Robustness of an MPC can refer to three different things:

- Robust feasibility when the process is exposed to disturbances
- Robust performance when the process is exposed to disturbances
- Robust performance in spite of model errors

The first item deals with the issue of preventing an infeasible optimization problem when the process experiences large disturbances. *Constraint relaxing* discussed in Chapter 6 is a method to avoid infeasibility in such scenarios. It is the last item in the list above that is the motivation for this chapter. The MPC must be able to handle differences between its internal prediction model and the real plant model. The meaning of this section is to test the performance of the MPC for different model errors. The developed single layer MPC is compared to the MPC that works as supervisory control for the PI controllers. It is the MPCs that were made by use of MPC Toolbox in MATLAB that are compared. Similar results from the decentralized controller are also added for comparison.

Simulation testing versus theoretical analysis

A good theoretical tool for robustness analysis for multivariable systems is the *structured singular value* (μ-analysis) (Zhou et al., 1996; Skogestad & Postlethwaite, 2005). It can be used to test a closed-loop system for *robust stability* and *robust performance*. By robust stability we mean that the system remains stable for all plants in the specified set of uncertainties. If the controlled system has robust performance it also satisfies some specified performance requirements in addition to be robust stable.

Such a theoretical analysis is an elegant solution to a sensitivity analysis, but the μ-analysis needs a linear plant function $G(s)$ and a controller matrix $K(s)$. The Kaibel distillation model is nonlinear and an MPC with constraints can not be written as a matrix $K(s)$. Bemporad & Morari (1999) also concludes that an analytical approach to MPC and robustness does not exist yet. Therefore a *simulation test* approach is more suitable for this part of the work, even though it require many time-consuming computer simulations. By comparing a set of simulations with model errors, different controllers can be tested against each other.

We want to test a controller for robust performance, i.e. the controller must satisfy some kind of specified constraints in an uncertainty set in order to fulfill a performance requirement.

Measurement of control performance

The optimal temperature setpoints that is used to control the distillation column were found by minimization of the total impurity flow;

$$J = D(1 - x_D) + S_1(1 - x_{S_1}) + S_2(1 - x_{S_2}) + B(1 - x_B). \tag{7.24}$$

This objective function can also be used as the control performance measurement. If we integrate this impurity flow over a simulation, and divide with the length of the simulation T_{sim}, the *average impurity flow* for the simulation can be found,

$$J_{\text{avg}} = \frac{1}{T_{\text{sim}}} \int \big(D(1 - x_D) + S_1(1 - x_{S_1}) + S_2(1 - x_{S_2}) + B(1 - x_B)\big)\mathrm{d}t. \tag{7.25}$$

Hence, if the average impurity flow is large, the control performance is poor. This impurity flow value is a very simple measurement of controller performance. E.g. the controlled temperatures can oscillate heavy around their setpoints, still giving an acceptable value of the impurity flow.

Control performance requirement

A control performance requirement should be specified in order to determine whether a simulation fulfils performance criteria or not. It is not possible to specify a maximum constraint for the impurity flow because it depends how the product purities are specified for the actual plant. If all product purities are supposed to be greater than 90%, it results in a maximum value of 0.1 mol/min for the impurity flow since the nominal feed flow is 1.0 mol/min. Strandberg & Skogestad (2006) has used the following specifications; $x_D \geq 0.975$, $x_{S_1} \geq 0.94$, $x_{S_2} \geq 0.94$ and $x_B \geq 0.975$. Using these specifications give a corresponding impurity flow of 0.0425 mol/min, which is probably a too strict performance criteria since the *average* impurity flow after a disturbance simulation is more than in nominal operation.

Input uncertainty

Robust performance can be tested by *input* uncertainty. That means the plant real input is not exactly the same as the controller output. This is illustrated in Figure 7.9. u is the controller output, but the plant input is $(I + W_I)u$, where W_I is a diagonal matrix which elements decide the input gain error for the different inputs;

$$\text{diag}(W_I) = \begin{bmatrix} w_{I,L} & w_{I,S_1} & w_{I,S_2} & w_{I,R_L} \end{bmatrix}. \tag{7.26}$$

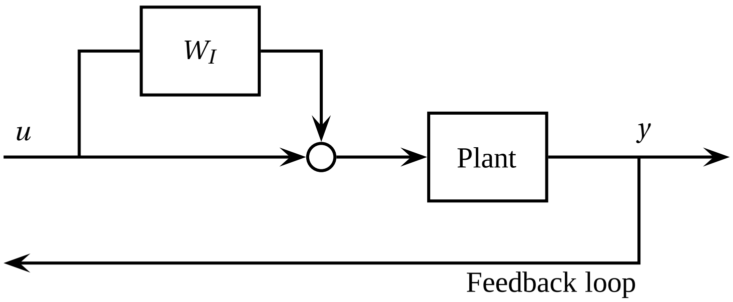


Figure 7.9: Input gain error.

Simulation number	$w_{I,L}$	w_{I,R_L}	w_{I,S_1}	w_{I,S_2}
1	−0.1	−0.1	−0.1	−0.1
2	−0.1	−0.1	−0.1	0.0
3	−0.1	−0.1	−0.1	0.1
⋮	⋮	⋮	⋮	⋮
40	0.0	0.0	0.0	−0.1
41	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.1
⋮	⋮	⋮	⋮	⋮
79	0.1	0.1	0.1	−0.1
80	0.1	0.1	0.1	1.0
81	0.1	0.1	0.1	0.1

Table 7.5: Input uncertainty simulation experiments. Simulation number 41 has no gain error and the corresponding impurity flow will be at its minimum value. A complete table is given in Appendix B.

Hence, when this matrix is a zero matrix, there are no gain errors.

The performance of a controller can be tested for different errors in the input gain, i.e. for different W_I's. The distillation model was therefore simulated with different combinations of input gain errors for the two different MPC approaches shown in Figure 7.4, and for the decentralized controller. The MPC simulation parameters used were the same ones as given in Table 7.3 and 7.4. All four inputs were tested with up to 10% gain error. Every combination was simulated in 81 different simulations shown in Table 7.5 (see Appendix B for a complete table). The simulations lasted for 500 min and the input gain error was constant during the whole simulation. No disturbances occurred during these simulations. Figure 7.10 show the average impurity flow (from Equation (7.25)) plotted for the different simulations. As it can be seen from Figure 7.10 it is clear that the supervisory MPC generally results in less impurity flows.

An interesting result from Figure 7.10 is that gain errors in the side streams do

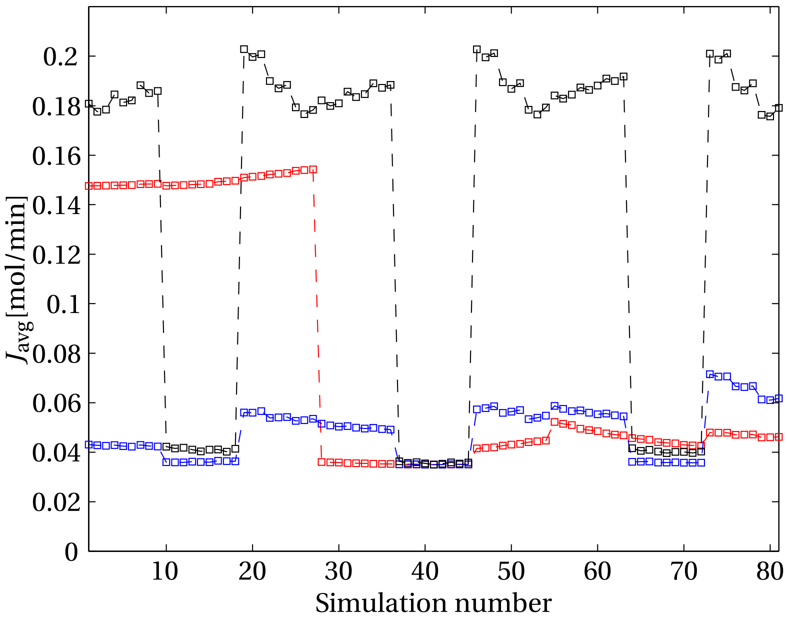


Figure 7.10: Values of the total impurity flow for different input gain errors. Decentralized control (black), single layer MPC (red), supervisory MPC (blue).

not affect the total setpoint error as much as the liquid split and the reflux flow. From the simulation table we have that the nine first simulations only changes the input gain for the side streams. In this interval (simulation 1 to 9) all controllers gives around the same impurity flow. The same happens for other intervals where only the side stream gains are changed (9 to 18, 18 to 27 etc).

Thus, the largest changes in the impurity flow come as a result of changes in reflux and the liquid split. Gain errors in the side stream inputs do not seem to affect the impurity to the same extent. For the decentralized controller it can be seen that it is important to keep the uncertainty in the liquid split low, since it is this input that makes greatest change on the impurity flow (see Figure 7.10 and Table 7.5).

To make a better illustration of the consequences of uncertainties in the reflux and the liquid split, some more simulation tests were performed. These simulations had no gain error in the side stream inputs. The reflux and liquid split ratio input gains were tested with up to 20% error ($w_L = -0.2\ldots0.2$ and $w_{R_L} = -0.2\ldots0.2$), with an input change of 4% for each simulation (gives totally 121 simulations). These simulations also lasted for 500 min. The results are shown in Figure 7.11 and 7.12. A similar plot is shown in Figure 7.13, this for the decentralized controller.

Also these plots show that the MPC in addition to PI-control results in a more robust performance, since the impurity flow function is *more flat* with respect to the input uncertainties. An interesting observation for the single layer MPC in Fig-

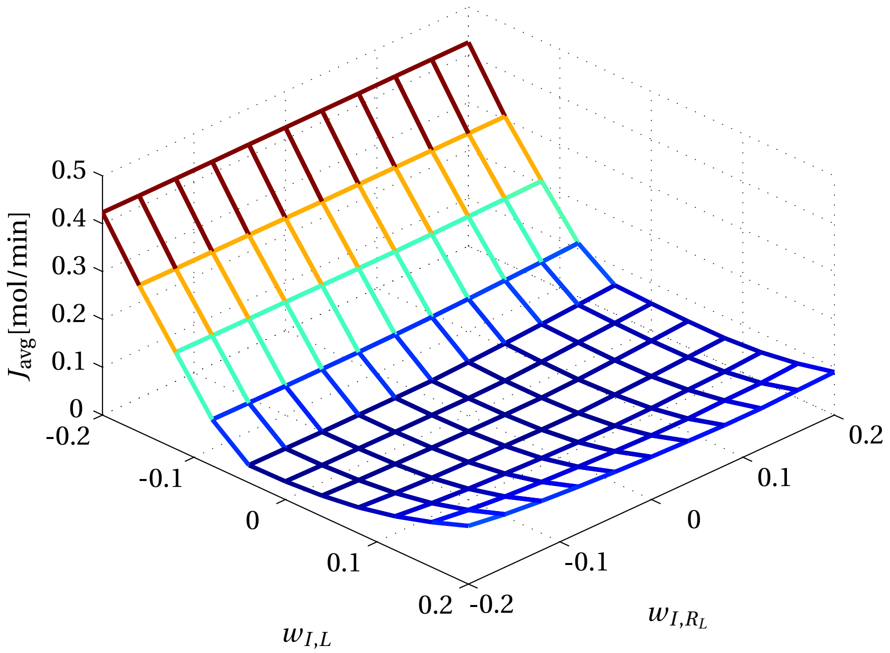


Figure 7.11: Single layer MPC.

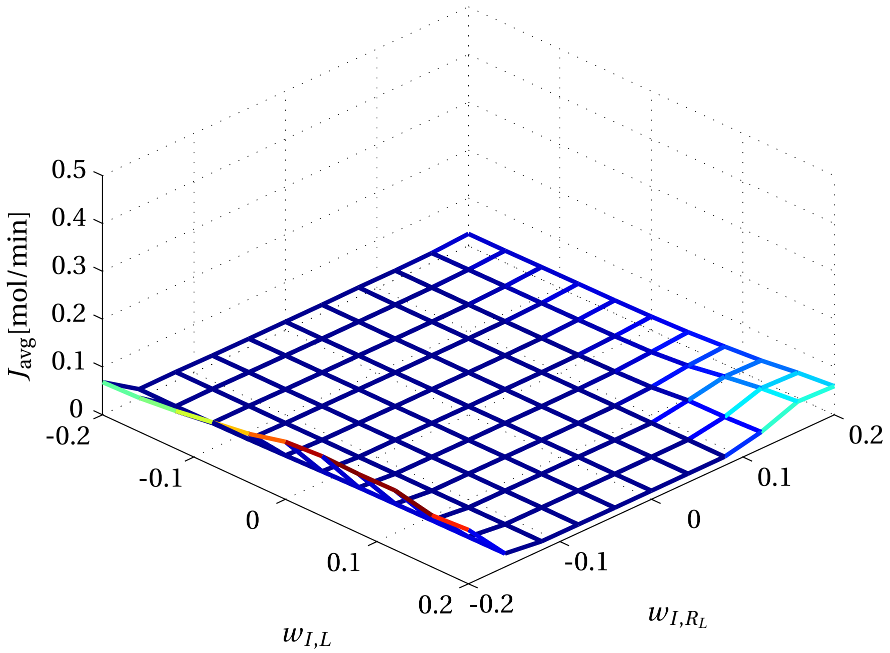


Figure 7.12: Supervisory MPC.

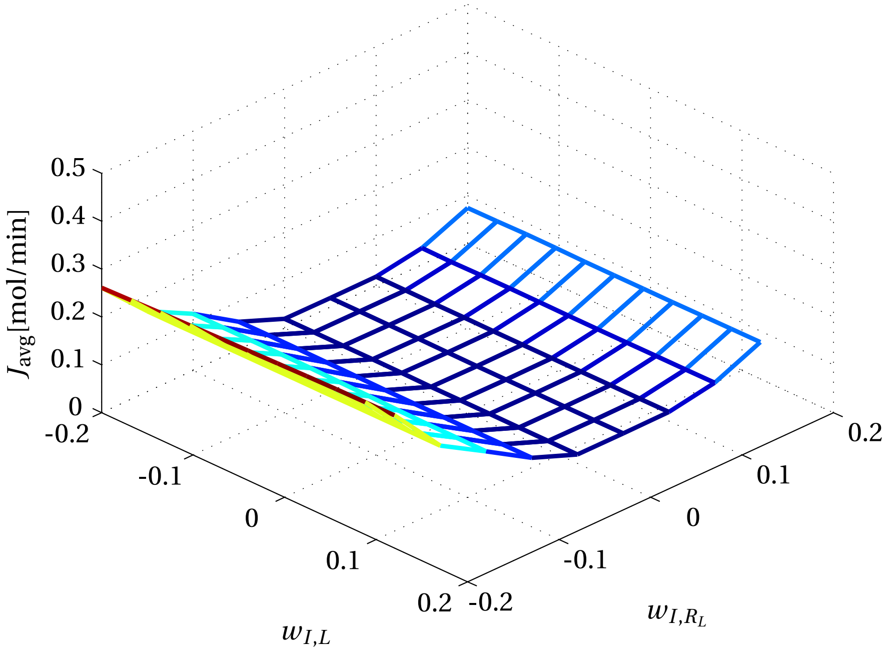


Figure 7.13: Decentralized PI-control. When $w_{I,L}$ was greater than 0.12 the simulations resulted in error caused by computational issues.

ure 7.11 is that the uncertainties in reflux results in more impurity flows than uncertainties in the liquid split. For PI-control (Figure 7.13), this is opposite. Without considering this, the figures show that the decentralized controller has a generally more flat impurity flow function than the single layer MPC when the uncertainties are as big as 20%. When the uncertainties were maximum 10%, the single layer MPC was more robust compared to the decentralized controller according to Figure 7.10.

Effect of time delay

The MPC must be able to handle some time delay in the feedback loop. The controllers were tested for different values of this time delay, and the results are shown in Table 7.6. The time delay was set to the same value for all measurements. The simulations lasted for 2000 min with an occurring disturbance at time 500 min.

We can see that the supervisory MPC achieves less impurity than the other MPC approach. As we can see from the table, the decentralized controller is actually the best controller when time delay is present.

		$F + 10\%$	$z_{S_1} + 20\%$	$R_V + 10\%$	$R_V + 50\%$
Single layer MPC					
Time delay [min]	0	0.0396	0.0347	0.0357	0.0899
	3	0.0563	0.0483	0.0460	0.0946
	5	0.0863	0.0729	0.0708	0.1110
	7	0.1255	0.1345	0.1189	0.1387
	10	0.1928	0.1875	0.1807	0.1780
	15	0.2490	0.2273	0.2358	0.3354
Supervisory MPC					
Time delay [min]	0	0.0396	0.0350	0.0357	0.0538
	3	0.0389	0.0400	0.0363	0.0737
	5	0.0403	0.0418	0.0373	0.0800
	7	0.0444	0.0472	0.0404	0.0926
	10	0.0532	0.0544	0.0538	0.1099
	15	0.0745	0.0749	0.0968	0.1466
Decentralized control					
Time delay [min]	0	0.0404	0.0360	0.0374	0.0728
	3	0.0442	0.0407	0.0379	0.0760
	5	0.0443	0.0408	0.0383	0.0770
	7	0.0444	0.0410	0.0389	0.0785
	10	0.0446	0.0413	0.0408	0.0812
	15	0.0451	0.0418	0.0471	0.0861

Table 7.6: Average impurity flow [mol/min] during disturbances for different time delays.

Robustness for different values of vapor bypass factor

Robust performance was also tested for different values of the vapor bypass factor introduced in Chapter 5. It is interesting to see how this new efficiency parameter affects the control performance. The Kaibel column model was simulated using different values of this bypass factor spanning from 0.010 up to 0.100. The results from these simulations are shown in Table 7.7. As explained in Chapter 5, increased vapor bypass is compensated by increased reflux flow by the controller. The controllers have a reflux flow saturation of 3 mol/min, which gives very poor performance when the bypass factor is too high (above 0.1). Hence, with the specified constraint, the control problem becomes infeasible. The compared controllers give approximately the same impurity flows for different bypass factors. The supervisory MPC gives slightly better performance for large bypassing factors.

		$F+10\%$	$z_{S_1}+20\%$	$R_V+10\%$	$R_V+50\%$
Single layer MPC					
Bypass factor	0.000	0.0396	0.0347	0.0357	0.0899
	0.010	0.0389	0.0390	0.0355	0.0936
	0.020	0.0390	0.0391	0.0355	0.0921
	0.050	0.0390	0.0392	0.0354	0.0899
	0.100	0.0531	0.0502	0.0489	0.1088
Supervisory MPC					
Bypass factor	0.000	0.0396	0.0350	0.0357	0.0538
	0.010	0.0389	0.0396	0.0360	0.0713
	0.020	0.0391	0.0397	0.0362	0.0713
	0.050	0.0400	0.0403	0.0372	0.0755
	0.100	0.0607	0.0578	0.0570	0.0699
Decentralized control					
Bypass factor	0.000	0.0404	0.0360	0.0374	0.0728
	0.010	0.0400	0.0408	0.0381	0.0757
	0.020	0.0407	0.0414	0.0388	0.0764
	0.050	0.0449	0.0454	0.0434	0.0774
	0.100	0.0680	0.0656	0.0649	0.0978

Table 7.7: Average impurity flow [mol/min] during disturbances for different values of the vapor bypass factor.

CHAPTER 8

EVALUATION OF ALTERNATIVE MPC APPROACHES

This chapter presents a brief discussion on other MPC approaches that can be used to control the distillation column. First, different optimization objectives are discussed followed by a discussion on replacing temperature with product compositions as controlled variables such that an MPC can control the purities instead of temperatures. After that, three alternative MPC designs, EMPC, NMPC and QDMC, are briefly discussed for control of the Kaibel distillation column.

8.1 Optimization objectives

The implemented MPC was based on setpoints found by minimization of the total impurity flow from the distillation column. Strandberg (2009) presents other objectives that can be optimized with respect to:

- Minimize energy usage
- Maximize profit
- Maximize throughput

The first item has an objective function equal to the total column energy consumption, together with minimum constraints for the product purities. The second item represents a more overall optimization objective, an economic objective, which is probably the most important for an industrial company that operates the column. Maximization of throughput is often relevant since increased feed flow into the column results in increased profit (Aske, 2009). Thus, the objective function will be the negative feed flow, together with the purity constraints.

8.2 Control of product compositions

For the implemented MPC combined with PI-control, both controllers (MPC and PI controller) use the same measurements, the temperature measurements. Hence, the MPC's control actions are computed from the objective of keeping the controlled temperatures close to their respective setpoints. It is possible to let the MPC compute optimal setpoints for the PI controllers by using the real objective function, e.g. the total impurity flow. By using this approach, the product compositions must be known or estimated.

As mentioned earlier compositions can be measured by use of a gas chromatograph, but it will add a measurement delay of 10-20 minutes (Mejdell & Skogestad, 1991). It is possible to estimate the compositions by an estimation algorithm. The measurement that is often available on a distillation column is temperature measurements. If all temperature measurements are collected in a vector θ it is possible to obtain a linear estimator that gives estimates of the compositions (Mejdell & Skogestad, 1991):

$$\begin{bmatrix} \hat{x}_D \\ \hat{x}_{S_1} \\ \hat{x}_{S_2} \\ \hat{x}_B \end{bmatrix} = K\theta. \tag{8.1}$$

The matrix K can be found using e.g. *Principal Component Regression* (PCR) or *Partial Least Squares Regression* (PLS) (described in Mejdell & Skogestad (1991)).

Figure 8.1 shows the block diagram of such an implementation.

By knowing the purities the MPC is able optimize over the real objective; the total impurity flow. Let the impurity flow be denoted J, then a possible optimization problem in the MPC can be

$$\begin{aligned} &\min J \\ &\text{subject to} \\ &x_D \geq 0.90, \\ &x_{S_1} \geq 0.90, \\ &x_{S_2} \geq 0.90, \\ &x_B \geq 0.90. \end{aligned}$$

The specified minimum constraints for the purities are of course different for different plants. Other constraints that also needs to be added to the optimization problem above is the distillation model itself, and other minimum and maximum constraints, e.g. for the inputs.

The principle of using regulatory layer controlled measurements for estimation of uncontrolled outputs that is related to the optimization objective is referred to as ”inferential control” in Skogestad & Postlethwaite (2005).

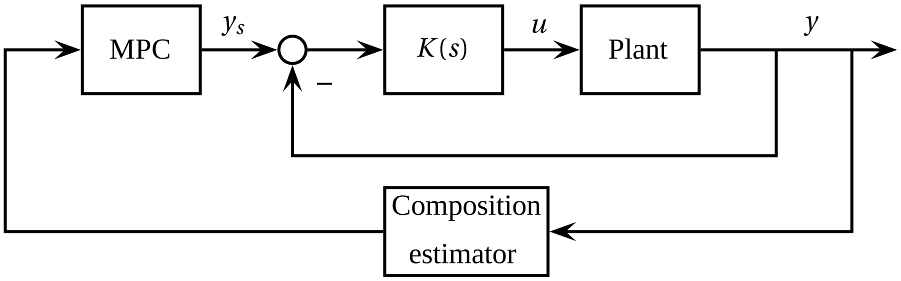


Figure 8.1: MPC on top of PI-control with a composition estimator.

8.3 Other MPC designs

DMC, EMPC and NMPC were introduced in Chapter 6 and a brief discussion of these MPC alternatives is done here.

Dynamic matrix control

An ordinary DMC does not handle constraints in inputs and outputs, but the ”extended” version, the quadratic dynamic matrix controller (QDMC), handles constrained control. A QDMC uses step response coefficients instead of a state-space model, as an MPC does. Ying & Joseph (1999) argues that QDMCs suffers of steady state errors, inconsistent setpoints and does not take into account economic objectives.

Because of the need of step responses, a QDMC is most appropriate for temperature control in a distillation column, and not other objectives.

Nonlinear model predictive control

An MPC based on a nonlinear prediction model (NMPC) should be considered if a linear MPC does not meet some specified performance criteria. A distillation column can be very ill-conditioned (Skogestad, 1997), i.e. be highly nonlinear, and it may be appropriate to use a nonlinear MPC. The drawback of an NMPC is its complexity; much time can be spent on implementation issues. An MPC combined with a decentralized controller can therefore be preferred before an NMPC. The inner decentralized control loop linearizes the plant as it is seen from the MPC (Skogestad & Postlethwaite, 2005), such that a nonlinear plant can be close to a linear plant the setpoints of the decentralized controller to the outputs.

There is another common trick to make the plant more linear as it is seen from the MPC. It is possible to perform a logarithmic transformation of the measurements to obtain more linear behavior. This is discussed in Kvernland (2008).

Explicit model predictive control

Explicit model predictive control (EMPC) was also briefly discussed in the end of Chapter 6 together with NMPC. The simpler implementation of an EMPC is an advantage compared to an ordinary MPC. However, the main motivation for using EMPC is its speed, which is good for processes with dominating time constants less than the time it takes to solve the optimization problem. For slow processes as distillation plants this is unlikely, hence this factor is not a motivation for this project.

Online tuning is not possible for an EMPC because the pre-computed look-up table must be recomputed for each change in the objective function or in the constraints.

Therefore, the only motivation for using an EMPC to control distillation arrangements is its simple implementation by a single look-up table and probable a state estimator. To the author's knowledge there does not exist any literature on EMPC of distillation columns.

CHAPTER 9

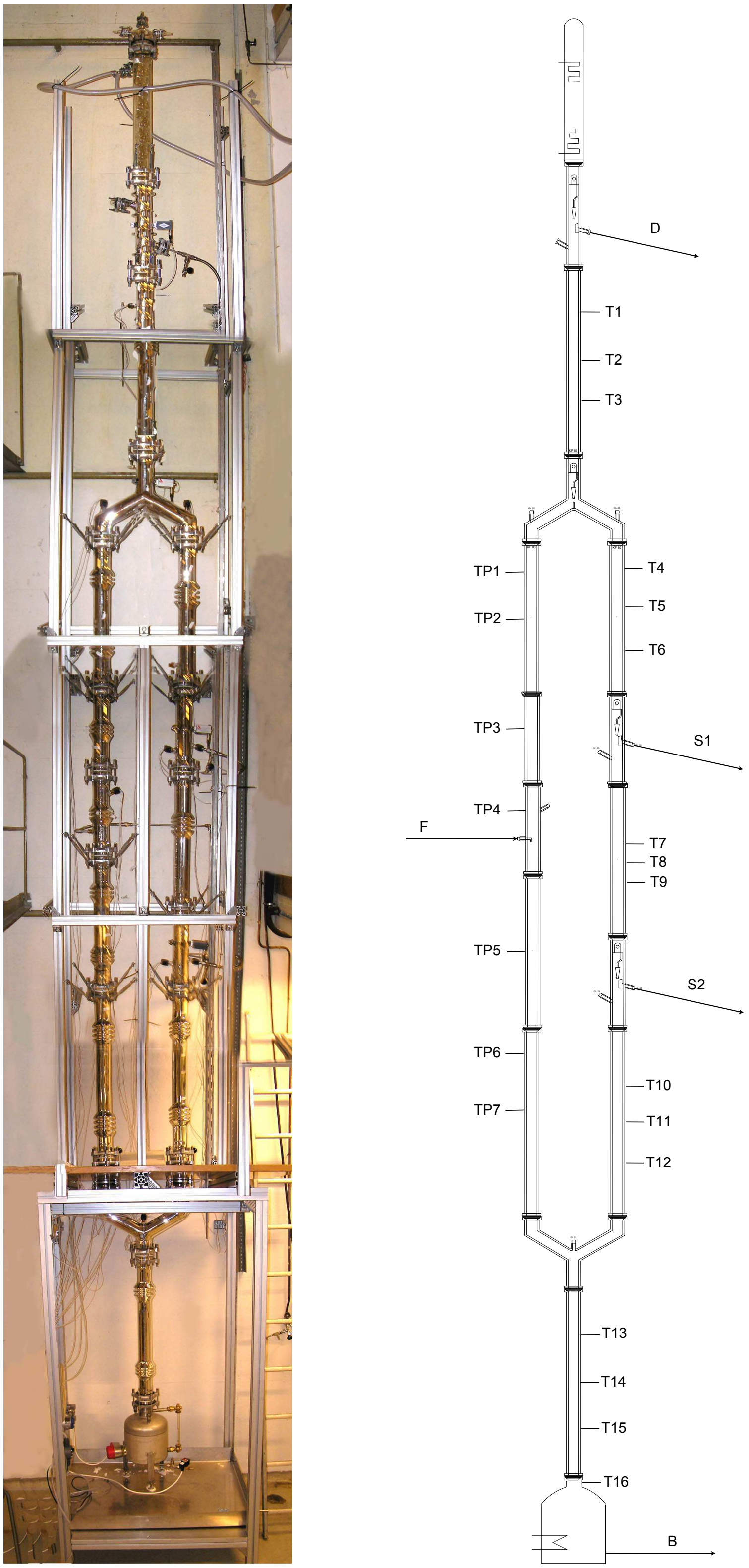
APPROACH FOR MPC IMPLEMENTATION AT THE LABORATORY COLUMN

The work done so far is hopefully important results for further work on the Kaibel distillation column and later industrial use. Another important thing is to do a practical implementation of the suggested MPCs such that better control performance compared to decentralized control can be concluded or not. Unfortunately, the laboratory column has not been available for experiments during this work. Therefore, the meaning of this chapter is to present the laboratory column at Department of Chemical Engineering, and how an MPC implementation should be done.

The chapter starts with a brief introduction to the laboratory column. Next, some practical issues regarding implementation of an MPC are presented. A description of the software used to implement an MPC is done followed by some suggestions to experiments that need to be performed on the column for implementation. The chapter ends with an implementation summary which lists the things to be done before the MPC can be tested experimentally.

9.1 Laboratory distillation column

The laboratory Kaibel distillation column was briefly introduced Chapter 2. A picture of the column and a scaled drawing are shown in Figure 9.1. The drawing shows the location of the temperature sensors installed to the column and the four product outlets, D, S_1, S_2 and B, where methanol, ethanol, propanol and butanol are drained respectively. The laboratory column does not have any reflux hold-up tank, which means that the condensed liquid from the condenser in the top is separated directly into distillate product and reflux continuously.



(a) Picture of pilot plant.

(b) Scaled drawing of pilot plant.

Figure 9.1: Pilot plant at Department of Chemical Engineering (Strandberg, 2009).

The amount of reflux versus the amount of distillate is determined by a valve in the top of the column. The liquid split is also a simple valve which determines how much liquid flow that goes into the prefractionator compared to the main column. The column arrangement has totally five storage tanks, four for the products and one for holding the feed. The reboiler is also actually a tank containing a heating element.

Valves

The valve construction is shown in Figure 9.2. A pulse-width signal operates an electromagnet that control a swinging funnel which determines where the liquid flow from the column section above goes. The time period of the pulse-width signal is set in the computer interface that is made for operation of the column. The input variable from the computer interface that controls the valve is between 0 and 1. A value of 1 means that the valve is closed; hence the product flow is zero. If the input value is 0.2 and the pulse-width period is 10 s, the swinging funnel will stay in the same position as in the figure for 8 s and 2 s in a position where the liquid passes the product outlet.

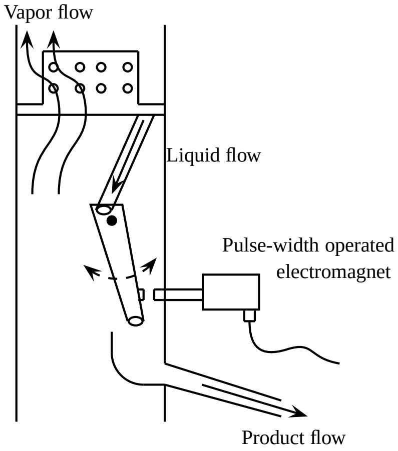


Figure 9.2: Valve construction.

If the prediction model in the MPC is based on a parameter adjusted version of the simulation model, the model inputs needs to be mapped into the inputs available in the computer interface. This issue is discussed in Section 9.2.

Column dynamics

The column is similar to the simulation model when it comes to the pressure inside the column. It is constructed with an open vent, such that the pressure inside goes to atmospheric pressure.

The laboratory column is a *packed column*[1], and does not have physical stages. The simulation model uses a specified number of stages which can work as an approximation of a packed column.

It was briefly mentioned in Chapter 3 that the equivalent size of the simulation column is larger than the laboratory column. Thus, the laboratory column has faster dynamic responses compared to the simulation column.

[1] A packed column is packed continuously with some kind of packing material instead of physical stages. The laboratory column is packed with so-called *Raschig rings*, which are small glass cylinders.

Decentralized control

As mentioned earlier, the column has earlier been controlled with a decentralized controller consisting of four PI controllers. Since, the distillate product flow is determined by the same valve that controls the reflux there is no separate controller for this flow. The bottom product flow is just set to a constant value. Hence, the four PI controllers control the top product flow, the liquid split and the two side streams. Temperature sensor T3 (see Figure 9.1b) has earlier been used for control of the top product flow, TP5 for the liquid split, T7 for side stream 1 and T13 for side stream 2.

The alcohols used in the feed have the following boiling temperatures (t_b) (Aylward & Findlay, 2008)

$$\begin{bmatrix} t_{b,\text{methanol}} \\ t_{b,\text{ethanol}} \\ t_{b,\text{propanol}} \\ t_{b,\text{butanol}} \end{bmatrix} = \begin{bmatrix} 64.7\,^\circ\text{C} \\ 78.3\,^\circ\text{C} \\ 97.3\,^\circ\text{C} \\ 117.7\,^\circ\text{C} \end{bmatrix} = \begin{bmatrix} 337.9\,\text{K} \\ 351.5\,\text{K} \\ 370.5\,\text{K} \\ 390.9\,\text{K} \end{bmatrix}. \tag{9.1}$$

It is desirable to drain a distillation product the same place where the column temperature is close to the product boiling temperature, therefore the controlled temperature setpoints must be chosen such this is fulfilled. The controlled temperatures are all below the corresponding product outlet (D, S_1 and S_2), which means that the controlled temperatures must be a little higher than the boiling temperatures since the temperatures decreases upwards in the column. In earlier laboratory experiments these setpoints have been used

$$y_{\text{ref}} = \begin{bmatrix} \text{T3}_{\text{ref}} \\ \text{TP5}_{\text{ref}} \\ \text{T7}_{\text{ref}} \\ \text{T13}_{\text{ref}} \end{bmatrix} = \begin{bmatrix} 72.0\,^\circ\text{C} \\ 83.0\,^\circ\text{C} \\ 86.0\,^\circ\text{C} \\ 105.0\,^\circ\text{C} \end{bmatrix} = \begin{bmatrix} 345.2\,\text{K} \\ 356.2\,\text{K} \\ 359.2\,\text{K} \\ 378.2\,\text{K} \end{bmatrix}. \tag{9.2}$$

9.2 Model predictive controller

An MPC should improve temperature responses compared to the existing decentralized control structure as we saw in Chapter 7.

Two different MPC approaches are emphasized in this thesis, an MPC that controls the physical inputs directly and an MPC that operates via a regulatory control layer. The MPC combined with PI-control indicated better performance than the single-layer MPC in the simulations and sensitivity analysis done in Chapter 7. However, both alternatives are discussed here as possible laboratory implementations.

Inputs

If the prediction model is made from a fitted version of the simulation model, i.e. grey box identification, the optimal inputs calculated by the MPC needs to be mapped into the inputs that are available in the computer interface controlling the column. This is however *not* necessary for the supervisory MPC approach, because the optimal inputs calculated are setpoints to the PI controllers, which are the same ones as in the simulation model.

The liquid split valve is in fact implemented the same way as in the simulation model, i.e.

$$u_{R_L} = R_L, \qquad u_{R_L} \in [0,1], \tag{9.3}$$

where u_{R_L} is the computer interface input. For the distillate flow and the side streams things becomes some different because of the valve construction like presented in Section 9.1. These mappings are as follows,

$$u_D = 1 - \frac{D}{L}, \qquad u_D \in [0,1], \tag{9.4}$$

$$u_{S_1} = 1 - \frac{S_1}{L_{4,\text{bottom}}}, \qquad u_{S_1} \in [0,1], \tag{9.5}$$

$$u_{S_2} = 1 - \frac{S_2}{L_{7,\text{bottom}}}, \qquad u_{S_2} \in [0,1]. \tag{9.6}$$

Where $L_{i,\text{bottom}}$ means the total liquid flow at the bottom stage of column section i, where the column section numbers are defined in Figure 3.5. These liquid flows are not measured and therefore they need to be estimated. The reflux (L) comes from the MPC, and does not need to be estimated. The nominal value from Table 3.2 can be used for the distillate product flow (D).

Estimation of liquid flows

If the input mappings given in Equation (9.5) and (9.6) are used in the MPC implementation, these liquid flows in column section 4 and 7 needs to be estimated. A simple estimate would be the following

$$\hat{L}_{4,\text{bottom}}(t) = (1 - R_L(t))L(t), \tag{9.7a}$$

$$\hat{L}_{7,\text{bottom}}(t) = \hat{L}_{4,\text{bottom}}(t) - S_1(t). \tag{9.7b}$$

That is, the estimated liquid flows do not take the time delay of the liquid to go from the top of the column down to the product outlet into consideration. It does neither take the effect of heat loss into consideration. Heat loss leads to increased liquid flow downwards the column because an amount of vapor is condensed at each stage as explained in Chapter 5. A more precise estimate of these liquid flows is therefore presented next.

Consider the liquid flow that comes in at the top of column section 4 at time t. An estimate of this flow can be

$$\hat{L}_{4,\text{top}}(t) = (1 - R_L(t))L(t - \theta_3) + L_{3,\text{heat loss}}(T_{S3}), \tag{9.8}$$

where θ_3 is the time the liquid uses to flow through column section 3. The extra amount of liquid that comes from column section 3 caused by heat loss is denoted

$L_{3,\text{heat loss}}(T_{\text{S3}})$. Note that this amount depends on the temperatures in column section 3, T_{S3} (see Equation (5.1)). Similar, estimates for the flows in the bottom of column section 4 and 7 can be found;

$$\hat{L}_{4,\text{bottom}}(t) = \hat{L}_{4,\text{top}}(t-\theta_4) + L_{4,\text{heat loss}}(T_{\text{S4}}), \tag{9.9a}$$

$$\hat{L}_{7,\text{bottom}}(t) = \hat{L}_{4,\text{bottom}}(t-\theta_7) - S_1(t-\theta_7) + L_{7,\text{heat loss}}(T_{\text{S7}}). \tag{9.9b}$$

Figure 9.3 shows the block diagrams of the two different MPC designs. The single layer approach needs the presented input mapping and liquid flow estimation in order to work, at least if the prediction model uses the original inputs. The temperature measurements are needed for the liquid flow estimation as indicated in its block diagram. This is because of the extra liquid caused by heat loss in Equation (9.9).

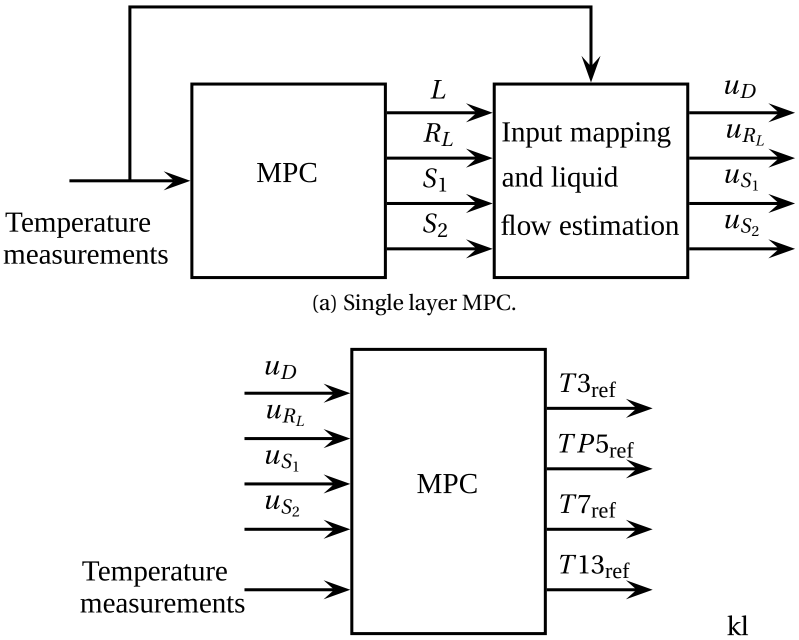
For the supervisory MPC approach there is no need for input mapping because the physical inputs only were added as measurements in order to constrain them as we saw in Chapter 7. Hence, for this MPC implementation only the real inputs need to be measured and constrained;

$$0 \le u_D \le 1, \tag{9.10a}$$

$$0 \le u_{R_L} \le 1, \tag{9.10b}$$

$$0 \le u_{S_1} \le 1, \tag{9.10c}$$

$$0 \le u_{S_2} \le 1. \tag{9.10d}$$



(a) Single layer MPC.

(b) Supervisory MPC. The decentralized controller is not drawn here since it is already implemented.

Figure 9.3: Block diagrams showing how the two different MPCs should be implemented at the laboratory.

9.3 Software implementation

The computer interface that is made as a part of earlier work by Jens Strandberg is a *LabVIEW interface*. This interface is further developed in this work in order to include an MPC. The LabVIEW software is connected to an I/O system that translate the variable values into signals that controls the valves, and the other way, converting sensor signals into temperature measurements. All temperature measurements are available in LabVIEW and the control inputs are manipulated from this program.

The MPCs that were implemented earlier in this thesis was made in Simulink. It is desirable to use the same environment when controlling the laboratory column with an MPC. Therefore, some kind of communication between Simulink and LabVIEW is needed. *OPC* (Object Linking and Embedding for Process Control) is here very suitable for this use. OPC is a standard that specifies the communication of real-time plant data between control devices from different manufacturers. LabVIEW includes as an OPC server, which is a library where all variables are stored. OPC Toolbox in MATLAB contains an OPC client, which connects to the OPC server such that it can access the variables.

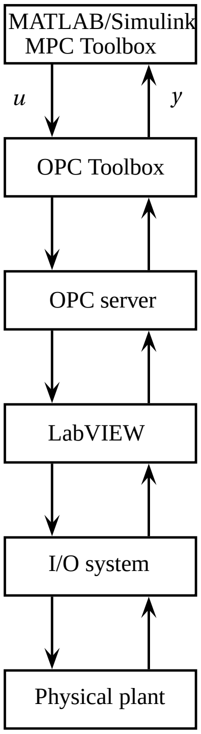


Figure 9.4: Data communication setup.

The data communication setup from the MPC in Simulink to the physical plant signals is shown in Figure 9.4. The OPC server can add a few seconds time delay to the feedback loop, but of no consequence since the process has time constants measured in minutes. Table 7.6 in Chapter 7 confirms this even though the laboratory has faster dynamics than the simulated column.

Simulink configuration

The Simulink part of the implementation is quite straight forward. The MPC setup in Simulink is shown as a block diagram in Figure 9.5. The OPC client is configured in a block called ”OPC Configuration” and by adding this to a Simulink diagram, the simulation will automatically be a real-time simulation. The shared variables can be read from the block ”OPC Read” and written to the block ”OPC Write”. The MPC itself is also just a simple block that uses the measurements and setpoints as

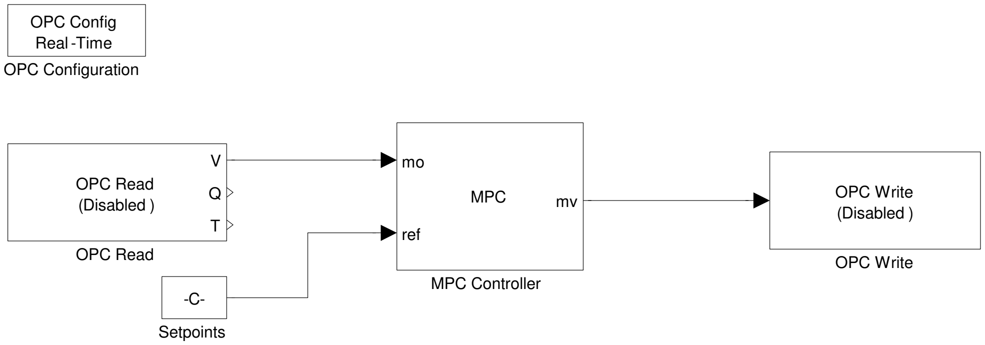


Figure 9.5: Simulink diagram showing the MPC setup.

inputs and the process inputs as outputs. These outputs are shown in Figure 9.3 for the two MPC approaches considered. If a single layer MPC is used together with a fitted prediction model, the MPC block must contain the input mapping and liquid flow estimation as well.

9.4 Experiments to be performed

This section proposes some experiments that need to be done to identify a prediction model and to tune the controllers. The section also suggests some experiments that can be performed in order to test the MPC performance.

Identification

Before the MPC can be tested a prediction model needs to be found. Two approaches are possible for identification of the prediction model used by the MPC; *black box* and *grey box identification,* which were introduced in Chapter 6. If black box identification is used to identify a model of the laboratory column it is basically identified with use of simple step responses. A small step is then perturbed at each input sequentially, and the responses of the outputs are recorded and fitted to a low-order linear response. If black box identification is used, the input mappings presented earlier can actually be avoided completely.

Another identification approach is to use the existing simulation model and find suitable values for different model parameters that can be adjusted such that simulation results are similar to experimental results. This is grey box identification. The model parameters that can be adjusted for the Kaibel model are the following:

- The efficiency parameter that was added to the model (see Chapter 5)
- Heat transfer coefficient (see Chapter 5)
- Number of stages in each column section

After suitable model parameters are found the model needs to be linearized in order to be used in a linear MPC.

Tuning

If the supervisory MPC is chosen for control of the laboratory column the decentralized controllers should be tuned properly as pointed out in the end of Chapter 6. The Simple / Skogestad internal model control (SIMC) tuning rules (Skogestad, 2003b) can be applied to find good tuning parameters for the controllers in a simple way. The SIMC tuning rules require a first order approximation of the process from the manipulated variable to the control variable. Since it is four manipulated variables controlling the laboratory column, four step responses around steady state operation is required.

Testing of MPC performance

The MPC should be compared with the existing decentralized controller to find out how much the operation performance can be increased.

In earlier simulations the different controllers have been tested with different steps in disturbances. As we remember from Chapter 3 the modeled disturbances are

$$d = \begin{bmatrix} V & R_V & F & z_D & z_{S_1} & z_{S_2} & q \end{bmatrix}^\top . \tag{9.11}$$

These inputs can be varied at the laboratory column. E.g. if it is desirable to do a step in the feed flow, the setpoint for the feed pump is changed. A step in the feed composition, e.g. $z_{S_1} + 20\%$, can easily be done by adding more ethanol into the feed tank. Another possibility to test performance is to change the setpoints given by Equation (9.2) while the column is in operation.

The simulations in Chapter 7 were compared by calculating the total impurity flow. This can also be done at the laboratory, but not continuously since *samples* must be taken from the product flows for later laboratory analysis in order to measure the purities exact.

9.5 Implementation summary

The table below summarizes the MPC implementation procedure for the laboratory column.

1	**Identify a prediction model**
	Two possible approaches: a) Step responses from the inputs in open-loop. b) Use existing simulation model and do a parameter adjustment (efficiency parameter, heat transfer coefficient and number of column stages) to fit experimental data.
2	**Linearize**
	Linearize the identified prediction model for use in the MPC.
3	**Determine prediction horizon**
	Perform some disturbance responses in closed-loop with decentralized controllers and set the prediction horizon at least to the same length as it takes for the temperatures to settle. The MPC sampling time can be chosen using the same ratio as in the simulation MPC; $T_s = \frac{T_{\text{settling time}}}{50}$. Thus, the MPC will then have a horizon length of 50 discrete time steps.
4	**Implement the prediction model in MATLAB**
	Two possible approaches also here: a) *Single layer MPC:* Use the linearized prediction model. Note that this approach requires input mapping if the prediction model uses the original inputs as we saw in Figure 9.3. b) *Supervisory MPC:* Use Equation (7.21a) and (7.23) to find the prediction model matrices. The decentralized control matrices G_0 and G_1 needs to be known. Note that the decentralized controllers needs to be well tuned when this MPC approach is used. Add u_D, u_{R_L}, u_{S_1} and u_{S_2} as measurements for the MPC and add constraints on these measurements as given in Equation (9.10).

Note that the linearized model to be used as a prediction model in the MPC must be discretized using the same sampling time as the MPC; T_s.

CHAPTER 10

DISCUSSION

This chapter provides a discussion of the results that are obtained in this work. Results from simulations that were done in the model extension part are discussed first, followed by discussions regarding model predictive control of the column model and the laboratory column.

10.1 Model extension

Chapter 5 presented the simulation results after the simulation model was extended to include vapor bypassing to describe insufficient mixing at the column stages. The product purities were plotted in Figure 5.2 for different bypass factors. This figure shows that all purities decrease except the bottom product purity, which increases some. This behavior corresponds with a theoretical view since when some vapor skips a stage in the column on its way upwards, the products above the bottom product gets less pure because the distillation process is less effective. This is why the vapor bypass factor also is referred to as an efficiency parameter.

Effect of vapor bypassing

A surprisingly result is that only a small amount of bypassing is needed to do a considerable change in the product purities in open-loop simulations. Only 1% ($\alpha = 0.01$) vapor bypassing resulted in a total impurity flow of 0.0945 mol/min when the response has settled. The impurity flow without bypassing is 0.0439 mol/min, hence, the impurities has increased with 115%.

To check whether this is an effect that is special only for a Kaibel column, the author performed some simple simulations on a binary distillation column ("Column A") from Skogestad & Postlethwaite (2005). The MATLAB code is given by Skogestad (1996). This 41-stage binary column was simulated with a vapor boil-up of 3.000 mol/min, a feed flow equal to 1.000 mol/min and product flows of 0.5000 mol/min each. The initial impurity flow without vapor bypassing was found to be 0.0136 mol/min and with 1% vapor bypassing the impurity flow became 0.0325 mol/min, giving an impurity flow increase of 139%. This example shows that vapor bypassing results in a significant increase in the impurity for an ordinary binary column also, not only for a Kaibel column.

Control compensation

In closed-loop the controllers counteract increased vapor bypass with increased reflux flow at the top of the column. Increased reflux makes the products above the bottom more pure as mentioned in the distillation introduction in Chapter 3. Increased reflux also leads to less product flow in the top since the condensed liquid from the condenser is separated into reflux and top product flow.

When the vapor bypassing becomes too high the controllers are not able to hold the temperature profile at its optimal position because of limited amount of reflux. Simulations that were performed for the sensitivity analysis in Chapter 7 this happened in closed-loop when the bypassing was above 10%.

10.2 MPC simulations

Three different MPCs were made for control of the Kaibel distillation column model. Their simulation results are discussed next.

The impurity flow plots (Figure 7.6) show that the MPCs perform better than the existing decentralized controller for the tested disturbances. This is because of the MPC's ability to look forward with its prediction and to handle process interactions in a more optimal manner. Especially two MPCs have been compared in this thesis; a single layer MPC operating directly on the physical inputs, and a supervisory MPC controlling the setpoints of a regulatory control layer.

Skogestad & Postlethwaite (2005) argues that a single layer optimal control approach (referred to as ”Integrated optimization and control”) is not normally used because of several reasons:

- *Cost of modeling*
- *Difficulty of controller design*
- *Maintenance and modification*
- *Robustness problems*

The different MPCs have different temperature responses (Figure 7.5) which are only comparable to some extent since they also depends on the chosen tuning parameters. The tuning parameters were found using a trial and error approach with an objective to obtain fast and at the same time smooth output responses. Generally it can be seen from the disturbance responses that the single layer MPC with disturbance estimation (first design) achieves its setpoints relatively fast with a smooth response. The second MPC design, singe layer MPC without a disturbance model made in MPC Toolbox, uses more time to achieve these setpoints.

Without any model errors, the single layer MPC and the supervisory MPC give quite similar performance. The nominal case, when no model errors are present, is shown in Table 7.6 (when the time delay equals zero) for the tested disturbances. The supervisory MPC has some better performance for the large disturbance in the vapor split. A drawback of the supervisory MPC is that it has quite oscillatory responses after the disturbances (Figure 7.5), but it should be possible to reduce these oscillations by spending more time on tuning.

For the setpoint change simulation that was done in Chapter 7 (Figure 7.7), all controllers, including the decentralized controller, achieved the setpoint in about the same time. The difference between the decentralized control approach and the MPCs is that the MPCs avoid much of the responses caused by process interaction, i.e. responses on other temperatures. The single layer MPCs reduce this effect most.

As discussed in the section ”Simulation parameters” in Chapter 7, the first MPC design needed considerable work on the tuning in order to achieve fast and smooth responses. This highlights the difficulty of controller design as listed above, since the supervisory MPC approach achieved quite nice responses without much tuning.

10.3 Simulations with model errors

Results from the section ”Sensitivity analysis” in Chapter 7 certainly shows that a supervisory MPC is preferable compared to a single layer version in terms of robustness. This corresponds with the list given above. The presented results do of course also depend on the chosen tuning parameters, but this should not change much of the discussion.

Input uncertainties

As we saw in the figures 7.10 -7.13 in Chapter 7, the supervisory MPC results in less impurity flows when input uncertainties are present compared the single layer MPC. The single layer MPC gives relatively large impurity flow when the reflux flow is too low (Figure 7.10 and 7.11). For a decentralized controller, it is gain errors in the liquid split that causes high impurity. Especially for too low values of R_L, i.e. too much liquid is leaded down the main column. As we can see from Figure 7.10 and 7.12, the supervisory MPC gives less impurity flows for input uncertainties.

Since the supervisory MPC has an inner PI-control loop, this results in better robust performance. This inner loop removes a lot of the input gain error and makes the MPC control problem easier. Skogestad & Postlethwaite (2005) also

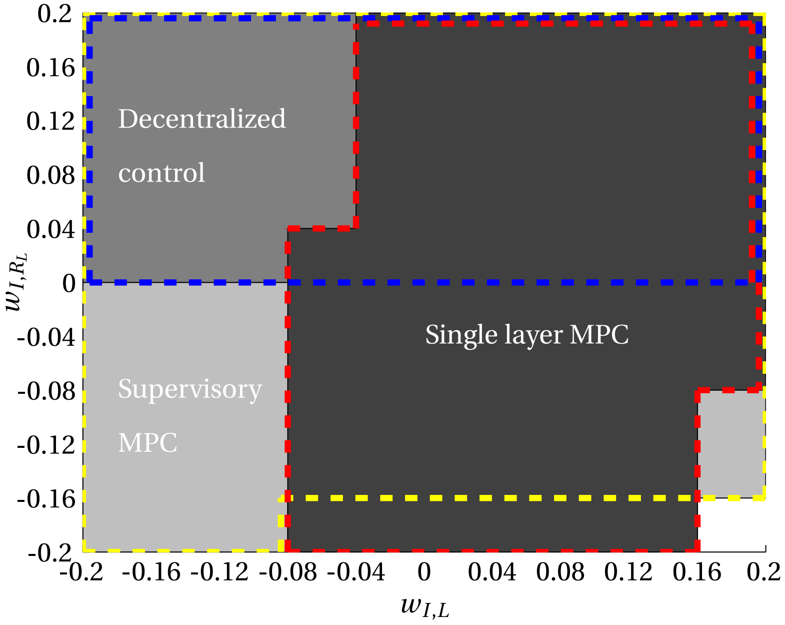


Figure 10.1: Feasible areas when the maximum allowable impurity flow is set to 0.1 mol/min.

states that an inner decentralized control loop linearizes the process seen from the setpoints of the inner loop controllers, which gives a simpler optimization problem for the MPC.

Under the sensitivity analysis section it was also briefly discussed that uncertainties in the side streams does not affect the impurity flow as much as uncertainties in the liquid split and the reflux flow. This can be seen by comparing Table 7.5 with Figure 7.10. This is an important result that emphasizes the importance of keeping uncertainties in the liquid split and the reflux flow inputs low for a Kaibel column.

The 3D-plots from Chapter 7 were projected down to a two-dimensional plot in Figure 10.1. The figure shows which area the impurity flow is less or equal to 0.1 mol/min for the different control structures. An impurity flow of 0.1 mol/min corresponds to a minimum purity of 90% for the products if the output flows are around their nominal values. As we can see in the figure, the largest covered area is represented by the supervisory MPC, only giving poor performance for too low values of R_L. The plot has poor resolution since the decentralized controller seemingly can not handle too low values of R_L at all, but the low resolution is caused by the number of simulations that were performed to create the plot. Hence, the performance limit is somewhere between 0 and -0.04 for the decentralized controller.

Effect of time delay

The supervisory MPC also gave better simulation results when time delays were added to the measurements (see Table 7.6) compared to the single layer MPC. If the same criteria as above is used (maximum impurity flow of 0.1 mol/min), the single layer MPC is able to handle time delays up to around 5 min for the tested disturbances. The supervisory MPC can handle around twice as much time delay. It should be noted that the regulatory control layer also suffered from the same time delays in the simulations, not only the MPC.

It was the decentralized controller that gave best performance when the process where suffering from time delay. This is because the prediction models used by the MPCs not were modeled with time delay. In this case, it can clearly be seen that a poor prediction model leads to poor performance for an MPC, and a simple PI-control structure gives better performance. However, it is unlikely that a temperature controlled distillation would suffer of time delay as much as 15 min. If the column was controlled by measuring the purities, it is a reasonable time delay because of slow measuring instruments, but it is not comparable since the column is temperature controlled in this case.

Effect of vapor bypassing

The decentralized controller and the compared MPCs were tested for different values of the vapor bypassing factor. The results were given in Table 7.7. The three controllers give quite similar performance as we can see in the table. This is probably because that the vapor bypassing effect is counteracted by simply increasing the reflux, which all three controllers manage easily.

10.4 Optimization objective

Normally, for a two layer control approach different outputs are controlled in the different layers. For the supervisory MPC made in this thesis both layer objectives is to control the four selected temperatures to their setpoints. Therefore, a composition estimator that was presented in Chapter 8 can be preferable for more optimal control of the column. It will be more optimal because the pre-computed temperature setpoints that are used now, changes when disturbances occurs. By knowing the purities the MPC will be able to optimize over the real objective that is used in this thesis; the total impurity flow.

10.5 Implementation at the laboratory column

Issues regarding implementation of an MPC were presented in Chapter 9. The MPC needs a prediction model that must be identified. Two possible identification methods are black box or grey box identification. As mentioned in Chapter 6, simple step responses are the most common industrial practice, which is black box identification. This is clearly the simplest approach to identify the laboratory column.

It is interesting to fit the existing model to the laboratory column in order to understand more of the column dynamics. The motive for adding the efficiency parameter was to describe a physical effect that happens in real columns, including the laboratory column. By fitting the simulation model to the laboratory column the equivalent number of stages, heat loss and the efficiency parameter can be estimated and from there it can be concluded whether e.g. the column suffers of to high heat loss. Parameter fitting is a much harder identification problem than a simple step response approach and require a lot of experimental data. Kvernland (2008) also concluded this.

Hence, identification by use of step responses is the fastest and most simple way to obtain a prediction model, but it does not give any new information about the column dynamics. Since the laboratory column is built for research purposes it is probably most interesting to do a parameter adjustment to learn more about the Kaibel column.

A single layer MPC versus a supervisory MPC was also compared in Chapter 9. Because of the input mappings caused by the valve constructions, the supervisory MPC approach is preferable since it does not require this input mapping at all. However, this approach needs a well tuned regulatory control layer, which can be time-consuming.

CHAPTER 11

CONCLUSION

The main focus of this work has been to do a model extension to the existing simulation model, and describe and implement model predictive control for the Kaibel column. In addition, do some sensitivity analysis of model errors and describe an approach for MPC implementation at the laboratory column.

11.1 Concluding remarks

The efficiency parameter that was added to the Kaibel model describes insufficient vapor mixing that happens between column stages in a distillation column. The parameter decides how much vapor that is bypassed to two column stages above. The total impurity flow increases when this bypassing increases. In open-loop, simulations have shown huge effect on the impurity flows only for small amounts of this bypassing. Simulations of a binary column gave similar results; hence the large increase in the total impurity flow is not particular for a Kaibel column. In closed-loop, increased reflux compensates against the effect of increased vapor bypassing. The limited amount of reflux gives an infeasible control problem when the bypassing becomes too large.

The developed MPCs performed better than the existing decentralized controller in simulations on the Kaibel column model. Typically improvements that can be seen from the temperature responses are less interaction between the controlled temperatures, and in some cases faster settling time.

The author has chosen to compare a single layer based MPC and a supervisory MPC that controls the setpoints of a regulatory control layer in the thesis. Without model errors, these two MPC approaches have given quite similar performance. When model errors are present, there is a major difference between these two approaches. Simulations have shown better robustness properties of the supervisory MPC. The MPCs have been tested with different gain errors on the inputs, together with different time delays from the measurements and in addition, for different values of the efficiency parameter that was added to the model. The supervisory MPC have shown considerable better performance when gain error and time delay is present compared to the single layer approach.

Simulations with gain errors on the inputs showed more loss (increased impurity flows) for errors in the reflux and the liquid split than the side streams, independent of the controller. This indicates that it is important to keep the uncertainty in the reflux and liquid split input low when operating a Kaibel column.

Other MPC alternatives have been briefly discussed. An inferential control structure can be appropriate for control of a Kaibel column, that is estimation of the purities and let the MPC optimize over these purity estimates. Explicit MPC (EMPC), nonlinear MPC (NMPC) and a quadratic dynamic matrix controller (QDMC) are also discussed as possible approaches for control of the column. A QDMC approach is most appropriate for temperature control only. An NMPC can be suitable for a highly nonlinear process as a distillation column, but it results in a complex implementation, and a supervisory MPC is considered as preferable. An EMPC does not support the ability for online tuning and in addition, its main motivating factor which is the computation speed is not necessary for a slow distillation process.

When it comes to implementation of an MPC at the existing laboratory facilities at Department of Chemical Engineering, a supervisory MPC is suggested for use. Its robustness properties, together with an easier implementation make this MPC approach as the recommended one. A parameter adjustment of the simulation model is recommended as the way to obtain a prediction model for the MPC. Using a fitted simulation model and a single layer MPC require a quite complex input mapping which is why the supervisory is easier for implementation.

The main conclusions of the thesis can then be summarized:

- Including physical effects to the simulation model like the efficiency parameter helps for further understanding of the column dynamics. This physical effect leads to increased impurity flows, but can be counteracted by increased reflux flow, as long as the reflux is below its maximum saturation limit.

- An MPC performs better than a decentralized controller such that it is easier to achieve the lower energy consumption that characterizes a Kaibel column. Especially a supervisory MPC gives much better performance than the existing decentralized control structure, in terms of dynamic behavior and robustness.

- A supervisory MPC is found to be the most appropriate MPC approach for control of the Kaibel column, both in simulations and for implementation at the laboratory column.

11.2 Contributions provided by this thesis

The existing simulation model does not include enough physical effects such that experimental results can be explained properly from computer simulations. Adding the efficiency parameter to the model would hopefully make it easier to understand the dynamics of a Kaibel distillation column especially in further work.

To the author's knowledge model predictive control has never been used to control a Kaibel distillation column and this can be an important contribution for making the industry prefer a Kaibel distillation column compared to a conventional column setup in the future.

11.3 Suggestions for further work

The lists below presents further work to be done regarding MPC on the Kaibel column.

- Use experimental data to identify the adjustable parameters in the simulation model in order to fit experimental and simulation results.
- Perform an experimental test of the MPC and compare with decentralized control.
- Develop a composition estimator such that an MPC that optimizes over the total impurity flow can be made.
- Develop an NMPC too see if it improves the performance results given by the supervisory MPC made in this work.

Bibliography

Allgöwer, F., Findeisen, R., & Nagy, Z. (2004). Nonlinear model predictive control: From theory to application. *Journal of the Chinese Institute of Chemical Engineers*, 35(3), 299–315.

Aske, E. M. B. (2009). *Design of plantwide control systems with focus on maximizing throughput*. PhD thesis, Department of Chemical Engineering, Norwegian University of Science and Technology.

Aylward, G. & Findlay, T. (2008). *SI Chemical Data*. John Wiley & Sons Australia, Ltd.

Balchen, J. G., Andresen, T., & Foss, B. A. (2003). *Reguleringsteknikk*. Deparment of Engineering Cybernetics, Norwegian University of Science and Technology.

Bemporad, A. & Morari, M. (1999). Robust model predictive control: A survey. *Lecture notes in control and information sciences*, (pp. 207–226).

Bemporad, A., Morari, M., & Ricker, N. L. (2009). *Model Predictive Control Toolbox User's Guide*. The Mathworks, Inc.

Borrelli, F. & Morari, M. (2007). Offset free model predictive control. *Proceedings of the 46th IEEE Conference on Decision and Control*.

Camacho, E. & Bordons, C. (2004). *Model Predictive Control*. Springer.

Chang, R. (2006). *General chemistry*. McGraw-Hill.

Chen, C. (1999). *Linear System Theory and Design*. Oxford University Press.

Cutler, C., Morshedi, A., & Haydel, J. (1983). An industrial perspective on advanced control. In *AIChE Annual Meeting*.

Cutler, C. & Ramaker, B. (1979). Dynamic matrix control–a computer algorithm.

Fletcher, R. (1987). Practical methods of optimization.

Foss, B. A. (2004). Linear quadratic control. Lecture notes from the course Optimization and Control, Department of Engineering Cybernetics, NTNU.

Garcia, C. & Morshedi, A. (1986). Quadratic programming solution of dynamic matrix control (qdmc). *Chemical Engineering Communications*, 46(1), 73–87.

Gmehling, J. & Onken, U. (1977). *Vapor-Liquid Equilibrium Data Collection*. Dechema.

Halvorsen, I. & Skogestad, S. (2006). Minimum energy for the four-product kaibel-column. *AIChE Annual meeting*.

Halvorsen, I. J. (2001). *Minimum Energy Requirements in Complex Distillation Arrangements*. PhD thesis, Department of Chemical Engineering, Norwegian University of Science and Technology.

Honeywell (2008). *UniSim Design Dynamic Modeling Reference Guide*. Honeywell.

Hovd, M. (2008). Lecture notes. Lecture notes from the course Advanced Control of Industrial Processes, Department of Engineering Cybernetics, NTNU.

Imsland, L. (2007). Introduction to model predictive control. Lecture notes from the course Optimization and Control, Department of Engineering Cybernetics, NTNU.

Kaibel, G. (1987). Distillation columns with vertical partitions. *Chemical engineering & technology*, 10(2), 92–98.

Kaibel, G., Miller, C., Stroezel, M., von Watzdorf, R., & Jansen, H. (2004). Industrial application of dividing wall distillation columns and thermally coupled distillation columns. *Chemie Ingenieur Technik*, 76(3), 258–258.

Kalman, R. (1960a). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.

Kalman, R. E. (1960b). Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex., II. Ser.*

Khalil, H. K. (2000). *Nonlinear Systems*. Pearson Education.

Kuhn, A. W. & Tucker, H. W. (1951). Nonlinear Programming. *University of California Press*, (pp. 481–492).

Kvernland, M. (2008). *Control and modelling of Kaibel distillation column*. Project work, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

Ljung, L. (1999). *System Identification, Theory for the User*. Prentice Hall PTR.

Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall.

Markus, L. & Lee, E. (1967). Foundations of optimal control theory. *J. Wiley, New York*.

Mejdell, T. & Skogestad, S. (1991). Estimation of distillation compositions from multiple temperature measurements using partial-least-squares regression. *Industrial & Engineering Chemistry Research*, 30(12), 2543–2555.

Morari, M. & Lee, J. H. (1999). Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23(4-5), 667–682.

Muske, K. & Rawlings, J. (1993). Model predictive control with linear models. *AIChE Journal*, 39(2), 262–287.

Nocedal, J. & Wright, S. J. (2006). *Numerical Optimization*. Springer.

Ognisty, T. (1995). Analyze distillation columns with thermodynamics. *Chemical Engineering Progress*, 91(2), 40–46.

Petlyuk, F., Platonov, V., & Slavinskii, D. (1965). Thermodynamically optimal method for separating multicomponent mixtures. *Int. Chem. Eng*, 5(3), 555–561.

Qin, S. & Badgwell, T. (1997). An overview of industrial model predictive control technology. 93, 232–256.

Skogestad, S. (1996). Distillation research and models. URL: http://www.nt.ntnu.no/users/skoge/distillation/.

Skogestad, S. (1997). Dynamics and control of distillation columns: A tutorial introduction. *Chemical Engineering Research and Design/Official Journal of the European Federation of Chemical Engineering: Part A*, 75(A6), 539–562.

Skogestad, S. (2003a). *Prosessteknikk, Masse- og energibalanser*. Tapir Akademisk Forlag.

Skogestad, S. (2003b). Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*.

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

Skogestad, S. (2007). The dos and don’ts of distillation column control. *Trans IChemE*.

Skogestad, S. & Postlethwaite, I. (2005). *Multivariable Feedback Control*. John Wiley & Sons, Ltd.

Strandberg, J. (2009). Draft for phd thesis. URL: http://www.nt.ntnu.no/users/skoge/publications/thesis/2009_strandberg/.

Strandberg, J. & Skogestad, S. (2006). Stabilizing operation of a 4-product integrated kaibel column. *Symposium Series no. 152*.

Strang, G. (2006). *Linear Algebra and its application*. Thomson.

Tøndel, P., Johansen, T., & Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3), 489–497.

Ž. Olujić, Jödecke, M., Shilkin, A., Schuch, G., & Kaibel, B. (2009). Equipment improvement trends in distillation. *Chemical Engineering and Processing*.

Wilhelmshavener Raffinerie Gesellschaft (2004). How a refinery works. URL: http://www.wrg-whv.de/en_new/index_en.htm.

Wright, R. O. (1949). Fractionation apparatus. US Patent 2,471,134.

Ying, C. & Joseph, B. (1999). Performance and stability analysis of LP-MPC and QP-MPC cascade control systems. *AIChE Journal*, 45(7).

Zhou, K., Doyle, J., & Glover, K. (1996). *Robust and optimal control*. Prentice Hall Upper Saddle River, NJ.

Appendix A

Derivation of PI-control matrices

The discrete decentralized PI controller was given in Chapter 7 as

$$u_k = u_{k-1} + G_0(y_{s,k} - y_k) + G_1(y_{s,k-1} - y_{k-1}), \tag{A.1}$$

where the matrices G_0 and G_1 contain the controller parameters. This appendix shows how these matrices are derived.

A PI controller represented in the time-domain is written as

$$u(t) = K_c\left(e(t) + \frac{1}{\tau_I}\int_0^t e(\tau)\mathrm{d}\tau\right), \tag{A.2}$$

where the tuning parameters K_c and τ_I is the proportional gain and the integral time respectively. The integral term itself is then written as

$$\frac{1}{\tau_I}\int_0^t e(\tau)\mathrm{d}\tau = \frac{u(t)}{K_c} - e(t). \tag{A.3}$$

The integral term can be divided into two integrals, as follows

$$\int_0^t e(\tau)\mathrm{d}\tau = \int_0^{t-T_s} e(\tau)\mathrm{d}\tau + \int_{t-T_s}^t e(\tau)\mathrm{d}\tau, \tag{A.4}$$

where T_s is the sampling time of the controller. Using the two last equations gives the following relationship,

$$\frac{u(t-T_s)}{K_c} - e(t-T_s) = \int_0^{t-T_s} e(\tau)\mathrm{d}\tau. \tag{A.5}$$

The last integral term from Equation (A.4) can be approximated by use of the *Trapezoidal rule*;

$$\int_{t-T_s}^t e(\tau)\mathrm{d}\tau \approx \frac{1}{\tau_I}\left(\frac{T_s}{2}\left(e(t) + e(t-T_s)\right)\right). \tag{A.6}$$

By replacing t with kT from the time-domain representation and inserting the integral terms gives the discrete PI controller;

$$\begin{aligned} u_k &= K_c \left(e_k + \left(\frac{u_{k-1}}{K_c} - e_{k-1} \right) + \frac{1}{\tau_I} \frac{T_s}{2} \left(e_k + e_{k-1} \right) \right) \\ &= u_{k-1} + K_c \left(1 + \frac{T_s}{2\tau_I} \right) e_k - K_c \left(1 - \frac{T_s}{2\tau_I} \right) e_{k-1}. \end{aligned} \tag{A.7}$$

e_k is the setpoint deviation $y_{s,k} - y_k$. Hence, the matrices G_0 and G_1 in Equation (A.1) for a multiple input multiple output PI controller would be

$$G_0 = \begin{bmatrix} K_{c,1} \left(1 + \frac{T_s}{2\tau_{I,1}} \right) & & \\ & \ddots & \\ & & K_{c,n_u} \left(1 + \frac{T_s}{2\tau_{I,n_u}} \right) \end{bmatrix}, \tag{A.8a}$$

$$G_1 = \begin{bmatrix} -K_{c,1} \left(1 - \frac{T_s}{2\tau_{I,1}} \right) & & \\ & \ddots & \\ & & -K_{c,n_u} \left(1 - \frac{T_s}{2\tau_{I,n_u}} \right) \end{bmatrix}. \tag{A.8b}$$

Appendix B

Gain error simulation table

Table B.1 (next page) shows a list of the input uncertainty simulations that were performed in Chapter 7. The table shows the simulations when the gains for the inputs L, R_L, S_1 and S_2 were changed.

Simulation number	$w_{I,L}$	w_{I,R_L}	w_{I,S_1}	w_{I,S_2}	Simulation number	$w_{I,L}$	w_{I,R_L}	w_{I,S_1}	w_{I,S_2}
1	−0.1	−0.1	−0.1	−0.1	41	0.0	0.0	0.0	0.0
2	−0.1	−0.1	−0.1	0.0	42	0.0	0.0	0.0	0.1
3	−0.1	−0.1	−0.1	0.1	43	0.0	0.0	0.1	−0.1
4	−0.1	−0.1	0.0	−0.1	44	0.0	0.0	0.1	0.0
5	−0.1	−0.1	0.0	0.0	45	0.0	0.0	0.1	0.1
6	−0.1	−0.1	0.0	0.1	46	0.0	0.1	−0.1	−0.1
7	−0.1	−0.1	0.1	−0.1	47	0.0	0.1	−0.1	0.0
8	−0.1	−0.1	0.1	0.0	48	0.0	0.1	−0.1	0.1
9	−0.1	−0.1	0.1	0.1	49	0.0	0.1	0.0	−0.1
10	−0.1	0.0	−0.1	−0.1	50	0.0	0.1	0.0	0.0
11	−0.1	0.0	−0.1	0.0	51	0.0	0.1	0.0	0.1
12	−0.1	0.0	−0.1	0.1	52	0.0	0.1	0.1	−0.1
13	−0.1	0.0	0.0	−0.1	53	0.0	0.1	0.1	0.0
14	−0.1	0.0	0.0	0.0	54	0.0	0.1	0.1	0.1
15	−0.1	0.0	0.0	0.1	55	0.1	−0.1	−0.1	−0.1
16	−0.1	0.0	0.1	−0.1	56	0.1	−0.1	−0.1	0.0
17	−0.1	0.0	0.1	0.0	57	0.1	−0.1	−0.1	0.1
18	−0.1	0.0	0.1	0.1	58	0.1	−0.1	0.0	−0.1
19	−0.1	0.1	−0.1	−0.1	59	0.1	−0.1	0.0	0.0
20	−0.1	0.1	−0.1	0.0	60	0.1	−0.1	0.0	0.1
21	−0.1	0.1	−0.1	0.1	61	0.1	−0.1	0.1	−0.1
22	−0.1	0.1	0.0	−0.1	62	0.1	−0.1	0.1	0.0
23	−0.1	0.1	0.0	0.0	63	0.1	−0.1	0.1	0.1
24	−0.1	0.1	0.0	0.1	64	0.1	0.0	−0.1	−0.1
25	−0.1	0.1	0.1	−0.1	65	0.1	0.0	−0.1	0.0
26	−0.1	0.1	0.1	0.0	66	0.1	0.0	−0.1	0.1
27	−0.1	0.1	0.1	0.1	67	0.1	0.0	0.0	−0.1
28	0.0	−0.1	−0.1	−0.1	68	0.1	0.0	0.0	0.0
29	0.0	−0.1	−0.1	0.0	69	0.1	0.0	0.0	0.1
30	0.0	−0.1	−0.1	0.1	70	0.1	0.0	0.1	−0.1
31	0.0	−0.1	0.0	−0.1	71	0.1	0.0	0.1	0.0
32	0.0	−0.1	0.0	0.0	72	0.1	0.0	0.1	0.1
33	0.0	−0.1	0.0	0.1	73	0.1	0.1	−0.1	−0.1
34	0.0	−0.1	0.1	−0.1	74	0.1	0.1	−0.1	0.0
35	0.0	−0.1	0.1	0.0	75	0.1	0.1	−0.1	0.1
36	0.0	−0.1	0.1	0.1	76	0.1	0.1	0.0	−0.1
37	0.0	0.0	−0.1	−0.1	77	0.1	0.1	0.0	0.0
38	0.0	0.0	−0.1	0.0	78	0.1	0.1	0.0	0.1
39	0.0	0.0	−0.1	0.1	79	0.1	0.1	0.1	−0.1
40	0.0	0.0	0.0	−0.1	80	0.1	0.1	0.1	0.0
					81	0.1	0.1	0.1	0.1

Table B.1: Input uncertainty simulation experiments.