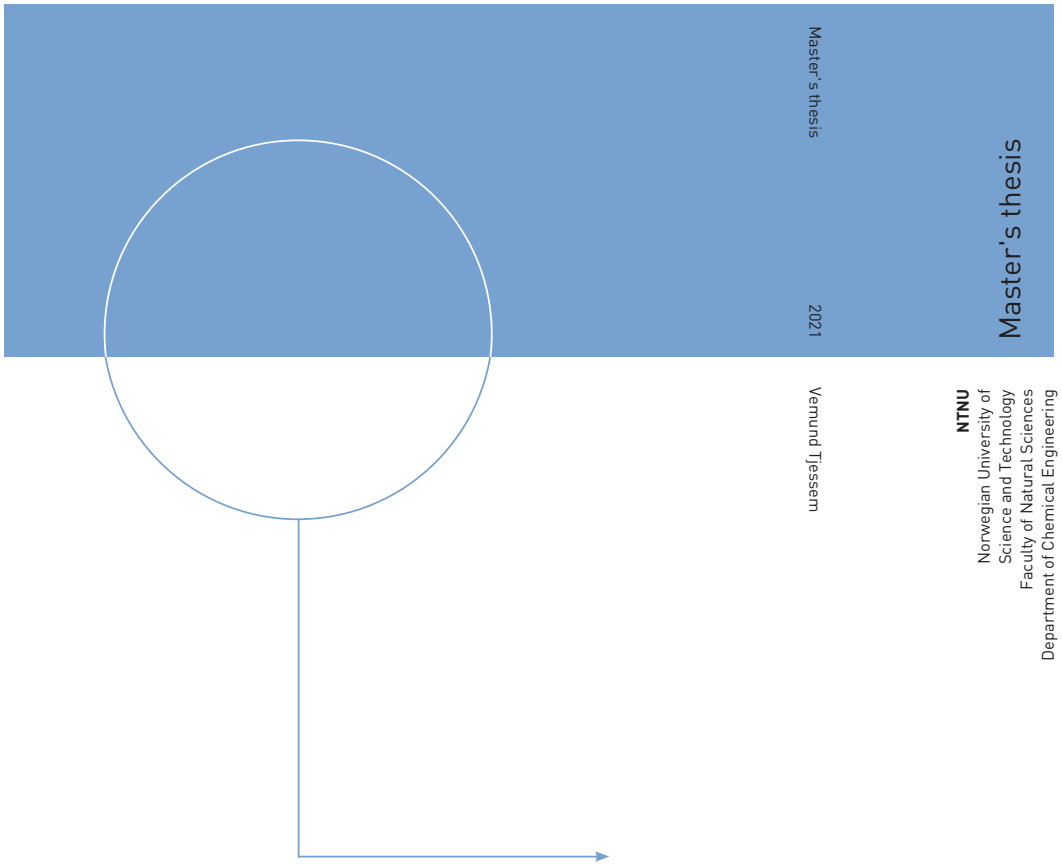



Vemund Tjessem

# Nonlinear model predictive control of a multi-absorber $CO_2$ capture plant

June 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

# Nonlinear model predictive control of a multi-absorber $CO_2$ capture plant

Vemund Tjessem

Chemical Engineering and Biotechnology
Submission date: June 2021
Supervisor: Magne Hillestad, IKP
Co-supervisor: Fredrik Gjertsen, Cybernetica AS
Svein Olav Hauger, Cybernetica AS
Peter Singstad, Cybernetica AS

Norwegian University of Science and Technology
Department of Chemical Engineering

# Abstract

In post-combustion $CO_2$ capture, $CO_2$ is absorbed in an amine solvent. The solvent is regenerated in the desorber column by heat supplied in the reboiler. A large cost when operating a post-combustion $CO_2$ capture plant is the reboiler duty. In this master's thesis, a control configuration based on nonlinear model predictive control (NMPC) was developed for a multi-absorber $CO_2$ capture plant, the objective function was formulated in a way that should control the plant towards minimum reboiler duty.

A $CO_2$ capture plant model received from Cybernetica AS was expanded to include multiple absorber columns. The model was split into two parts, one with the absorber columns and one with the desorber column and the other process units. Several simple absorber models were developed and paired with the desorber half of the model in two-NMPC control configurations. The configurations were tested and developed until a final control structure was produced.

The developed two-NMPC configurations were tested and compared with a benchmark control configuration. The benchmark control configuration consisted of one NMPC using the original model. The original model was used as the plant replacement model in Cybernetica RealSim in all simulations, so the benchmark controller had no model mismatch with the plant, unlike the two-NMPC configurations. The two-NMPC configurations exhibited varying performance, but the final configuration offered comparable performance to the benchmark controller with a 73.6 % reduction in average computational time per sample. There was also developed another configuration with individual NMPCs for each absorber column, which reduced the average computational time by 83.1 %.

The results showed that the developed control configurations could deliver acceptable performance while reducing the computational time. The time reduction becomes increasingly significant the more absorber columns are added to the plant.

# Sammendrag

I et etterforbrenning karbonfangstanlegg blir det brukt en aminløsning for å absorbere $CO_2$. Aminløsningen blir regenerert i desorberkolonnen gjennom varme tilført i omkokeren. En stor driftskostnad ved et karbonfangstanlegg er energitilførselen i omkokeren. Det ble i denne masteroppgaven utviklet en kontrollkonfigurasjon for et karbonfangstanlegg med multiple absorbere, basert på ulineær modellbasert prediktiv regulering (NMPC). Målfunksjonen ble formulert på et vis som styrer anlegget mot minimalt energiforbruk samtidig som settpunkt på fansgtgrad opprettholdes.

En modell av et karbonfangstanlegg mottatt fra Cybernetica AS ble utvidet til å inkludere multiple absorberkolonner. Modellen ble delt i to deler, en med absorberkolonnene og en med desorberkolonnen og de andre enhetsoperasjonene. Flere enkle absorbermodeller ble utviklet og kombinert med desorberhalvdelen av modellen i to-NMPC kontrollkonfigurasjoner. Konfigurasjonene ble testet og utviklet til en endelig kontrollstruktur ble ferdigstilt.

De utviklede to-NMPC konfigurasjonene ble testet og sammenlignet med en referansekonfigurasjon. Referansekonfigurasjonen bestod av én NMPC som bruker den originale modellen. Den originale modellen ble benyttet som anleggerstatningsmodell i Cybernetica RealSim i alle simuleringer. I motsetning til to-NMPC konfigurasjonene hadde ikke referansekonfigurasjonen modellavvik fra anlegget. To-NMPC konfigurasjonene hadde varierende ytelse, men den endelige konfigurasjonen hadde sammenlignbar ytelse med referansekonfigurasjonen med en 73.6 % reduksjon i gjennomsnittlig beregningstid per sampel. Det ble også utviklet en annen konfigurasjon med en individuell NMPC for hver absorberkolonne som reduserte den gjennomsnittlige beregningstiden med 83.1 %.

Resultatene viste at de utviklede kontrollkonfigurasjonene kunne levere akseptabel ytelse med redusert beregningstid. Reduksjonen i beregningstid blir enda viktigere jo flere absorberkolonner som blir lagt til i anlegget.

# Preface

This master's thesis represents the finalisation of the degree of Master of Science in Chemical Engineering at the Norwegian University of Science and Technology. The work in this master's thesis is a continuation of the work done in a specialisation project during the fall of 2020. The specialisation project was a continuation of work done in a summer internship at Cybernetica AS during the summer of 2020.

I would like to thank my supervisor from the Department of Chemical Engineering at NTNU, Magne Hillestad, for his advice during the specialisation project and master's thesis work.

I would also like to extend my gratitude towards the employees of Cybernetica AS for being friendly and including me in office activities in spite of the current pandemic. Special thanks go towards my co-supervisors from Cybernetica, Fredrik Gjertsen, Svein Olav Hauger and Peter Singstad for their continuous encouragement and advice.

## Declaration of Compliance

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

June 25, 2021
Trondheim, Norway

Vemund Tjessem

# Contents

# List of Figures

# List of Tables

# List of Code boxes

# Nomenclature

**Acronyms**

| | |
|---|---|
| AMP | 2-Amino-2-methylpropan-1-ol |
| BVP | Boundary value problem |
| CCUS | Carbon Capture Utilisation and Storage |
| CESAR | $CO_2$ Enhanced Separation and Recovery |
| CR | Capture ratio |
| CV | Controlled variable |
| DOCPCC | Demonstration of Optimal Control of Post-Combustion Capture Processes |
| DOF | Degree of freedom |
| DV | Disturbance variable |
| EKF | Extended Kalman filter |
| FOPTD | First-order-plus-time-delay |
| IVP | Initial value problem |
| KF | Kalman filter |
| MEA | Monoethanolamine |
| MHE | Moving horizon estimator |
| MI | Mission Innovation |
| MPC | Model Predictive Control |
| MV | Manipulated variable |
| NLP | Nonlinear programming |
| NMPC | Nonlinear Model Predictive Control |

| | |
|---|---|
| ODE | Ordinary differential equation |
| OPC | Open Platform Communications |
| P&ID | Piping and instrumentation diagram |
| PID | Proportional Integral Derivative |
| PZ | Piperazine |
| QP | Quadratic programming |
| RHC | Receding horizon control |
| SEPTIC | Statoil Estimation and Prediction Tool for Identification and Control |
| SIMC | Skogestad Internal Model Control |
| SISO | Single input single output system |
| SQP | Sequential quadratic programming |
| SRD | Specific reboiler duty |
| WMO | World Meteorological Organization |

Chapter 1

# Introduction

$CO_2$ is perceived as the most important greenhouse gas, as it is the greenhouse gas contributing the most to human-induced climate change. A greenhouse gas is a gas that absorbs infrared radiation emitted from the Earth's surface and reradiates it back. This contributes to the greenhouse effect and leads to a warming of the Earth's surface[1]. Other greenhouse gases include water vapour, $N_2O$, methane and F-gases.

The $CO_2$ pollution in Norway comes mainly from the combustion of fossil fuels. The primary sources are transportation, oil and gas production and industry. The trend after 1990 was a significant increase in $CO_2$ emissions from transportation and oil and gas production until about 2008. Since then, it has been more stable, and in recent years there has been a decrease in emissions from transportation. From 1990 to 2019, the $CO_2$ emissions have increased by 19 %. The increase in oil and gas production is an important reason for this[2].

The increase of $CO_2$ in the atmosphere is a global problem. In 2010 $CO_2$ stood for 76 % of greenhouse gas emissions from human activities on a global scale[3]. A press release from World Meteorological Organization (WMO) states that $CO_2$ levels continued at record levels in 2020 after having breached 410 ppm in 2019[4]. $CO_2$ emissions are a global problem that humanity has to solve. One method of decreasing $CO_2$ emissions is through Carbon Capture Utilisation and Storage (CCUS). There are several methods for $CO_2$ capture, including both pre and post-combustion $CO_2$ capture[5].

This master's thesis was written in collaboration with Cybernetica AS. Cybernetica AS is a Trondheim based company specialising in model-based control systems. The topic of this master's thesis relates to the control of post-combustion amine-based capture plants, which Cybernetica AS has been involved in for many years now[1]. Enaasen Flø laid the foundations in her PhD thesis, where she developed a dynamic process model of a $CO_2$ capture plant[6]. This model was later implemented in C and further developed by Cybernetica AS during various projects, such as DOCPCC 1 and 2[7]. These project resulted in improvements to

[1] http://cybernetica.no/cybe_case/docpcc/

the dynamic process model. For the interested reader, there are published papers detailing the results of the tests[8;9].

Other attempts have also been made to control $CO_2$ capture plants. Panahi and Skogestad[10;11] proposed control structures for different operational regions using self-optimising control. There are also studies which have proposed Model Predictive Control (MPC) solutions. He et al.[12] implemented both MPC and nonlinear MPC to a $CO_2$ capture process and found economic improvements compared to a traditional configuration with PID controllers. Sultan et al.[13] developed a $2 \times 2$ MPC control strategy of a $CO_2$ capture plant using a 2nd order continuous-time state-space model.

The objective of this master's thesis is to develop control configurations for a multi-absorber $CO_2$ capture plant based on nonlinear model predictive control (NMPC). The control configurations must be able to satisfy capture ratio setpoints while steering the plant towards minimum reboiler duty.

## 1.1 REALISE CCUS

The work in this master's thesis is related to the project REALISE CCUS. REALISE CCUS is a project which brings together partners from science and industry in Europe, China and South Korea to demonstrate the full $CO_2$ chain for industrial clusters centred on refineries. The project is a three-year project which started in May 2020 and has received funding from the European Union's Horizon 2020 research and innovation programme. The project consists of seven work packages. These are Optimisation, Capture, Industrial clusters, Society, Dissemination, Collaboration with Mission Innovation (MI) countries and Project management[14]. Cybernetica's part of the project relates to controlling the plant. The plant, in this thesis, is a theoretical plant with multiple absorber columns and one common desorber column.

## 1.2 Summary of the specialisation project

This master's thesis is a continuation of the work done in a specialisation project during the fall of 2020. The purpose of the specialisation project was to compare different control structures for controlling a $CO_2$ capture plant with an NMPC. First, the original mechanistic model was split into two parts. One part included the absorber column and absorber sump, and the other part included the heat exchanger, desorber, condenser and reboiler. It was first attempted to control the $CO_2$ capture plant by using two NMPCs in tandem, each running one of the parts of the mechanistic model. This did not work, as the desorber must know how rich loading and flow are affected by its inputs to determine minimum reboiler duty. In order to accomplish this, a simplified linear absorber model was developed based on the step response of the original absorber model. This simplified model was added to the desorber part of the model. This resulted in three different control configurations. The first one had the complete mechanistic model in the NMPC and served as a benchmark for the other

configurations. The second had the desorber part with the simplified absorber model. The third consisted of two NMPCs, one with the desorber part with the simplified absorber model and one with the absorber part of the model. All three configurations were tested using the original mechanistic model of the whole plant as plant replacement in the simulator. The last control structure delivered acceptable performance while giving a significant reduction in computational time and laid the foundation for the work in this thesis.

## 1.3 Structure of the thesis

The main part of this master's thesis is split into nine chapters. The first chapter is the introduction, which contextualises the thesis and gives a summary of the specialisation project which laid the foundation for the work in this thesis. Chapter 2 introduces the NMPC concept and gives the relevant theory. Chapter 3 gives a description of the $CO_2$ capture plant and formulates the control objective. Chapter 4 describes the models, both the models received form Cybernetica and the simple absorber models developed during this thesis. Chapter 5 describes the work process behind developing the control configurations. Chapter 6 gives a more detailed description of the final control configurations. Chapter 7 shows the simulation results. Chapter 8 contains discussion of the development process and results, as well as other considerations. Chapter 9 consists of concluding remarks as well as suggestions for future work.

Chapter 2

# Nonlinear model predictive control

Model predictive control (MPC) is a form of advanced process control where the current control action is determined by solving a finite horizon open-loop optimal control problem. The current state is used as the initial condition, and the optimisation yields an optimal control sequence for the horizon. The first input is applied to the plant, a new state is obtained, and the optimisation problem is solved once again. This is done at each sampling instant. When the process model is nonlinear, it is known as nonlinear model predictive control (NMPC). The algorithm is given in Algorithm 1. This type of control, where only the first input is applied, is also known as receding horizon control (RHC).

**Algorithm 1:** State feedback NMPC procedure[15]

**for** $k = 0, 1, 2, \ldots$ **do**
  Get the current state $x_k$.
  Solve a dynamic optimization problem on the prediction horizon from $k$ to $k + P$ with $x_k$ as the initial condition.
  Apply the first control move $u_k$ from the solution above
**end**

Cybernetica's tool for NMPC, CENIT[1], has been used in this thesis. Cybernetica CENIT is a powerful software suite for NMPC which utilises nonlinear mechanistic process models. The nonlinear in NMPC refers to the process model, not the optimisation problem, although the optimisation problem is also nonlinear. The formulation of the optimisation problem and how it is solved will be given later in the chapter. Figure 2.1 shows the different components of CENIT and how they work together. The estimator uses the measurements estimated by the model to correct states and parameters in the model, and the NMPC algorithm uses the model when solving the finite horizon open-loop control problem. Plant can either be the real physical plant or a model in a process simulator such as Cybernetica RealSim. The communication between CENIT and RealSim goes through an Open Platform Communications (OPC) server. More information about CENIT, RealSim and the OPC server is given

[1] http://cybernetica.no/technology/model-predictive-control/

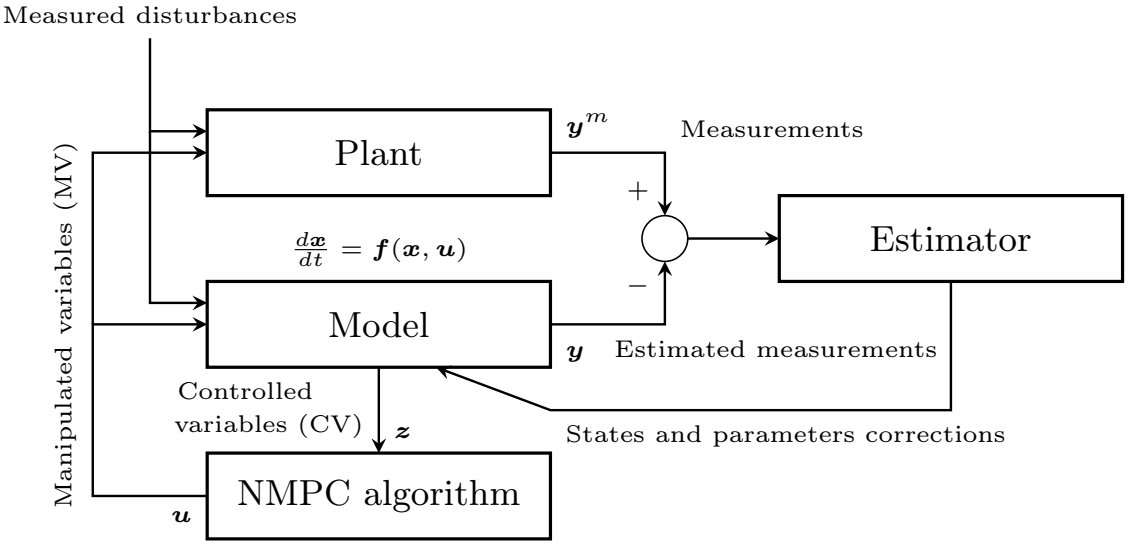



Figure 2.1: The NMPC system, consisting of nonlinear process model, an online estimator and the NMPC algorithm[9]

in Appendix B.

The process model can be expressed as a system of ordinary differential equations (ODE),

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{2.1a}$$

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}), \tag{2.1b}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}). \tag{2.1c}$$

$\boldsymbol{x}$ is a vector of the $n_x$ state variables and $\boldsymbol{x}_0$ is the initial state. $\boldsymbol{u}$ is a vector of the $n_u$ inputs or manipulated variables (MV) and disturbance variables (DV). $\boldsymbol{z}$ is a vector of the $n_z$ outputs or controlled variables (CV), while $\boldsymbol{y}$ is a vector of the $n_y$ calculated measurements.

## 2.1 Mathematical description of NMPC

As mentioned above, the current control action is determined by solving a finite horizon open-loop control problem. The mathematical formulation of this problem can vary, but the one implemented in CENIT and used in this thesis is given in Equation 2.2. The model in Equation 2.1 is solved in discrete time steps by the numerical integrator, the CVs and predicted measurements are calculated at the same discrete time steps.

$$\min_U J = \frac{1}{2}(Z - Z_{ref})^T Q(Z - Z_{ref}) + \frac{1}{2}\Delta U^T S \Delta U + \boldsymbol{r}_1^T \boldsymbol{\epsilon} + \frac{1}{2}\boldsymbol{\epsilon}^T \mathrm{diag}(\boldsymbol{r}_2)\boldsymbol{\epsilon} \tag{2.2a}$$

subject to

$$\boldsymbol{x}_{k+j} = \boldsymbol{f}(\boldsymbol{x}_{k+j-1}, \boldsymbol{u}_{k+j-1}, \boldsymbol{\nu}_k), \tag{2.2b}$$

$$\boldsymbol{z}_{k+j} = \boldsymbol{h}(\boldsymbol{x}_{k+j}, \boldsymbol{u}_{k+j}), \tag{2.2c}$$

$$Z_{\min} - \boldsymbol{\epsilon} \leq Z \leq Z_{\max} + \boldsymbol{\epsilon}, \tag{2.2d}$$

$$0 \leq \boldsymbol{\epsilon} \leq \boldsymbol{\epsilon}_{\max}, \tag{2.2e}$$

$$U_{\min} \leq U \leq U_{\max}, \tag{2.2f}$$

$$\Delta U_{\min} \leq \Delta U \leq \Delta U_{\max}, \tag{2.2g}$$

where

$$Q \succeq 0, \tag{2.2h}$$

$$S \succeq 0, \tag{2.2i}$$

$$\boldsymbol{r}_1 \geq 0, \tag{2.2j}$$

$$\boldsymbol{r}_2 \geq 0, \tag{2.2k}$$

$$Z = \begin{bmatrix} \boldsymbol{z}_{k+1}^T & \boldsymbol{z}_{k+2}^T & \cdots & \boldsymbol{z}_{k+P}^T \end{bmatrix}^T, \tag{2.2l}$$

$$U = \begin{bmatrix} \boldsymbol{u}_k^T & \boldsymbol{u}_{k+1}^T & \cdots & \boldsymbol{u}_{k+M-1}^T \end{bmatrix}^T, \tag{2.2m}$$

$$\Delta U = \begin{bmatrix} \Delta\boldsymbol{u}_k^T & \Delta\boldsymbol{u}_{k+1}^T & \cdots & \Delta\boldsymbol{u}_{k+M-1}^T \end{bmatrix}^T, \tag{2.2n}$$

$$\Delta\boldsymbol{u}_k^T = \boldsymbol{u}_k^T - \boldsymbol{u}_{k-1}^T. \tag{2.2o}$$

The vector $\boldsymbol{z}_{k+j}$ is a column vector of the $n_z$ CVs at time step $k+j$. All the $P$ smaller vectors, $\boldsymbol{z}_{k+j}$, $j = 1, \ldots, P$, are contained in the larger vector $Z$, which is present in the formulation of the objective function. $\boldsymbol{u}_k$ is a column vector of the $n_u$ inputs at time step $k$. $\boldsymbol{x}_k$ is a vector of states at time step $k$. All the $M-1$ smaller input vectors, $\boldsymbol{u}_{k+j}$, $j = 0, \ldots, M-1$, are contained in $U$. $U$ is present in the objective function and contains both MVs and DVs. Equation 2.2a is the objective function that is minimised to determine the optimal input sequence. In this case, there are quadratic penalties on setpoint deviation and input change and both linear and quadratic penalties on constraint violations. Other terms can also be added. In, for example, Equinor's MPC tool SEPTIC (Statoil Estimation and Prediction Tool for Identification and Control), there is a term in the objective function which aims to keep the inputs at ideal values[16]. In Equation 2.2b, $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\nu})$ is the process model, this constraint ensures that the solution satisfies the process model. In Equation 2.2c, $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$ is a function for calculating the CVs based on the states and inputs. Equation 2.2d are maximum and minimum constraints on the CVs. These constraints are soft constraints by the addition of $\boldsymbol{\epsilon}$, which means that they can be violated. However, the violation is penalised by the addition of the last two terms of the objective function. There can be a limit to how much the constraints can be violated, given by $\epsilon_{max}$. There are constraints on the values which

can be chosen for the inputs. These are hard constraints and cannot be violated. There are also constraints on how much the inputs can change in one control action. The constraints on input values and change can come from the physical limitations of the plant or a design choice.

$P$ is the amount of CV evaluation points and $M$ is the amount of MV blocked intervals. $Q$ and $S$ are diagonal matrices of weights for setpoint deviation and input change. $Q$ is a $(P \cdot n_z) \times (P \cdot n_z)$ matrix and $S$ is a $(M \cdot n_u) \times (M \cdot n_u)$ matrix. $Q$ has a repeating diagonal and can be thought of as the Kronecker product between a $P \times P$ identity matrix and a $n_z \times n_z$ diagonal matrix of weights for each CV. The same applies for $S$, except it is the Kronecker product between a $M \times M$ identity matrix and a $n_u \times n_u$ diagonal matrix of weights. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of nonzero elements which give the linear and quadratic weights on CV constraint violations. They are both $P \cdot n_z$ dimensioned vectors with $P$ repeating subvectors of $n_z$ elements.

## 2.2 Graphical illustration

A graphical illustration of NMPC is given in Figure 2.2. This is a simple example where the NMPC has one input and one output, also known as a single input single output (SISO) system. In this example, the task of the NMPC is to keep the CV, $z$, at its reference value, $z^{ref}$. This is done by solving the finite horizon open-loop control problem as described above.

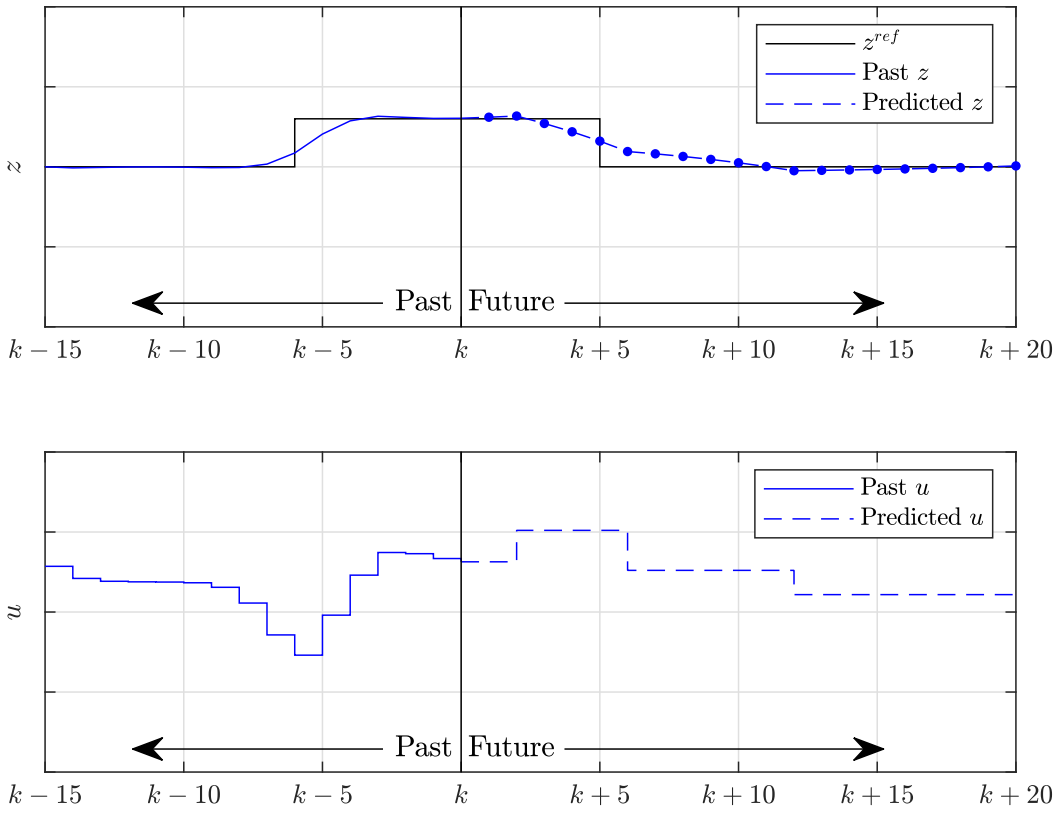



Figure 2.2: Graphical illustration of NMPC[17]

The upper plot in Figure 2.2 shows the CV. In the left part of the plot, the history is shown. In the right part of the plot, the future behaviour as predicted by the NMPC is shown. In the plot, there are several dots that indicate the location of the $P$ number of CV evaluation points. In this case $P = 20$ so we have $z_{k+1}$ to $z_{k+20}$ as the CV evaluation points. In this example, the evaluation points are evenly spaced, but that does not necessarily have to be the case. One could, for example, have more evaluation points at the beginning of the prediction horizon. The lower plot in Figure 2.2 shows the input history and the predicted optimal input trajectory. In the right-hand part of the plot, it can be seen that the inputs are grouped in blocks of increasing size. This is known as input blocking, and it is used to reduce the degrees of freedom in the optimisation problem, which will reduce the computational time.

## 2.3 Nonlinear programming

The open-loop finite-horizon optimisation problem is solved by choosing the inputs which minimise the objective function. In the case where the process model $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\nu})$ is nonlinear this becomes a nonlinear programming (NLP) problem. A general constrained optimisation problem is given in Equation 2.3.

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}), \tag{2.3a}$$

subject to

$$c_i(\boldsymbol{x}) = 0, \quad i \in \mathcal{E}, \tag{2.3b}$$
$$c_i(\boldsymbol{x}) \geq 0, \quad i \in \mathcal{I}, \tag{2.3c}$$

where the objective function $f$ and the constraints $c_i$ are all smooth, real-valued functions on a subset of $\mathbb{R}^n$. $\mathcal{E}$ and $\mathcal{I}$ are index sets of equality and inequality constraints, respectively. In the mathematical description of NMPC given in Section 2.1 the constraints in Equation 2.2b and Equation 2.2c would be in $\mathcal{E}$, while the constraints from Equation 2.2d to Equation 2.2g would be in $\mathcal{I}$. The feasible set $\Omega$ is the set of all state vectors $\boldsymbol{x}$ that satisfy the constraints,

$$\Omega = \{\boldsymbol{x} \,|\, c_i(\boldsymbol{x}) = 0,\, i \in \mathcal{E};\, c_i(\boldsymbol{x}) \geq 0,\, i \in \mathcal{I}\}. \tag{2.4}$$

$\boldsymbol{\epsilon}$ is subtracted on the left-hand side and added on the right-hand side of Equation 2.2d to expand the feasible set to ensure that the NMPC can find a solution.

## 2.4 Formulation of the optimisation problem

In Cybernetica CENIT, the optimisation problem is formulated using the sequential approach method. This method is also known as *single shooting* or *reduced space*. When the sequential approach is used, only the inputs in $U$ are decision variables in the optimisation problem. First, for a given $U$, the simulation routine computes trajectories of states, $\boldsymbol{x}_{k+j}$ for all $k+j$ in the prediction horizon. It also computes the trajectories of the CVs, $Z$ from Equation 2.2c. Secondly, the optimisation routine updates the values of $U$ to iterate towards the optimal solution. This sequence of simulation and optimisation is repeated until the optimal solution is found, hence the name, sequential approach.

The sequential approach method is known as *reduced space* since only the elements of $U$ are decision variables. In the simultaneous approach, on the other hand, both $U$ and $Z$ and the states are decision variables. The simultaneous approach is also known as *full space*. In the simultaneous approach, both simulation and optimisation are performed simultaneously, hence the name. The name *single shooting* comes from the fact that after determining $U$, the simulator "shoots" the model until the end of the prediction horizon.

Conversely, there is another method known as *multiple shooting*. In *multiple shooting*, the model is "shot" a short interval at a time, so multiple shots are required to reach the end of the prediction horizon. In *multiple shooting*, the states at the intervals are kept as decision variables. Therefore, this method can be seen as a combination of the sequential and the simultaneous approach. New constraints are added to ensure that the end state of one interval matches the initial state of the next interval[18].

The optimisation problem is solved using sequential quadratic programming (SQP). SQP is an iterative method for solving constrained nonlinear programs. SQP methods generate steps by solving quadratic subproblems and can be used both with line search and trust-region frameworks.

## 2.5 Estimation

One function of the estimator block in Figure 2.1 is to update the states and parameters based on the difference between measurements from the plant and predicted measurements from the model. There are several forms of estimation algorithms available. Examples of these include the Kalman filter (KF), extended Kalman filter (EKF) and the moving horizon estimator (MHE)[19]. The EKF is a built-in component of CENIT[9]. It is not always possible or necessary to use the EKF, then there are simpler estimation algorithms such as bias updating.

### 2.5.1 Model updating

Process models can be updated using bias updating. The values of the bias variables are updated based on the difference between measurements from the plant, $\boldsymbol{y}^m$, and measurements

predicted by the model, $\boldsymbol{y}$. Including bias updating gives integral action and will remove the offset between the measurements and the predictions. The rule for updating the bias variables is

$$\boldsymbol{\beta}_{x,k} = \boldsymbol{\beta}_{x,k-1} + K_{\boldsymbol{\beta}_x}(\boldsymbol{y}_k^m - \boldsymbol{y}_k), \tag{2.5}$$

where $\boldsymbol{\beta}_{x,k}$ is the bias variable vector at time step $k$, $K_{\boldsymbol{\beta}_x}$ is a matrix of gains for bias updating, $\boldsymbol{y}_k^m$ is the measurement vector and $\boldsymbol{y}_k$ is the measurement vector predicted by the model. $K_{\boldsymbol{\beta}_x}$ is a $n_x \times n_y$ matrix, the values in $K_{\boldsymbol{\beta}_x}$ will determine the speed of the updating and which measurements are used to updated which states. If a value is set to 0, there is no bias updating. The bias variable can be included in the process model. This gives

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_k, \boldsymbol{\beta}_{x,k}), \tag{2.6}$$

where $\boldsymbol{x}_{k+1}$ is the new state vector and $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_k, \boldsymbol{\beta}_{x,k})$ is the process model. As the process model is derivative of the state, adding a bias variable will affect the whole state vector $\boldsymbol{x}_{k+1}$. This is closely related to the EKF, where a physical parameter is updated. The bias variable can represent a physical parameter or a physical deviation, or an empirical correction.

### 2.5.2 Updating controlled variables

The most important job of the controller is to keep the CVs at their setpoints. The estimator can help with this by ensuring offset-free control. Offset-free control means zero deviation between the CVs and their setpoints. The bias variable is updated in a similar manner as before:

$$\boldsymbol{\beta}_{z,k} = \boldsymbol{\beta}_{z,k-1} + K_{\boldsymbol{\beta}_z}(\boldsymbol{y}_k^m - \boldsymbol{y}_k), \tag{2.7}$$

where $\boldsymbol{\beta}_{z,k}$ is the bias variable at time step $k$, $K_{\beta_z}$ is the gain matrix of the bias updating, $\boldsymbol{y}_k^m$ is the measurement vector from the plant and $\boldsymbol{y}_k$ is the predicted measurement vector. The bias term is added directly on the CVs

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\beta}_{z,k}, \tag{2.8}$$

where $\boldsymbol{z}_k$ is the CV vector at time step $k$, $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is a function for calculating the CVs based on the state vector, $\boldsymbol{x}_k$, and the input vector, $\boldsymbol{u}_k$. This bias update feedback technique is a common method used by industrial MPC controllers. For stable processes, the constant output disturbance model provides integral action to the controller[19]. The bias variable

must also be added to the corresponding measurement prediction

$$y_k = g(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\beta}_{z,k} = z_k, \tag{2.9}$$

to ensure convergence.

## 2.6 Controller tuning

When tuning a PID controller, there are different tuning rules which can be used, such as Skogestad Internal Model Control (SIMC) [20] or the Ziegler-Nichols method [21]. When tuning an NMPC there are no such rules. Instead, one must rely on process knowledge, experience and experimenting through trial and error. There are several tuning parameters in an NMPC. The maximum and minimum constraints on $Z$ and $U$ are usually determined from knowledge of the process. The maximum and minimum constraints on $\Delta U$ are there to ensure that the inputs determined by the controller can be realised. $\boldsymbol{\epsilon}_{\max}$ is often set to infinity to avoid infeasibility, but can also be limited. The values of the weights in $Q$ and $S$ will affect the optimal solution, as they are in the formulation of the objective function in Equation 2.2a. The values should be chosen based on the control objective and how much input usage is desired. Since the control variables and inputs do not necessarily have the same scale, the weights have to be scaled. In CENIT, this is done by dividing the weights by the square of a scaling factor. The scaling factor is referred to as span and is the range in which the variables vary. The weights in $Q$ are then weight $= q/zSpan^2$, where $q$ is a tuning parameter and $zSpan$ is the span of the relevant CV. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ give the penalties for violating the constraints and can be used to prioritise between the constraints. Other tuning parameters include the length of the prediction horizon, parameterisation of the CV evaluation points and the input blocking.

Chapter 3

# Process plant description

A post-combustion $CO_2$ capture plant can consist of several process units. The theoretical plant in this thesis consists of two/three absorber columns, a heat exchanger, a desorber column and a buffer tank for lean solvent. In a $CO_2$ capture plant such as this, the solvent is circulated in the plant. The mole amount of $CO_2$ per mole of amine is known as loading and varies in the plant. The flow with low $CO_2$ content is referred to as lean flow, and its loading is known as lean loading ($\alpha_L$). The flow with high $CO_2$ content is referred to as rich flow, and its loading is known as rich loading ($\alpha_R$). The lean solvent is injected at the top of the absorber columns, absorbs $CO_2$, and rich solvent leaves the absorber. The rich solvent is sent back to the desorber column where $CO_2$ is removed, and the lean solvent is regenerated. Figure 3.1 shows a simple piping and instrumentation diagram (P&ID) of the process. The valves marked in red are the MVs in the NMPCs.

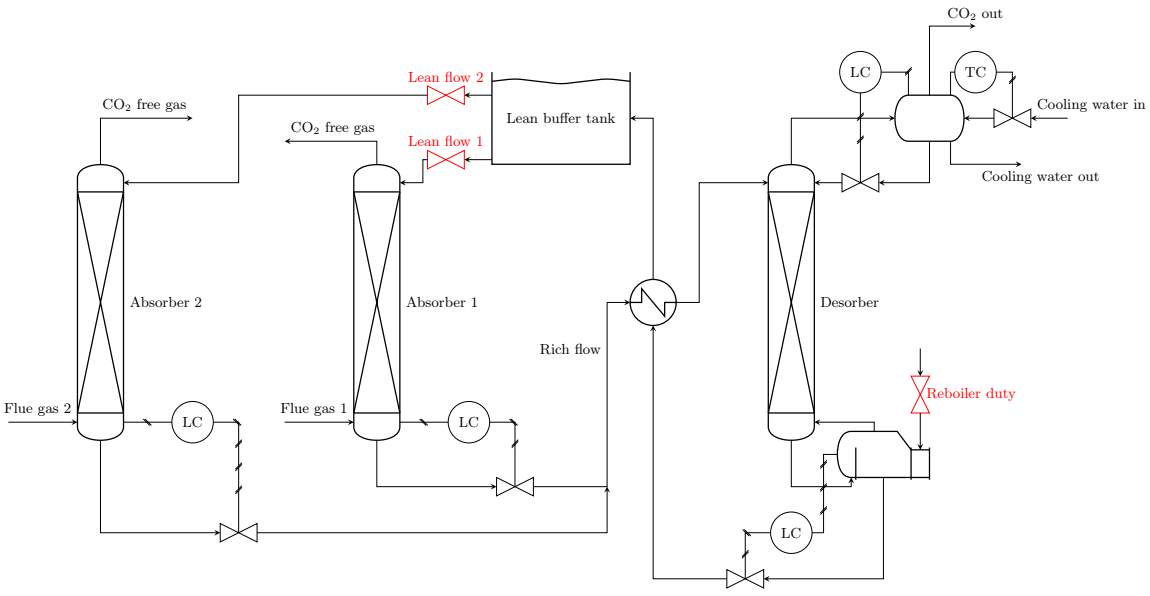



Figure 3.1: P&ID of the process. The valves marked in red are the MVs

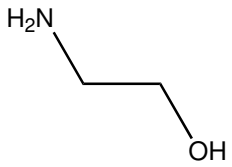



Figure 3.2: Skeletal formula of MEA

## 3.1 Absorbers

There are two/three absorber columns in the theoretical $CO_2$ capture plant. In the absorber columns, $CO_2$ is removed from the flue gas by chemical absorption. $CO_2$ rich flue gas is injected at the bottom of the column and contacted counter-currently with the lean solvent flowing down from the top of the columns. Absorber columns are kept at high pressure and low temperature, as this is beneficial with regards to the solubility of $CO_2$ in the solvent. There are several solvents suitable for post-combustion $CO_2$ capture. The most well-known solvent is monoethanolamine (MEA), which is a primary amine. The net reaction between $CO_2$ and a primary amine ($RNH_2$) is[22]

$$CO_2 + 2\,RNH_2 \rightleftharpoons RNHCOO^- + RNH_3{}^+. \tag{3.1}$$

When the amine is MEA, the R refers to the alkanol group $OHCH_2CH_2$. The skeletal formula of MEA is given in Figure 3.2. The solvent used in the theoretical plant in this thesis is called CESAR 1. $CO_2$ Enhanced Separation and Recovery (CESAR) was a project aimed at making breakthroughs in the development of low-cost post-combustion $CO_2$ capture technology. As part of this project, several solvents were tested at different facilities. Of these solvents, CESAR 1 performed the best. CESAR 1 is a mixture of 2-amino-2-methylpropan-1-ol (AMP) and piperazine (PZ)[23]. In the models the ratio between AMP and PZ is 2:1. The skeletal formulas of the two components are given in Figure 3.3.

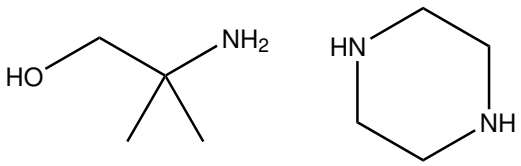



Figure 3.3: Skeletal formulas of the components of CESAR 1, AMP to the left and PZ to the right

Absorption using chemical solvents is the preferred method when the partial pressure of $CO_2$ is low, such as it often is when performing post-combustion $CO_2$ capture. The rich solvent in the absorber sumps at the bottom of the columns is sent back to the desorber through the heat exchanger, where it is heated up by the hot lean flow from the reboiler. The purified flue gas is passed through a water wash section before being emitted at the top of the absorber

columns. The capture ratio (CR) in absorber column $i$ is

$$CR_i = 100\frac{F_{i,in}^{\mathrm{CO_2}} - F_{i,out}^{\mathrm{CO_2}}}{F_{i,in}^{\mathrm{CO_2}}}, \tag{3.2}$$

where $F_{i,in}^{\mathrm{CO_2}}$ is the amount of $CO_2$ entering the bottom of absorber column $i$ and $F_{i,out}^{\mathrm{CO_2}}$ is the amount of $CO_2$ leaving the top absorber column $i$.

## 3.2 Desorber

In the desorber, the lean solvent is regenerated by removing $CO_2$. The desorber is operated at low pressure and high temperature, as this favours $CO_2$ stripping. Heat is supplied in the reboiler where the solvent is boiling. $CO_2$ rises through the column up to the condenser along with gaseous solvent. In the condenser, the solvent condenses while the $CO_2$ is removed in gas form. The hot lean solvent in the reboiler is sent from the reboiler to the heat exchanger. The hot lean solvent is heat integrated with the cold rich solvent from the absorbers in the heat exchanger. After the heat exchanger, the lean solvent is sent to the lean buffer tank. The CR in the desorber is

$$CR_{des} = \frac{F_{des,out}^{\mathrm{CO_2}}}{\sum_{i=1}^{N} F_{i,in}^{\mathrm{CO_2}}}, \tag{3.3}$$

where $F_{des,out}^{\mathrm{CO_2}}$ is the amount of $CO_2$ leaving the top of the desorber column, and $N$ is the number of absorber columns.

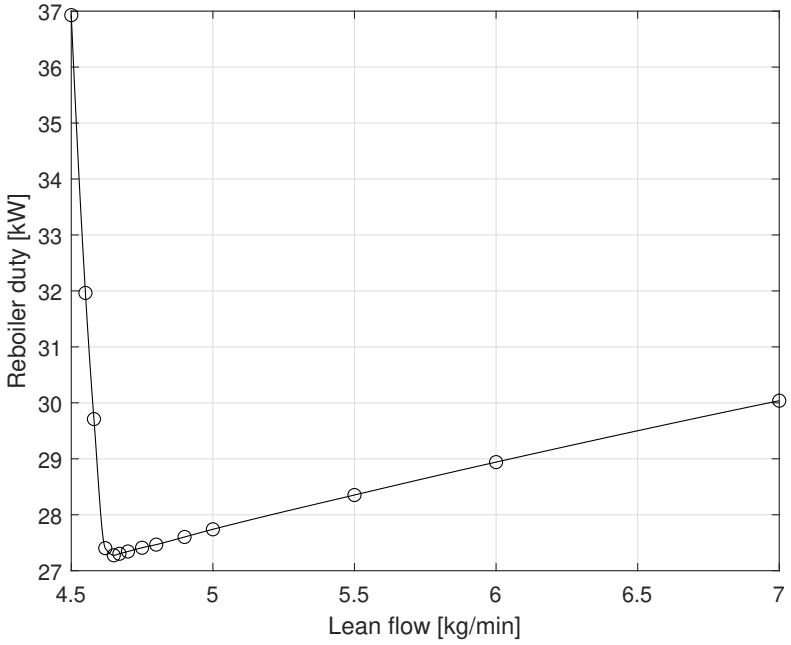



Figure 3.4: U-curve for a plant with one absorber with constant CR at 84 %[17]

## 3.3 Optimal operation of the plant

The NMPCs developed in this thesis have three/four MVs to control the plant, depending on the number of absorber columns. The control objective of the NMPCs is to keep the CRs at a given setpoint while using minimum reboiler duty. The CR setpoints can be achieved with an infinite amount of MV combinations, but only one of the combinations will give minimum reboiler duty. The objective function in the NMPC is formulated in a way that should result in the combination that utilises the lowest reboiler duty. An illustration of this problem is given in Figure 3.4, which shows a U-curve for a plant with only one absorber column. Lean flow is on the first axis, and reboiler duty is on the second axis. The figure shows the lean flow and the corresponding reboiler duty needed to achieve a CR of 84 %. In this case, minimum reboiler duty is achieved when the lean flow is just over $4.6\,\mathrm{kg\,min^{-1}}$. From the plot, it can be seen that choosing a lean flow that is lower than the optimum is particularly disadvantageous due to the steep gradient to the left of the minimum. If a NMPC is to determine the minimum reboiler duty, both lean flows and reboiler duty need to be part of the optimisation problem. Hence the NMPC needs to have both the flows and reboiler duty as MVs.

Chapter 4

# Models

The models used in this master's thesis are based on a model received from Cybernetica AS in June 2020. This model is based on work done by Enaasen Flø[6], but has since been modified and improved by Cybernetica AS, such as in the DOCPCC project[7]. The model from Cybernetica AS is split into four submodels, one for the absorber, one for the desorber, one for the heat exchanger and one for the lean and rich buffer tanks.

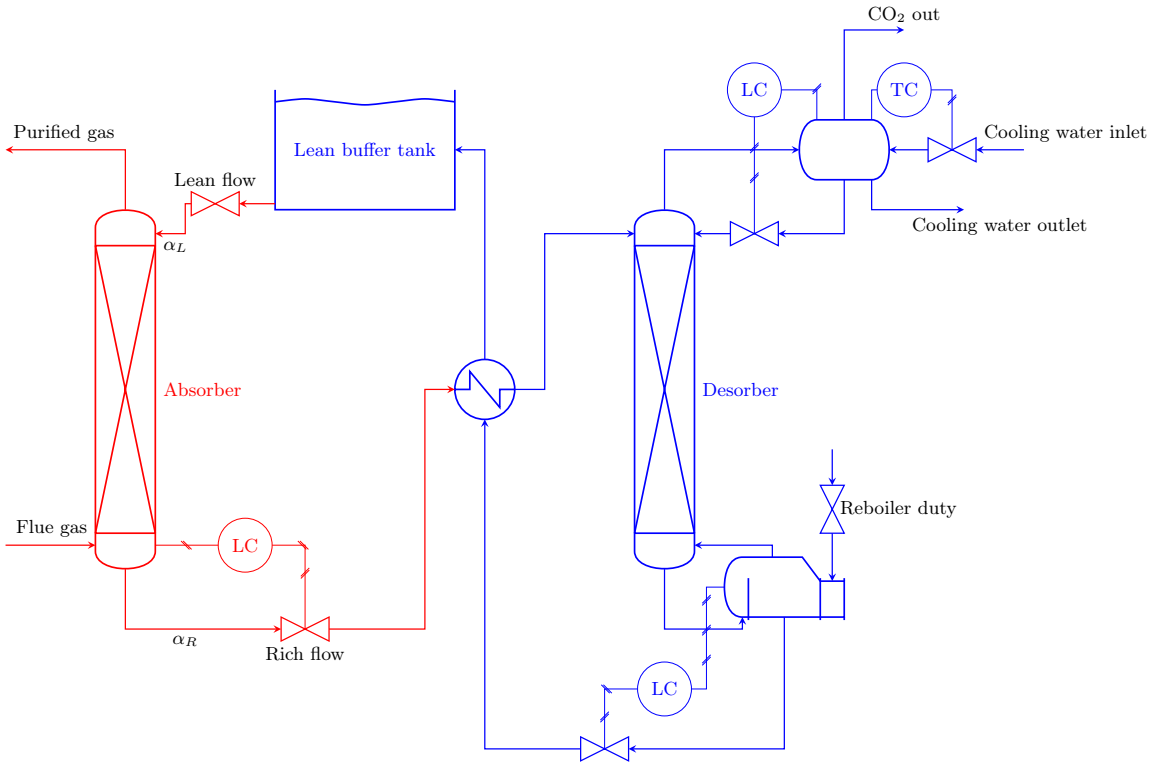



Figure 4.1: Simple P&ID showing how the plant was split into two separate applications

## 4.1 About the model

The model supplied from Cybernetica AS consists of models of all the major process units. The models are mechanistic and developed from first principles conservation laws. This model will be referred to as the *original model* in the rest of this thesis. The model was split into two parts in the project thesis, one with the absorber and one with the desorber, lean buffer tank and heat exchanger. The split is illustrated in Figure 4.1, where all units with the same colour are in the same model. The red part will later be referred to as the *absorber model* while the blue part will be referred to as the *desorber model*. The reason behind splitting the *original model* into multiple parts is that as the model increases in size by adding more absorber columns, it becomes infeasible to handle the whole model in one application. This is due to the increase in model complexity and computational time. Another reason is that in the case where there are several absorbers at different locations in an industrial park with different owners and base control systems, it might be a problem to have the whole cluster in one common application.

A simple absorber model based on the step-response of the *absorber model* while controlled was added to the *desorber model*. Looking at Figure 3.4 it can be seen that in order to determine minimum reboiler duty, both the reboiler duty and the lean flow must be part of the optimization problem. A model which includes models of all the process units in the plant is needed to know how the two inputs affect the whole plant. Because of this, the simple absorber model was added to the *desorber model*. In this master's thesis, the models are further expanded to include two/three absorber columns, along with their respective absorber sumps. During the master's thesis work, several simple absorber models were developed and tested. All the different models are presented here.

## 4.2 Step response model

The step response of a process is the time evolution of its outputs for a given initial state when its inputs are Heaviside step functions. The step response of a process gives insight into its behaviour. If the process can be approximated as a first-order linear model, then the model parameters can be determined by inspecting the step response. The step response of a generic first-order-plus-time-delay model (FOPTD) is given in Figure 4.2. The time constant, $\tau$, time delay, $\theta$ and gain $K$ are indicated in the figure. The FOPTD model itself in the Laplace domain is

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}. \tag{4.1}$$

The process is often neither first-order nor linear. The output often has noise in the measurements, there can be disturbances affecting the process in an unknown manner, and it can be difficult to achieve steps in the input. Due to these reasons, the step response of a process rarely exhibits exactly first-order behaviour. It was believed that much of the nonlinearity

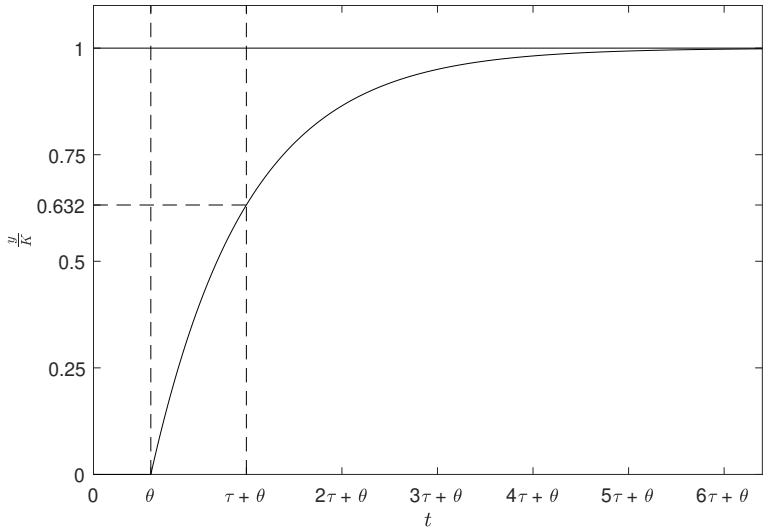



Figure 4.2: Step response of a generic FOPTD model [17]

of the plant was related to the desorber and that the absorber columns by themselves were quite linear when the capture ratio was constant. A step response test was performed on the *absorber model*. Figure 4.3 shows the step response of the mechanistic *absorber model* to steps in lean mass flow into the top of the absorber when the capture ratio is controlled at a setpoint. The flue gas flow in is $80\,\mathrm{m^3\,h^{-1}}$ and the amount of $CO_2$ is 12.3 %. The loading of the lean mass flow entering the top of the column is used as an MV in an NMPC, with the CV being the CR, which should be at 84 %. Lean loading is not an actual MV in the plant, but it makes sense to use it when only the absorber model is run, as it is in a way what is controlled when manipulating the reboiler duty. The responses of rich flow and rich loading seems to be quite linear. Hence linear models were fitted to these outputs. The formulation of the linear models is

$$\dot{x} = \frac{1}{\tau}\left(K(u - u_0) - (x - x_0)\right). \tag{4.2}$$

This model gives the state $x$'s response to changes in the input $u$. $\tau$ is the time constant of the process, $K$ is the gain of the process, $x_0$ is the nominal value of the state, and $u_0$ is the nominal value of the input. The model parameters were found by using linear regression. Figure 4.4 shows a plot of data points for rich loading and rich flow at different lean flow levels and lines from linear regression. The lines were found using the function `fitlm` from Statistics and Machine Learning Toolbox in MATLAB. $R^2$ is 1 for both models. The model values are given in Table 4.1.

Table 4.1: Model parameters in the two simplified models

| Model | $K$ [$\mathrm{min\,kg^{-1}}$/-] | $\tau$ [min] | $x_0$ [-/$\mathrm{kg\,min^{-1}}$] | $u_0$ [$\mathrm{kg\,min^{-1}}$] |
|---|---|---|---|---|
| Rich loading | -0.046861 | 25 | 0.6402 | 2.2 |
| Rich flow | 1.0005 | 20 | 2.4411 | 2.2 |

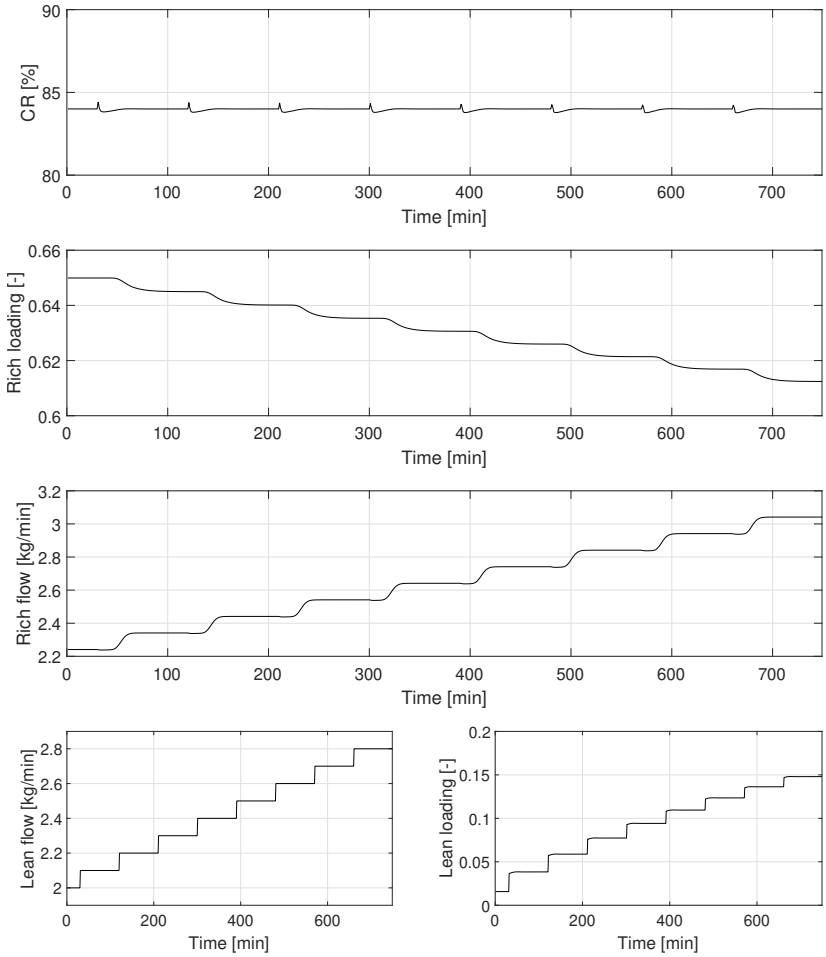



Figure 4.3: Step response of the *absorber model* when steps are done on lean flow while lean loading is used to control the CR

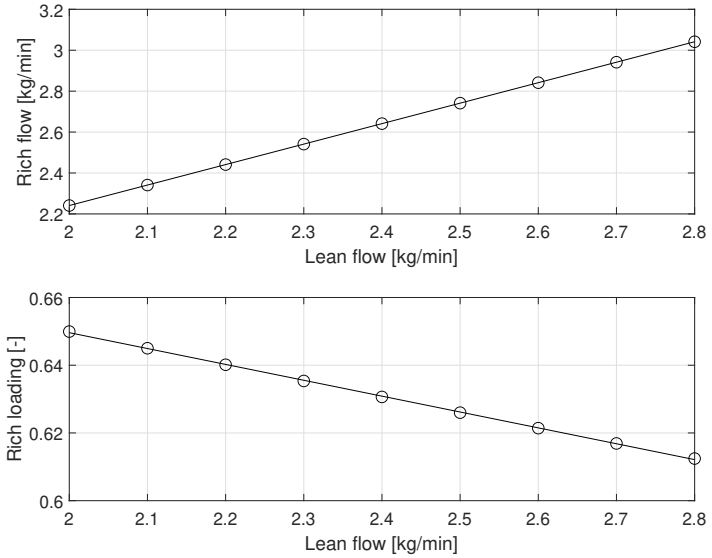



Figure 4.4: Datapoints and linear regression lines for rich loading and rich flow as functions of lean flow in an absorber column controlled to 84 % CR

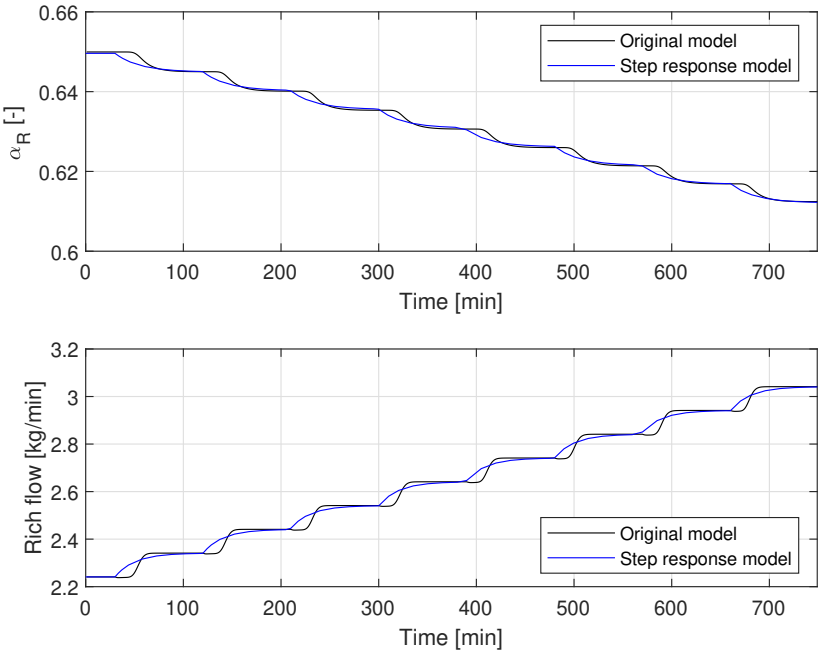



Figure 4.5: Comparison of the original *absorber model* with the step response model

The step response models for rich loading and rich flow compared to the original mechanistic absorber model is given in Figure 4.5. The steady state values seem to correspond well, but dynamically it is a bit off.

## 4.3 Absorber model based on mass transfer

The mass flux of $CO_2$ from gas to liquid is given by

$$J = k_y(y - y_I) = k_x(x_I - x), \tag{4.3}$$

where $y$ is the bulk gas mole fraction and $x$ is the bulk liquid mole fraction. The equilibrium $y_I = mx_I$ is assumed at the interface according to Henry's law, where $m$ is an equilibrium constant for $CO_2$. The concentrations at the interface are unknown, but they can be eliminated by rearranging Equation 4.3. This gives

$$\frac{J}{k_y} = y - y_I, \tag{4.4}$$

$$\frac{J}{k_x} = x_I - x. \tag{4.5}$$

By multiplying Equation 4.5 by the equilibrium constant $m$ and adding the two equations,

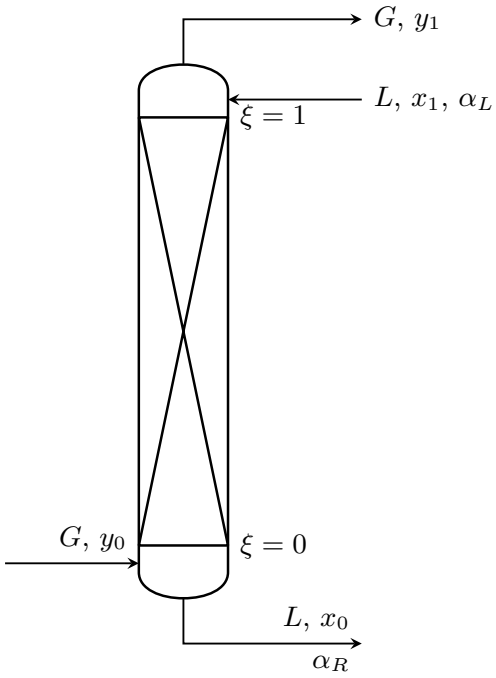



Figure 4.6: Arbitrary absorber column

we get

$$J\left(\frac{1}{k_y}+\frac{m}{k_x}\right)=y-mx. \tag{4.6}$$

The mass flux can be written as a driving force from the equilibrium

$$J=K_y(y-mx), \tag{4.7}$$

where

$$K_y=\left(\frac{1}{k_y}+\frac{m}{k_x}\right)^{-1}. \tag{4.8}$$

Figure 4.6 shows an arbitrary absorber column. $L$ and $G$ are total molar flows of liquid and gas in $\mathrm{kmol\,h^{-1}}$. $x$ and $y$ are mole fractions of $CO_2$ in liquid and gas, respectively. $\xi$ is a dimensionless length coordinate, $\xi = z/H$. $\alpha_L$ and $\alpha_R$ are lean and rich loading, respectively. Both $L$ and $G$ are assumed to be plug flows, implying no axial dispersion. Further, it is assumed that the column is isothermal in order to avoid adding a temperature equation. Using $A$ as the total gas/liquid interfacial area for the column gives a mole balance for $CO_2$ in the gas phase

$$\frac{d(Gy)}{d\xi}=-JA. \tag{4.9}$$

When the liquid is flowing counter currently to the gas phase, the mole balance in the liquid phase is

$$\frac{d(Lx)}{d\xi} = -JA. \tag{4.10}$$

Since the amount of $CO_2$ is low it is assumed that the total molar flows $L$ and $G$ are constant. Inserting Equation 4.7 for $J$ and introducing $y^* = mx$ gives the system of ODEs in Equation 4.11.

$$\frac{dy}{d\xi} = -\frac{K_y}{G}A(y - y^*) \tag{4.11a}$$

$$\frac{dx}{d\xi} = -\frac{K_y}{L}A(y - y^*) \tag{4.11b}$$

The equilibrium mole fraction, $y^*$, is unknown. It can be assumed that the partial pressure of a gas is equal to its mole fraction times the total pressure, $P_i = y_i\, P_{tot}$. Rearranging this expression gives an expression for the gaseous mole fraction, $y_i = P_i/P_{tot}$. The total pressure $P_{tot}$ is known, and the equilibrium partial pressure can be found using correlations dependent on the composition and temperature of the liquid[24]. Cybernetica AS provided correlations for the CESAR 1 solvent. Combining this gives $y^* = P_i^*(\boldsymbol{x}, T_L)/P_{tot}$, where $\boldsymbol{x}$ is a vector containing the compositions of the liquid, and $T_L$ is the liquid temperature.

### 4.3.1 Solving the system of ordinary differential equations

The rich loading is found by solving the system of ODEs in Equation 4.11. The composition, mass flow and temperature of both the gas entering the bottom of the column and the lean amine entering the top of the column are known. Since we know states at both ends of the column, this is a boundary value problem (BVP). There are several methods for solving BVPs. In this case, a shooting method was used. In the shooting method, the system is converted to an initial value problem (IVP) and a guess is made for the unknown initial condition. The system is then "shot" or integrated from one boundary to the other, and the guess is updated until the correct value is found. We define $s$ as the guess and introduce the notation $x(1; s)$ which is the value of $x$ at $\xi = 1$ given $s$ was used as the initial condition. The boundary condition is $x(1) = x_1$. Can then define a residual function

$$\phi(s) = x(1; s) - x_1, \tag{4.12}$$

which returns the difference between the end state and the boundary condition as a function of the guessed value for the initial condition. The correct value $s = s^*$ is found when

$$\phi(s^*) = 0. \tag{4.13}$$

The guess can be updated using a root solving algorithm. The IVP was integrated using the classic Runge-Kutta method, given in Algorithm 2, where

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0. \tag{4.14}$$

$t_0$ is the starting value of the independent variable, $t$. $h$ is the step length, and $N$ is the number of steps. This is a fourth-order method with local error $\mathcal{O}(h^5)$ and global error $\mathcal{O}(h^4)$.

**Algorithm 2:** The classic Runge-Kutta method[25]

**Input:** $f(t, y)$, $y_0$, $t_0$, $h$, $N$
**for** $n = 0, 1, 2, \ldots, N-1$ **do**
  $k_1 = f(t_n, y_n)$
  $k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$
  $k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$
  $k_4 = f(t_n + \frac{h}{2}, y_n + hk_3)$
  $y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
  $t_{n+1} = t_n + h$
**end**

For this model $\xi$ is the independent variable, and the system is integrated from $\xi = 0$ to $\xi = 1$.

## 4.4 Simple absorber model based on mole balances

A model based on steady state mole balances was developed and used in two different ways. At first, it was used to calculate predictions for rich flow and rich loading, assuming that the capture ratio is known. It was later used to calculate the capture ratio when both the rich flow and rich loading were assumed to be known.

### 4.4.1 Predicting rich flow and rich loading

A figure representing an arbitrary absorber column is given in Figure 4.7. $V_0$ is the amount of flue gas entering the column, $y_{0,\mathrm{CO_2}}$ is the mole fraction of $CO_2$ in the flue gas. At a given capture ratio $c$, the amount of $CO_2$ transferred from gas to liquid, $F$, equals

$$F = cy_0V_0. \tag{4.15}$$

$L_1$ is the amount of amine solution entering the top of the column, and $\boldsymbol{x}_1$ is a vector of the mole fractions in the amine entering the absorber. The components of $\boldsymbol{x}_1$ are

$$\boldsymbol{x}_1 = \begin{bmatrix} x_{1,\mathrm{CO_2}} & x_{1,\mathrm{H_2O}} & x_{1,\mathrm{amine}} \end{bmatrix}. \tag{4.16}$$

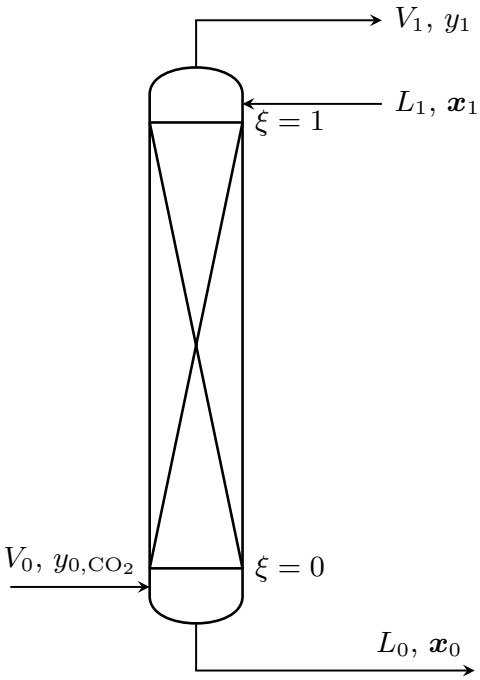



Figure 4.7: Absorber

$\boldsymbol{x}_0$ is formulated in a similar manner. The amount of $CO_2$ in the bottom of the column is then $x_{1,\mathrm{CO_2}}L_1 + F$. The amount of amine solution leaving the bottom of the column, $L_0$, then equals $L_0 = L_1 + F$. The new mole fractions in $\boldsymbol{x}_0$ are

$$x_{0,\mathrm{CO_2}} = \frac{x_{1,\mathrm{CO_2}}L_1 + F}{L_0}, \tag{4.17}$$

$$x_{0,\mathrm{H_2O}} = \frac{x_{1,\mathrm{H_2O}}L_1}{L_0}, \tag{4.18}$$

$$x_{0,\mathrm{amine}} = \frac{x_{1,\mathrm{amine}}L_1}{L_0}. \tag{4.19}$$

From this model there is a prediction of rich flow, $L_0$, and of rich loading, $\alpha_R = x_{0,\mathrm{CO_2}}/x_{0,\mathrm{amine}}$. The value of $c$ can either be a constant or received as a DV.

### 4.4.2 Predicting capture ratio

The model can be converted to calculate the CR. Instead of assuming that the CR is known, we assume that rich flow and rich loading are known. The step response models are used to predict rich flow and rich loading coming for the absorber columns. The CR in the column equals the difference in $CO_2$ content between the rich flow and the lean flow, divided by the amount of $CO_2$ entering the column in the flue gas:

$$CR = \frac{x_{0,\mathrm{CO_2}}L_0 - x_{1,\mathrm{CO_2}}L_1}{y_{0,\mathrm{CO_2}}V_0}. \tag{4.20}$$

$\boldsymbol{x}_1$ is known from the model of the lean buffer tank. It is possible to calculate $x_{0,\mathrm{CO_2}}$ based on rich loading and the mole fractions entering at the top of the column, $\boldsymbol{x}_1$. The sum of the mole fractions equals 1:

$$x_{0,\mathrm{amine}} + x_{0,\mathrm{CO_2}} + x_{0,\mathrm{H_2O}} = 1. \tag{4.21}$$

The ratio between the mole fraction of $CO_2$ and amine is by definition equal to the rich loading, $\alpha_R$. Assuming that no water is lost in the column, the ratio between water and amine is unchanged at the bottom of the column. Inserting these assumptions into the sum of the mole fractions gives

$$\left(1 + \alpha_R + \frac{x_{1,\mathrm{H_2O}}}{x_{1,\mathrm{amine}}}\right) x_{0,\mathrm{amine}} = 1. \tag{4.22}$$

This equation can be rearranged with regards to $x_{0,\mathrm{amine}}$, and with the two expressions we have just used it is possible to get expressions for all the mole fractions:

$$x_{0,\mathrm{amine}} = \frac{1}{1 + \alpha_R + \frac{x_{1,\mathrm{H_2O}}}{x_{1,\mathrm{amine}}}}, \tag{4.23}$$

$$x_{0,\mathrm{CO_2}} = \frac{\alpha_R}{1 + \alpha_R + \frac{x_{1,\mathrm{H_2O}}}{x_{1,\mathrm{amine}}}}, \tag{4.24}$$

$$x_{0,\mathrm{H_2O}} = \frac{x_{1,\mathrm{H_2O}}/x_{1,\mathrm{amine}}}{1 + \alpha_R + \frac{x_{1,\mathrm{H_2O}}}{x_{1,\mathrm{amine}}}}. \tag{4.25}$$

Based on these values the CR can be calculated from Equation 4.20.

## 4.5 Expanding the models to include multiple absorber columns

The *original model* was expanded to include multiple absorber column models. The first step in this process was adding an additional absorber column with all its states, CVs, MVs and DVs to the model. Additionally, all other parameters, functions and everything else needed to run the new absorber model was included. After the new absorber model itself had been successfully included in the code, it was time to connect it to the rest of the plant. There are two places where the absorber columns connect with the rest of the plant. This is at the lean buffer tank, from which the absorber columns receive their lean flow, and at the heat exchanger, where the absorber columns inject their rich flow.

In the lean buffer tank, the change is in what is leaving the buffer tank. From the buffer tanks point of view, it does not matter where the lean flow goes, $10\,\mathrm{kg\,min^{-1}}$ into one absorber is the same as $5\,\mathrm{kg\,min^{-1}}$ into two absorbers. The change in the lean buffer tank model is then

from

$$L_{btl}^{out} = L_{abs}^{in}, \tag{4.26}$$

to

$$L_{btl}^{out} = L_{abs,1}^{in} + L_{abs,2}^{in}, \tag{4.27}$$

where $L_{btl}^{out}$ is the lean flow out of the buffer tank, $L_{abs}^{in}$ is the lean flow into the absorber in a one absorber column plant, and $L_{abs,1}^{in}$ and $L_{abs,2}^{in}$ are the lean flows into absorber column 1 and 2 in a two absorber column plant respectively. For a plant with $N$ absorber columns Equation 4.27 can be written as

$$L_{btl}^{out} = \sum_{i=1}^{N} L_{abs,i}^{in}. \tag{4.28}$$

The heat exchanger receives rich flow from the two absorber columns. When there is only one absorber column, the model is

$$R_{hx,in}^{\mathrm{CO_2}} = x_{as}^{\mathrm{CO_2}}\, R_{as}^{out}, \tag{4.29a}$$
$$R_{hx,in}^{\mathrm{H_2O}} = x_{as}^{\mathrm{H_2O}}\, R_{as}^{out}, \tag{4.29b}$$
$$R_{hx,in}^{\mathrm{Amine}} = x_{as}^{\mathrm{Amine}}\, R_{as}^{out}, \tag{4.29c}$$
$$T_{hx,in}^{R} = T_{as}. \tag{4.29d}$$

$R_{hx,in}^{\mathrm{CO_2}}$, $R_{hx,in}^{\mathrm{H_2O}}$ and $R_{hx,in}^{\mathrm{Amine}}$ are the molar flows of $CO_2$, $H_2O$ and Amine entering the cold side of the heat exchanger respectively. $R_{as}^{out}$ is the total molar flow out from the absorber sump. $x_{as}^{\mathrm{CO_2}}$, $x_{as}^{\mathrm{H_2O}}$ and $x_{as}^{\mathrm{Amine}}$ are the mole fractions in the absorber sump. $T_{hx,in}^{R}$ is the temperature of the liquid entering the cold side of the heat exchanger and $T_{as}$ is the temperature of the liquid in the absorber sump. When there are multiple absorber columns, the molar flows entering the heat exchanger will be the sum of the molar flows out of the absorber sumps. As temperature is an intensive property, it is not as easy as just adding the temperatures together. If one assumes that all the flows have the same heat capacity, the mixture's temperature will equal the sum of the product of total molar flow and temperature divided by the sum of the

total molar flows. For a plant with $N$ absorber columns, this becomes

$$R_{hx,in}^{\mathrm{CO_2}} = \sum_{i=1}^{N} x_{as,i}^{\mathrm{CO_2}} \, R_{as,i}^{out}, \tag{4.30a}$$

$$R_{hx,in}^{\mathrm{H_2O}} = \sum_{i=1}^{N} x_{as,i}^{\mathrm{H_2O}} \, R_{as,i}^{out}, \tag{4.30b}$$

$$R_{hx,in}^{\mathrm{Amine}} = \sum_{i=1}^{N} x_{as,i}^{\mathrm{Amine}} \, R_{as,i}^{out}, \tag{4.30c}$$

$$T_{hx,in}^{R} = \frac{\sum_{i=1}^{N} R_{as,i}^{out} \, T_{as,i}}{\sum_{i=1}^{N} R_{as,i}^{out}}. \tag{4.30d}$$

Similar changes were made in the models using the *desorber model* plus one of the simple absorber models. The difference is in how the values coming from the absorber sumps are found. Expanding the *absorber model* to include multiple absorber columns was a straightforward procedure. As the absorber columns are not connected, no additional modelling was needed. The only work required was adding the column with all its states, CVs, MVs, parameters, functions and other bits and pieces.

Chapter 5

# Developing the control configurations

During the specialisation project[17] it was found that a two-NMPC configuration performed well enough when controlling a $CO_2$ capture plant with one absorber and should be further developed during the master's thesis work. In the configuration, there were two NMPCs, which were called NMPC abs and NMPC des. NMPC abs used the *absorber model*, and NMPC des used the *desorber model* with linear step-response models to predict rich flow and rich loading. NMPC abs's MV was the lean flow into the absorber column, $u_2$, and the CV was the CR in the absorber column, $z_1 = CR$. It received the current value and a predicted trajectory of the lean loading from NMPC des. It also received a trajectory of the lean flow calculated by NMPC des, which it uses as a soft constraint for its own MV. NMPC des's MVs were the reboiler duty, $u_1$ and the lean flow to the absorber column, $u_2$. The reboiler duty is applied to the plant, but the lean flow was only used locally by NMPC des to allow it to find the minimum reboiler duty. NMPC des's CVs were the CR in the desorber, $z_1 = CR_{des}$, and an unreachable max constraint on reboiler duty, $z_2 = u_1$. The structure of the controller can be seen in Figure 5.1. $u_1$ is the reboiler duty that is found by NMPC des and applied to the plant. $u_2$ is the lean flow. $u_2$ is first calculated by NMPC des, and the trajectory is sent as a reference to NMPC abs. NMPC abs also calculates $u_2$ with soft constraints based on the reference trajectory received from NMPC des. $\alpha_L^{ref}$ is the lean loading reference trajectory, which is sent from NMPC des to NMPC abs. The idea behind this split is that NMPC des can optimise the reboiler duty and steer the plant towards minimum reboiler duty while NMPC abs ensures that the CR in the absorber column is kept at its setpoint. Some code examples of how the models were implemented are given in Appendix D.

## 5.1 Expanding to multiple absorber columns

When adapting the control configuration to a plant with multiple absorbers, the first step was expanding the models to have multiple absorber columns. The *absorber model* was easily expanded to have multiple absorber columns as they are all independent, except for the fact that they have the same incoming lean loading, as there is still only one desorber. An MV and

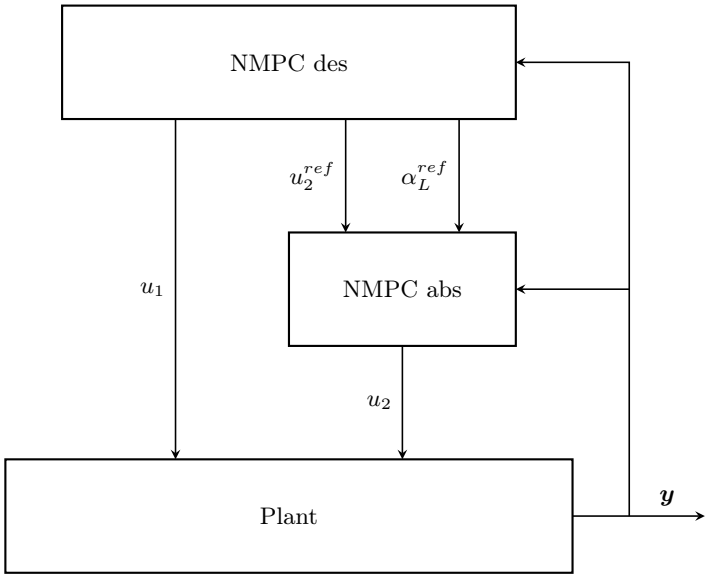



Figure 5.1: Control structure of the two-NMPC control configuration from the specialisation project[17]

a CV for the new column was added. It also has its own flue gas flow and composition inputs, which can be viewed as DVs. The CVs in NMPC abs are now $z_1 = CR_1$ and $z_2 = CR_2$. The MVs are the lean flows into the two columns, $u_2$ and $u_3$. The *desorber model* was also expanded to include linear step-response models for rich flow and rich loading from both columns. A new MV was added for the lean flow to the second absorber column. This new MV gives NMPC des one more degree of freedom (DOF). Hence a new CV had to be added. Without a new CV NMPC des does not have enough information to distribute the flow between the two absorber columns, as it is the desorber CR which is controlled in NMPC des. A new CV was added to handle this. This CV was the ratio between the two lean flows:

$$z_3 = \frac{u_2}{u_3}. \tag{5.1}$$

Its setpoint was the ratio between the amount of $CO_2$ that is supposed to be captured in the two columns

$$z_3^{sp} = \frac{CR_1^{sp} F_{1,in}^{\mathrm{CO_2}}}{CR_2^{sp} F_{2,in}^{\mathrm{CO_2}}}, \tag{5.2}$$

where $F_{1,in}^{\mathrm{CO_2}}$ is the $CO_2$ flow into absorber column 1 and $F_{2,in}^{\mathrm{CO_2}}$ is the $CO_2$ flow into absorber column 2. $CR_1^{sp}$ is the setpoint in absorber column 1 and $CR_2^{sp}$ is the setpoint in absorber column 2. The CVs in NMPC des are now $z_1 = CR_{des}$, $z_2 = u_1$ and $z_3$, while the MVs are $u_1$, $u_2$ and $u_3$. The communication between the two NMPCs was updated such that NMPC des sends the sum of its lean flows to NMPC abs. NMPC des gave no limitations on the distribution of the flows in NMPC abs, just the total amount. When there is just one absorber column, the setpoint for the CR in the desorber column is the same as in the

absorber column. When there are multiple absorber columns the CR setpoint and the amount of $CO_2$ entering the different absorber columns are not necessarily the same. The setpoint for the desorber column CR, $CR_{des}^{sp}$, can then be calculated based on information from the absorbers

$$CR_{des}^{SP} = \frac{CR_1^{sp} F_{1,in}^{CO_2} + CR_2^{sp} F_{2,in}^{CO_2}}{F_{1,in}^{CO_2} + F_{2,in}^{CO_2}}. \tag{5.3}$$

Section 7.1.2 shows the results of a controller test where the setpoint for CR in absorber 1 is increased from 84 % to 90 % after 60 minutes. The flue gas flow into absorber 2 is increased after 240 minutes, and the whole test is 720 minutes long. The correlation between the plant and the NMPCs is quite good initially, but there starts to be more deviation as the simulation progresses. It was believed that the deviation between controller and plant could be due to modelling error. Hence a new simple absorber model was implemented.

## 5.2 Using absorber models based on mass transfer

The step response models for rich loading were replaced with the model in Section 4.3. This model is based on modelling the mass transfer of $CO_2$ from gas to liquid and integrating a system of ODEs from the bottom to the top of the column. This becomes a BVP as we know the properties of the flue gas entering the bottom of the column and the lean amine entering the top of the column. The BVP is solved using a shooting method where it is converted to a IVP, and a guess for the unknown initial condition is updated until the boundary condition is satisfied. In this case, the unknown initial condition is the mole fraction of $CO_2$ in the liquid.

At first, it was attempted to use the secant method for updating the initial condition guess. Using $s_i$ for the $i$'th guess and $\phi(s_i)$ for the residual function the method is defined by the recurrence relation[26]

$$s_n = s_{n-1} - \phi(s_{n-1}) \frac{s_{n-1} - s_{n-2}}{\phi(s_{n-1}) - \phi(s_{n-2})}. \tag{5.4}$$

This method was not good enough. It needs two initial guesses, and it is too reliant on those guesses being close to the root. It was then attempted to use Brent's method, which is a combination of the bisection method, the secant method and inverse quadratic interpolation[27]. This method was better than the secant method, but would still occasionally fail. Due to this, it was decided to implement the KINSOL solver from SUNDIALS[1]. This solver proved to be more successful. Additionally, it only needs one initial guess. The model performed worse than the step-response model. The main problem was that the predicted rich loading was much higher than the actual value. It was attempted to fix this by adding temperature

[1] https://computing.llnl.gov/projects/sundials/kinsol

dependence. The mechanistic *absorber model* was run at typical conditions, and the temperature profile in the column was found. This profile was then used and interpolated in the mass transfer model. This did not help much, and there were still significant errors. It was decided that this model did not give good enough results to warrant any further investigation, as the point of the model is to be simple, and it was becoming increasingly complex. It is also a risk running a root solver, as it may fail or give the wrong root.

## 5.3 Using absorber models based on mole balances

It was attempted to use the model in Subsection 4.4.1 to predict rich flow and rich loading instead of the step-response models. The CVs and MVs are the same as in the previous configuration, only the models for predicting rich flow and rich loading are different. The results of the same test as before can be seen in Section 7.1.2. This seems to be an improvement compared to the step-response models initially, but there is still some deviation between the NMPCs and the plant at the end of the simulation. As explained in Subsection 4.4.1 the value of the CR in the model, $c$, can either be constant or come from somewhere as a DV. In this case, it is a DV, and the current value and a trajectory are received from NMPC abs. The communication now goes between the NMPCs in both directions. The control structure is given in Figure 5.2. $\boldsymbol{\theta}^{des}$ is a vector containing current values and trajectories of the sum of the lean flows ($u_2$ and $u_3$) and lean loading and is sent from NMPC des to NMPC abs. $\boldsymbol{\theta}^{abs}$ contains the current CR and a predicted trajectory for both columns.

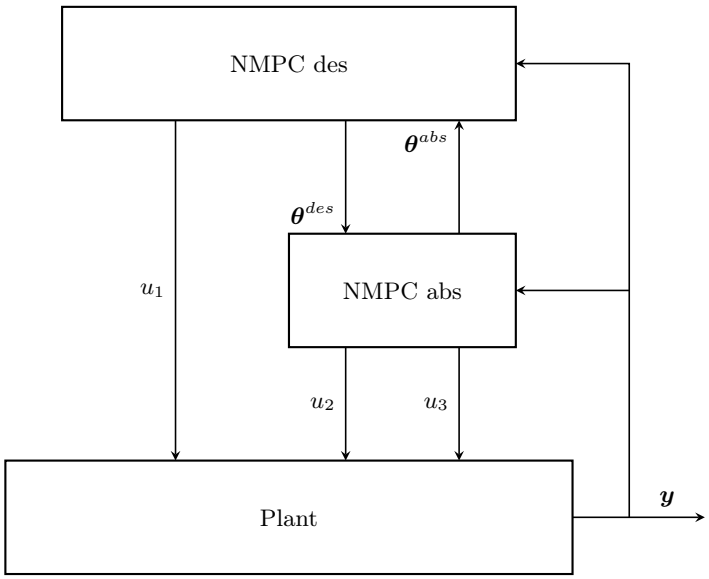



Figure 5.2: Control structure of the configuration with simple absorber models based on mole balances. $u_1$ is the reboiler duty while $u_2$ and $u_3$ are the lean flows into absorber column 1 and 2 respectively. $\theta^{des}$ is a vector with values and predicted trajectories sent from NMPC des to NMPC abs. $\theta^{abs}$ is a vector with values and predicted trajectories sent from NMPC abs to NMPC des.

## 5.4 Step-response plus absorber model based on mole balances

A new solution was proposed where the step response models are used to predict rich flow and rich loading, and the model in Subsection 4.4.2 is used to calculate the CR in the absorber columns. These CRs would now be the CVs in NMPC des. The benefits of this are that NMPC des now has a factual basis for distributing the flows, instead of a CV which does not necessarily have to be fulfilled at the optimal solution. Another benefit is the fact that we are no longer controlling the CR in the desorber, but rather the CRs in the absorber columns. At steady state these should be the same, but dynamically the CRs in the absorber columns should be controlled, as this is where $CO_2$ is emitted. It also allows for bias estimation of rich loading and CR, which helps ensure offset-free control.

## 5.5 Implementing bias updating

From the results in Section 7.1.2 it can be seen that the controllers performed well initially, but there is some deviation in the CR after some time. Looking at the plots of rich and lean loading, it can be seen that there is some deviation between the values predicted in NMPC des and the actual value measured from the plant. It was theorised that this could be the reason for the deviation in CR. The reboiler duty needed is strongly dependent on the rich loading, and lean loading strongly affects the amount of lean flow needed to capture a given amount of $CO_2$. Removing this model error could be the solution to removing the deviation in CR. It can be assumed that it is possible to measure the loading in the plants. There are correlations developed by Thor Mejdell at Sintef that can predict the loading based on density measurements. Hence, the assumption of having a measurement of loading available is acceptable, even though this is a measurement derived from the density measurement. It is also assumed that the CR can be derived based on flow and composition measurements from the plant, so a measurement of the CR is also available.

### 5.5.1 Updating the step response models

The step response models are used to predict some properties of the flow coming from the absorbers. There will be some model errors when using step response models. This error can be removed by using bias updating of the models as shown in Section 2.5. Adding the bias variable, $\beta_x$, to Equation 4.2 and using $x = \alpha_R$ gives

$$\dot{\alpha}_R = \frac{1}{\tau}\left(K(u - u_0) - (\alpha_R - (\alpha_{R,0} + \beta_{\alpha_R}))\right). \tag{5.5}$$

$\alpha_R$ is rich loading, $\tau$ is the time constant, $K$ is the gain of the process, $\alpha_{R,0}$ is the nominal value of the rich loading and $u_0$ is the nominal value of the input. The input, in this case, is the lean flow into the corresponding column. As Equation 5.5 is a linear model, the bias variable could be placed anywhere in the equation as long as it is not multiplied with $u$ or $\alpha_R$. The bias variable was placed in the term $\alpha_{R,0} + \beta_{\alpha_R}$ as this will give it the same unit as

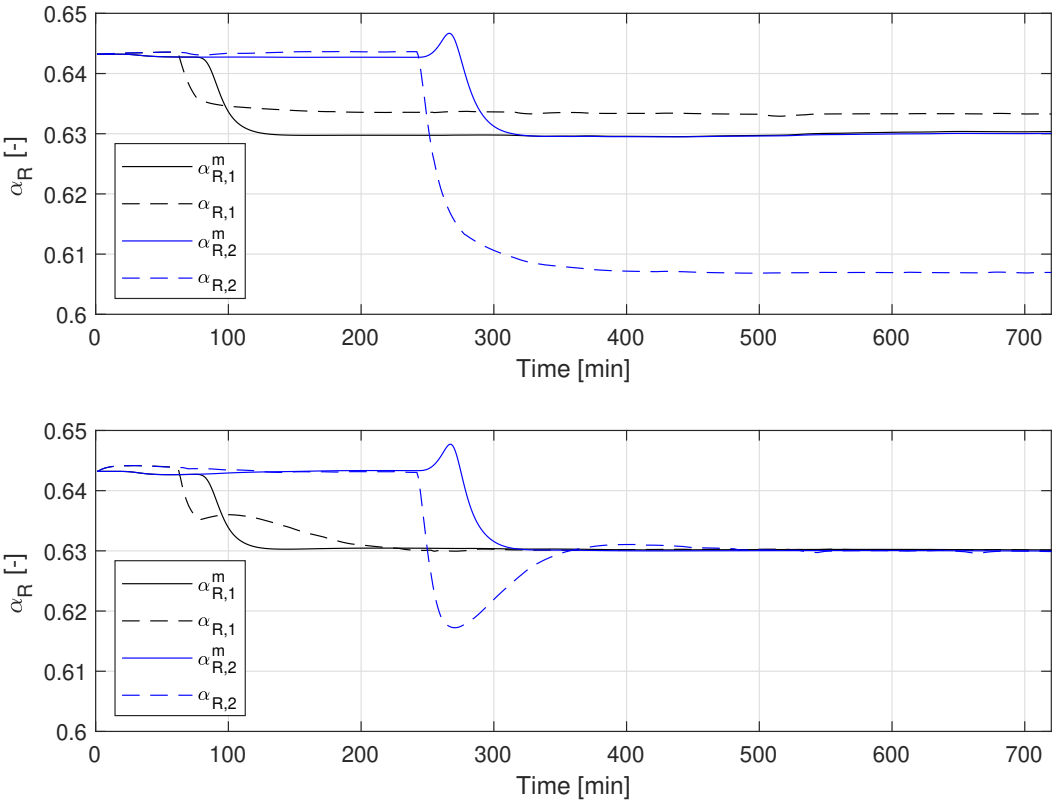



Figure 5.3: Measured vs predicted rich loading with and without bias updating of the model. The solid lines are measurements, while the stippled lines are from the model. The upper plot is without bias updating, while the lower plot is with bias updating. Here the subscripts 1 and 2 denote which absorber column the rich loading is in

$\alpha_R$. The bias variable can be viewed as an update of the parameter $\alpha_{R,0}$. The bias value is updated based on the difference between the rich loading measured from the plant, $\alpha^m_R$, and the rich loading predicted by the model, $\alpha_R$. This gives integral action and will remove the offset between measurements and predictions. The rule for updating the bias is then

$$\beta_{\alpha_R,k} = \beta_{\alpha_R,k-1} + K_{\beta_{\alpha_R}}(\alpha^m_{R,k} - \alpha_{R,k}) \tag{5.6}$$

Where $\beta_{\alpha_R,k}$ is the bias variable at time step $k$, $K_{\beta_{\alpha_R}}$ is the gain for bias updating, $\alpha^m_{R,k}$ is the measured rich loading at time step $k$ and $\alpha_{R,k}$ is the rich loading predicted by the model at time step $k$. Figure 5.3 shows a comparison between the rich loading with and without bias updating of the model in a test run of the NMPCs in a two absorber plant. $\alpha^m_{R,1}$ and $\alpha_{R,1}$ are measured and predicted rich loading in absorber column 1, and $\alpha^m_{R,2}$ and $\alpha_{R,2}$ are measured and predicted rich loading in absorber column 2. It can be seen that the bias helps remove the deviation in rich loading. However, this was not enough to remove the deviation in CR completely and the predicted lean loading was also still wrong. Figure 5.4 shows the values of the bias variables in the same test. $\beta_{\alpha_{R,1}}$ is the bias in absorber column 1 and $\beta_{\alpha_{R,2}}$ is the bias in absorber column 2.

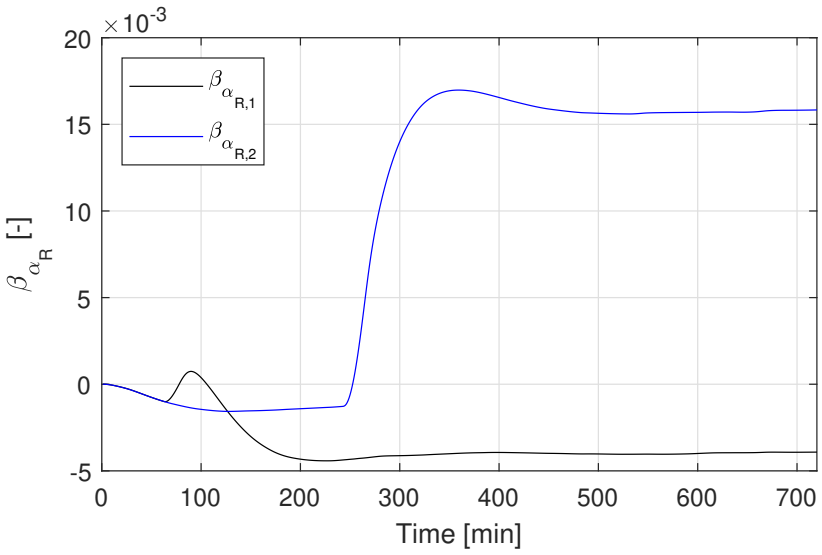



Figure 5.4: The values of the bias variables. The subscripts 1 and 2 denote which absorber column the bias relates to.

### 5.5.2 Updating the capture ratio

Even though the bias updating of the step-response models was enough to remove the deviation in rich loading, the CRs in the plant and in the controllers were still not the same. CRs are CVs which can be updated using bias variables as shown in Section 2.5. This was done to ensure offset free control. The bias was added both in NMPC abs and in NMPC des. The bias variable is updated similarly to the bias in the rich loading models,

$$\beta_{CR,k} = \beta_{CR,k-1} + K_{\beta_{CR}}(CR^m - CR), \tag{5.7}$$

where $\beta_{CR,k}$ is the bias variable at time step $k$, $K_{\beta_{CR}}$ is the gain of the bias updating, $CR^m$ is the measured CR and $CR$ is the CR predicted by the model. The bias is added directly on the CV which gives

$$CR = h(\boldsymbol{x_k}, \boldsymbol{u}_k) + \beta_{CR,k}. \tag{5.8}$$

This removed the offset between the CR in the plant and in the controller. The results of the same test as before using this control configuration can be seen in Section 7.1.2.

## 5.6 Individual NMPC for each absorber column

The final step in the development of the control configurations was splitting NMPC abs into an individual NMPC for each absorber column. With NMPC des now having the CRs in the absorber columns instead of the CR in the desorber column and a the lean flow ratio as CVs, more trust could be trust in the lean flows found by NMPC des. Instead of NMPC des sending the sum of $u_2$ and $u_3$ to NMPC abs, now $u_2$ is sent to the NMPC controlling

absorber column 1 and $u_3$ is sent to the NMPC controlling absorber column 2. The new NMPCs will be referred to as NMPS abs1 and NMPC abs2 after which absorber column they control. The control structure can be seen in Figure 5.5. Splitting NMPC abs this way is beneficial with regards to computational time due to two reasons. Firstly the computational time of one NMPC for all absorbers is larger than the sum of the computational times with one NMPC per absorber. Secondly, as the columns are independent, the NMPCs can be run simultaneously on different processor cores. In addition to this, it might be preferential to have separate NMPCs in the case where the absorber columns are owned and operated by different companies. The results of this control configuration to the same test as before is given in Subsection 7.1.3.

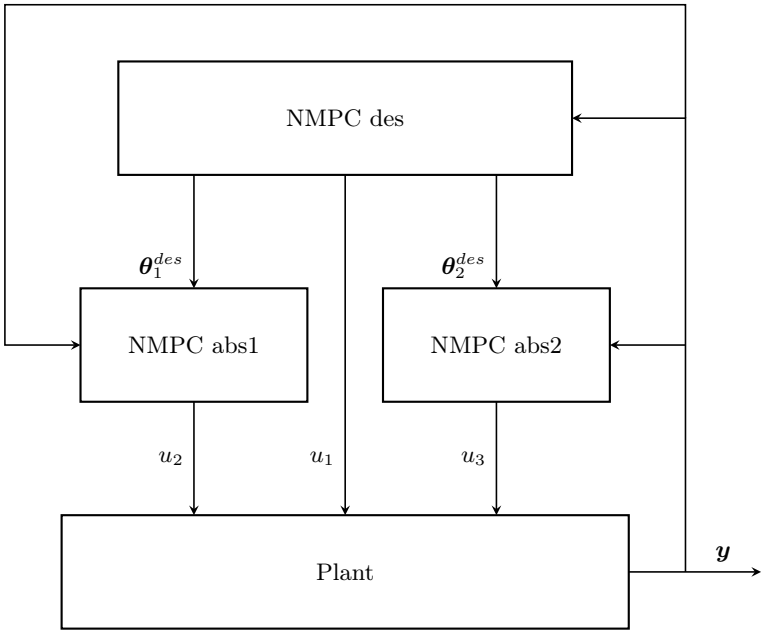



Figure 5.5: Control structure of the configuration with an individual NMPC for each absorber column. $u_1$ is the reboiler duty while $u_2$ and $u_3$ are the lean flows into absorber column 1 and 2 respectively. $\theta_1^{des}$ is a vector with values and predicted trajectories sent from NMPC des to NMPC abs1. $\theta_2^{des}$ is a vector with values and predicted trajectories sent from NMPC des to NMPC abs2.

Chapter 6

# Final control configurations

The final control configurations used the step-response models to predict the rich loadings and flows plus absorber models based on mole balances. It also included bias updating of the step response models for rich loading and updating of the predicted CRs in the absorber columns. There were two final configurations, the difference between them is that one had just one NMPC for all the absorber columns while the other had individual NMPCs for each absorber column. The configurations were tested and compared with each other. Finally, the control configurations will be compared with a one-NMPC configuration running the *original model* in the NMPC, which will serve as a benchmark. More details on the control configurations will be given in this chapter.

## 6.1 Final two-NMPC configuration

### 6.1.1 NMPC des

NMPC des runs the *desorber model* with step-response models for predicting rich loading and rich flow and uses the absorber model based on mole balances to calculate the CRs in the absorber columns. It also has bias updating of the step-response model for rich loading and the CRs. The gains for bias updating were set to 0.015 for all the parameters. This gain gave a quick response while also not causing any oscillations by overshooting, which was experienced at larger gain values. Its MVs are the reboiler duty, $u_1$ and its own local lean flows to each absorber column, $u_2$ and $u_3$. The CVs are the CRs in the absorber columns, $z_1 = CR_1$ and $z_2 = CR_2$. They both have a setpoint, a maximum constraint and a minimum constraint. The minimum constraint is equal to the setpoint and will make it more beneficial for NMPC des to capture a bit more $CO_2$ than required than a bit less. The last CV is the reboiler duty, $z_3 = u_1$. There is no setpoint on this CV, but it has an unreachable max constraint. The maximum constraint is set to 0, and violating this constraint is penalised. With the correct tuning, this will cause the controller to satisfy the setpoint for CR while attempting to minimise the reboiler duty. A summary of the CVs is given in Table 6.1.

Table 6.1: Controlled variables in NMPC des

| Symbol | Description | Setpoint | Min | Max | q | zSpan | $r_1$ |
|---|---|---|---|---|---|---|---|
| $z_1$ | Capture ratio 1 | $CR_1^{sp}$ | $CR_1^{sp}$ | 95 | 1.0 | 2.12 | 2.0 |
| $z_2$ | Capture ratio 2 | $CR_2^{sp}$ | $CR_2^{sp}$ | 95 | 1.0 | 2.12 | 2.0 |
| $z_3$ | Reboiler duty | None | 0 | 0 | 0 | 3.0 | 0.05 |

The code for specifying the parameterisation of the inputs is given in Code box 6.1. All the inputs have the same parameterisation. There are four blocks for the inputs with increasing size. The blocks start at 0, 15, 45 and 90 minutes into the prediction horizon. The fourth block ends at the end of the prediction horizon, which is 300 minutes long.

Code box 6.1: C Code for deciding the MV parameterisation in CENIT

```
/* Standard parameterisation of the inputs. There are four MV blocks,
the first from 0 to 15, second from 15 to 45, third from 45 to 90
and fourth from 90 to the end of the horizon (300) */
for (int i = 1; i <= NU; i++) {
    Upar->m[i][1] = 15;
    Upar->m[i][2] = 45;
    Upar->m[i][3] = 90;
    // Disable the rest of the blocks by setting them to 0
    for (int j = (NCOL_UPAR_MPC_MV+1); j <= NCOL_UPAR_MPC; j++)
        Upar->m[i][j] = 0;
}
```

The code for specifying the parameterisation of the CVs is given in Code box 6.2. The standard parameterisation of the CVs has the first evaluation point at 100 minutes, and the next ones are every fifth minute after until the end of the prediction horizon. On the other hand, the CRs does not have the standard parameterisation. For the CRs, the evaluation points start at 10 minutes. The next ones come every tenth minute until it catches up with the standard parameterisation at 190 minutes. It then follows the standard parameterisation until the end. Thus, in the optimisation problem in NMPC des, deviations from the setpoints and constraint violations are penalised quite early for the CRs. At the same time, the penalty for reboiler duty only comes into effect more towards the end of the horizon. Therefore, this tuning allows NMPC des to increase the reboiler duty initially if it allows for a reduced reboiler duty towards the end of the prediction horizon.

Code box 6.2: C Code for deciding the CV parameterisation in CENIT

```
/* The standard CV parameterisation. The first evaluation point
is 100 minutes in the horizon and the next are every fifth minute
after that until the end of the horizon */
for (int i = 1; i <= NZ; i++) {
    for (int j = 1; j <= NCOL_ZPAR_MPC; j++)
        Zpar->m[i][j] = 95 + 5*j;
}
```

```
/* The capture ratios have their own parameterisation.
First evaluation point is after 10 minutes and the next are every
tenth minute after that until it catches up and follows
the standard parameterisation */
for (int j = 1; j <= 18; j++) {
    Zpar->m[1 + i_z_CaptureRatio_1][j] = j * 10;
    Zpar->m[1 + i_z_CaptureRatio_2][j] = j * 10;
}
```

### 6.1.2 NMPC abs

NMPC abs runs the *absorber model* with a model for each column in the plant. It has bias updating on the CRs to ensure offset free control. The gains for the bias updating were 0.01 for both bias variables. The MVs are the lean flows into the two absorber columns, $u_2$ and $u_3$, which for NMPC abs are applied to the plant. The CVs are the CRs in the absorber columns, $z_1 = CR_1$ and $z_2 = CR_2$, and the sum of the lean flows, $z_3 = u_2 + u_3$. The setpoints and constraints for the CRs are the same as in NMPC des. The third CV does not have a setpoint, only constraints. It receives the current value and trajectories for the lean flows from NMPC des, $F_{des}$. These values are used to find the constraints for $z_3$. The minimum is $F_{des} - 0.1\,\mathrm{kg\,min^{-1}}$ and the maximum constraint is $F_{des} + 0.1\,\mathrm{kg\,min^{-1}}$. A summary of the CVs is given in Table 6.2.

Table 6.2: Controlled variables in NMPC abs

| Symbol | Description | Setpoint | Min | Max | q | zSpan | $r_1$ |
|---|---|---|---|---|---|---|---|
| $z_1$ | Capture ratio 1 | $CR_1^{sp}$ | $CR_1^{sp}$ | 95 | 1.0 | 2.12 | 0.3 |
| $z_2$ | Capture ratio 2 | $CR_2^{sp}$ | $CR_2^{sp}$ | 95 | 1.0 | 2.12 | 0.3 |
| $z_3$ | Sum of lean flows | None | $F_{des} - 0.1$ | $F_{des} + 0.1$ | 0 | 1.0 | 5.0 |

Code box 6.3 shows the C code used to parameterise the inputs in CENIT. The standard input parameterisation is the same as in NMPC des. The prediction horizon is also the same. All inputs except `U_LeanLoading_in`, which is the lean loading entering the column, have the standard parameterisation. For this input, the fourth block goes from 90 to 299 minutes. Then there is a fifth from 299 to the end of the horizon. The reason for this is that `U_LeanLoading_in` is a DV which receives its predicted trajectory from NMPC des. If it has the same parameterisation as the other inputs, the trajectory will cut off at 90 minutes and keep this value until the prediction horizon's end. By adding this extra block, the whole horizon can be used.

Code box 6.3: C Code for deciding the MV parameterisation in CENIT

```
/* Standard parameterisation of the inputs. There are four MV blocks,
the first from 0 to 15, second from 15 to 45, third from 45 to 90
and fourth from 90 to the end of the horizon (300) */
for (int i = 1; i <= NU; i++) {
    Upar->m[i][1] = 15;
```

```
    Upar->m[i][2] = 45;
    Upar->m[i][3] = 90;
    // Disable the rest of the blocks by setting them to 0
    for (int j = (NCOL_UPAR_MPC_MV+1); j <= NCOL_UPAR_MPC; j++)
        Upar->m[i][j] = 0;
}
/* Add an extra block for the lean loading that goes to the last step
before the end of the prediction horizon */
Upar->m[1 + i_U_LeanLoading_in][NCOL_UPAR_MPC_MV + 1] = 299;
```

Code box 6.4 shows the C code for defining the parameterisation of the CVs. The standard parameterisation starts with an evaluation point at 100 minutes, and the next ones are every fifth minute after that until the end of the horizon. The capture ratios have their own parameterisation with the first evaluation point after 10 minutes. The next ones are every tenth minute after that until it catches up with the standard parameterisation at 190 minutes, which it then follows until the end. The sum of the lean flows is also a CV which has its own limits. The first evaluation point is after 70 minutes. The next ones are every tenth minute after that until 130 minutes, where it catches up with the standard parameterisation and follows it until the end of the prediction horizon. This allows NMPC abs to choose the values of the lean flows freely until 70 minutes, where the constraints begin to be enforced. NMPC abs is allowed to breach the constraint initially but should aim to satisfy them towards the end of the prediction horizon. The combination of this with the parameterisation of the CRs means that for the first 70 minutes, the only objective of NMPC abs is to have the CRs at their setpoint. After 70 minutes, it should try to keep the CRs at their setpoint while also satisfying the constraints on the sum of the lean flows.

Code box 6.4: C Code for deciding the CV parameterisation in CENIT

```
/* The standard CV parameterisation. The first evaluation point
is 100 minutes in the horizon and the next are every fifth minute
after that until the end of the horizon */
for (int i = 1; i <= NZ; i++) {
    for (int j = 1; j <= NCOL_ZPAR_MPC; j++)
        Zpar->m[i][j] = 95 + 5*j;
}
/* The capture ratios have their own parameterisation. First evaluation
point is after 10 minutes and the next are every tenth minute after that
until it catches up and follows the standard parameterisation */
for (int j = 1; j <= 18; j++) {
    Zpar->m[1 + i_z_CaptureRatio_1][j] = j * 10;
    Zpar->m[1 + i_z_CaptureRatio_2][j] = j * 10;
}
/* The sum of the lean flows has its own parameterisation. The first evaluation
point is after 70 minutes, and the next ones are every tenth minute after that
until 130 minutes where it catches and follows the standard parameterisation */
for (int j = 1; j <= 7; j++) {
    Zpar->m[1 + i_z_Fl_Lean_Abs_Mass][j] = 60 + j * 10;
}
```

## 6.2 Final multi-NMPC configuration

As explained in Section 5.6 NMPC abs was split into an individual NMPC for each absorber column. The tuning of NMPC des was identical, the only change is that instead of sending the sum of the lean flows to NMPC abs it sends the trajectory of $u_2$ to NMPC abs1 and the trajectory of $u_3$ to NMPC abs2. The CVs are now $z_1 = CR_1$ and $z_2 = u_2^{abs1}$ in NMPC abs1, and $z_1 = CR_2$ and $z_2 = u_3^{abs2}$ in NMPC abs2. The tuning of the two new NMPCs is also similar to when there was only one, the difference is in the constraints on $z_2$. These constraints are now $u_2^{des} \pm 0.05\,\mathrm{kg\,min^{-1}}$ in NMPC abs1 and $u_3^{des} \pm 0.05\,\mathrm{kg\,min^{-1}}$ in NMPC abs2, where $u_2^{des}$ and $u_3^{des}$ are the values of $u_2$ and $u_3$ calculated by NMPC des.

## 6.3 The benchmark configuration with one NMPC

The benchmark control configuration had a similar tuning to the two-NMPC control configuration. The benchmark control configuration uses the *original model*, same as the one used as plant replacement in RealSim. The MVs are the reboiler duty, $u_1$ and the lean flows into the absorber columns, $u_2$ and $u_3$. Its CVs are the CRs in the absorber columns, $z_1 = CR_1$ and $z_2 = CR_2$, and the reboiler duty, $z_3 = u_1$. $z_1$ and $z_2$ have both setpoints and constraints, and the minimum constraint is equal to the setpoint. $z_3$ does not have a setpoint, only an unreachable maximum constraint, same as in NMPC des. A summary is given in Table 6.3. The parameterisation of the inputs and CVs is identical to the parameterisation in NMPC des. The C code in Code box 6.1 and Code box 6.2 is also present in the code for the benchmark configuration.

Table 6.3: Controlled variables in the benchmark configuration

| Symbol | Description | Setpoint | Min | Max | q | zSpan | $r_1$ |
|---|---|---|---|---|---|---|---|
| $z_1$ | Capture ratio 1 | $CR_1^{sp}$ | $CR_1^{sp}$ | 100 | 1.0 | 2.12 | 0.3 |
| $z_2$ | Capture ratio 2 | $CR_2^{sp}$ | $CR_2^{sp}$ | 100 | 1.0 | 2.12 | 0.3 |
| $z_3$ | Reboiler duty | None | 0 | 0 | 0 | 1.0 | 0.05 |

Chapter 7

# Results

In this chapter the results for tests of different control configurations are given. One test was performed on a $CO_2$ capture plant with two absorber columns and one test was performed on a $CO_2$ capture plant with three absorber columns. In this chapter the configurations are numbered 1 through 5 and these numbers will be used when referring to them later in the chapter. Table 7.1 gives an overview of the different NMPC configurations with their number, a description of the control configuration and which section it is possible to read more about the configuration.

Table 7.1: Overview of the tested NMPC configurations

| Configuration | Description | Section |
|---|---|---|
| 1 | One NMPC using the same model as the plant | Section 6.3 |
| 2 | Two NMPCs, NMPC des and NMPC abs. NMPC des has step response models for predicting rich flow and loading | Section 5.1 |
| 3 | Same as 2 NMPC step, but NMPC des uses mole balance models for predicting rich flow and loading | Section 5.3 |
| 4 | Same as 2 NMPC step, but mole balance models are used to calculate absorber CRs in NMPC des. These CRs are used as CVs instead of the CR in the desorber. Also uses bias updating of the step response models for rich loading and of the absorber CRs in NMPC abs and NMPC des | Section 6.1 |
| 5 | Same as 2 NMPC bias, but NMPC abs is split into individual NMPCs for each absorber column | Section 6.2 |

## 7.1 $CO_2$ capture plant with two absorber columns

A test was run to compare the performance between the developed two-NMPC configurations with the benchmark. In the test the process starts at the optimal point and after 60 minutes the CR setpoint in absorber column 1, $CR_1^{sp}$, is increased from 84 % to 90 %. After 240

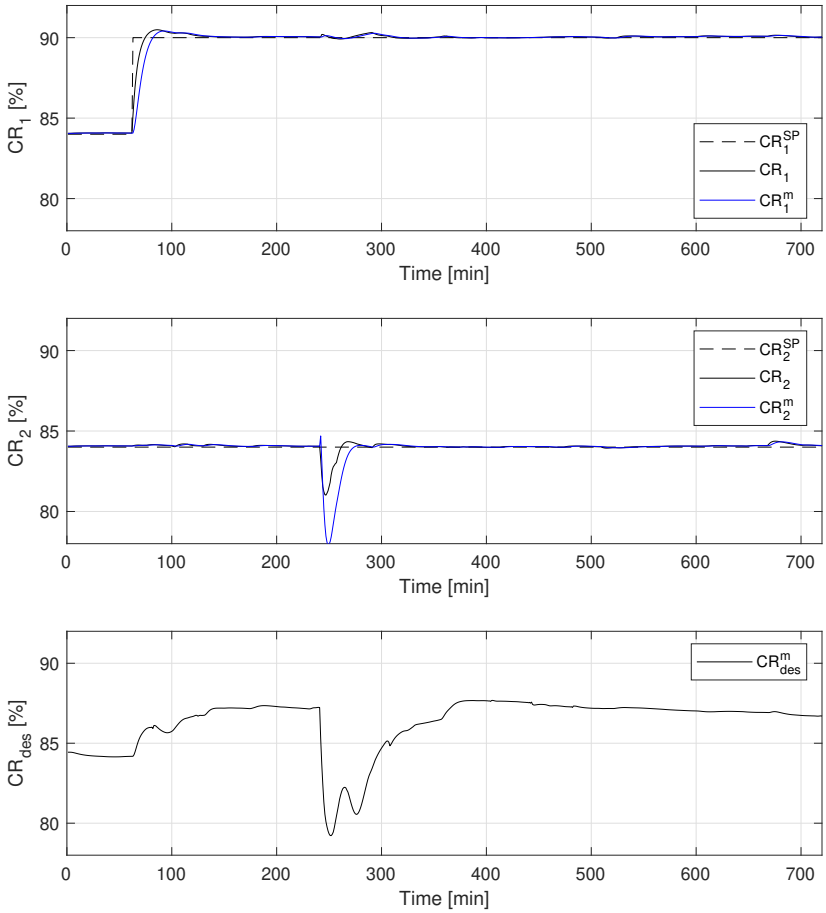



Figure 7.1: The CRs when using the control configuration 1. The upper plot shows the CR in absorber column 1 and the middle plot shows the CRs in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted CRs, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$. The lower plot shows the CR from the desorber column measured from the plant. It is not a CV in this NMPC configuration.

minutes the flue gas flow into absorber column 2 is increased from $80\,\mathrm{m^3\,h^{-1}}$ to $100\,\mathrm{m^3\,h^{-1}}$. No additional changes were made and the whole test was 720 minutes long. Some of the two-NMPC configurations developed in this thesis and the benchmark control configuration were tested.

### 7.1.1 Benchmark control configuration

First the benchmark control configuration was tested, number 1 in Table 7.1. The CV setpoint tracking performance can be seen in Figure 7.1. The upper plot shows the CR in absorber column 1 while the middle plot shows the CR in absorber column 2. The plots show the setpoints, measured values from the plant in RealSim and predicted values from the NMPC in CENIT. The lower plot shows the CR measured in the desorber column. The response to the setpoint change is fast and the increased flue flow is quickly accounted for.

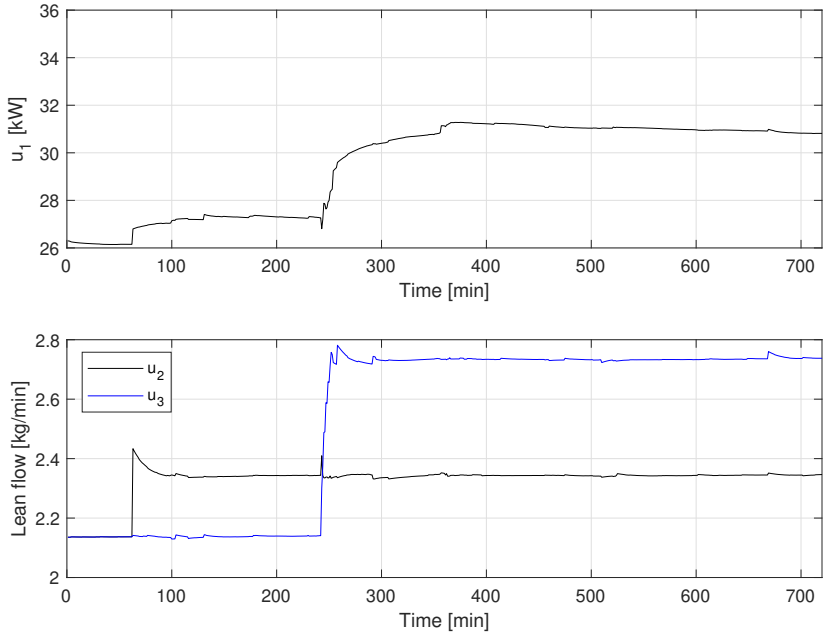



Figure 7.2: The MVs when using the control configuration 1. The upper plot shows the reboiler duty, $u_1$. The lower plot shows the lean flows, $u_2$ and $u_3$.

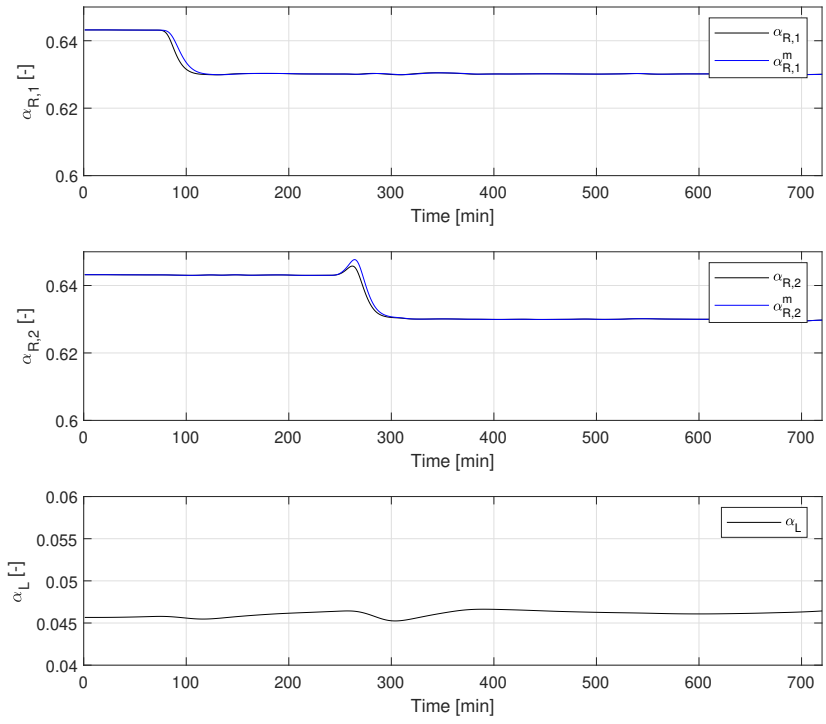



Figure 7.3: Lean and rich loading when using control configuration 1. The upper plot shows the rich loading in the outlet flows from absorber column 1. $\alpha_{R,1}$ is the value predicted by the NMPC and $\alpha^m_{R,1}$ is the measurement from the plant. The middle plot shows the rich loading in the outlet flows from absorber column 2, $\alpha_{R,2}$ is the value predicted by the NMPC and $\alpha^m_{R,2}$ is the measurement from the plant. The lower plot shows the lean loading in the lean buffer tank, $\alpha_L$.

Figure 7.2 shows the inputs from the benchmark controller. The upper plot shows the reboiler duty while the lower plot shows the lean flows. The reboiler duty is increased rather slowly and progressively, while the lean flows are increased fast with only small corrections afterwards. The loadings in the plant can be seen in Figure 7.3. The upper plot is the rich loading in the outlet flow from absorber column 1, the middle plot is the rich loading in the outlet from absorber column 2 while the lower plot shows the lean loading in the lean buffer tank. There is almost no difference between the measured and predicted values.

### 7.1.2 Two-NMPC control configurations

In Chapter 5 several iterations of the two-NMPC control configuration were developed. The test results of some of these are given here.

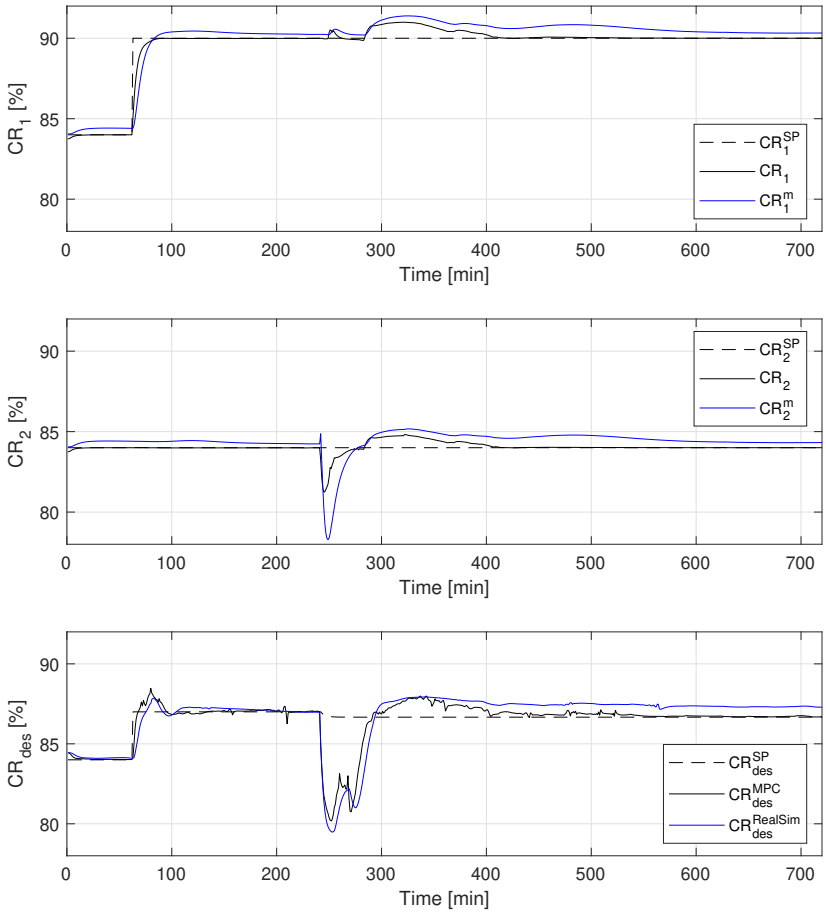



Figure 7.4: The CRs when using control configuration 2. The upper plot shows the CR in absorber column 1 and the middle plot shows the CRs in absorber column 2. These are the CVs in NMPC abs. The bottom plot shows the CR in the desorber column, which is the CV in NMPC des. The stippled black lines are the setpoints, $CR_1^{sp}$, $CR_2^{sp}$ and $CR_{des}^{sp}$. The solid black lines are the predicted values, $CR_1$ and $CR_2$ from NMPC abs, and $CR_{des}$ from NMPC des. The blue lines are the measured CRs from the plant, $CR_1^m$, $CR_2^m$ and $CR_{des}^m$.

### Two-NMPC control configuration with step response models

The CV setpoint tracking performance when using the two-NMPC configuration with step response models from Section 5.1, number 2 in Table 7.1, can be seen in Figure 7.4. The upper plot shows the CR in absorber column 1 while the middle plot shows the CR in absorber column 2. The bottom plot shows the CR in the desorber column. The response to the setpoint change is a little slower here and it also uses longer time to account for the flue gas increase. From around 300 minutes it also catches more $CO_2$ than necessary. There is some deviation between measurements and predictions, the measured CRs are consistently higher for all CRs.

The MVs can be seen in Figure 7.5. The upper plot shows the reboiler duty calculated by NMPC des and the lower plot shows the two lean flows calculated by NMPC abs. The lean flows look pretty similar here to the previous configuration. The reboiler duty on the other hand is used much more aggressively in this configuration.

A comparison between the lean flows found by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$, and the ones found by NMPC des, $u_2^{des}$ and $u_3^{des}$, is given in the upper plot in Figure 7.6. They are similar, but the ones calculated by NMPC des are a lot more up and down than the ones calculated by NMPC abs and generally a bit larger. The lower plot shows the CV, $z_3$, in NMPC abs and the limits which it receives from NMPC des. It mostly stays within the limits, but breaches them right after the disturbances.

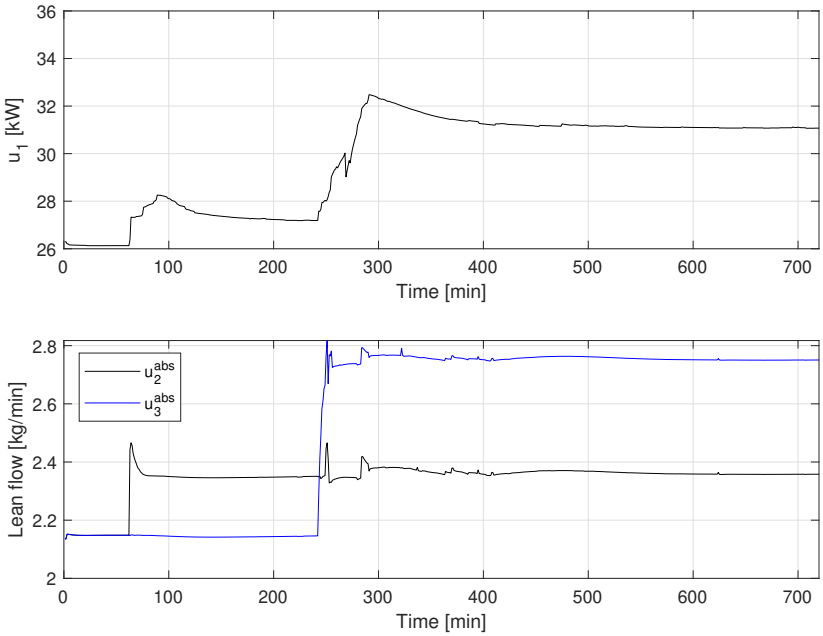



Figure 7.5: The MVs when using control configuration 2. The upper plot shows the reboiler duty, $u_1$. The lower plot shows the lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$.

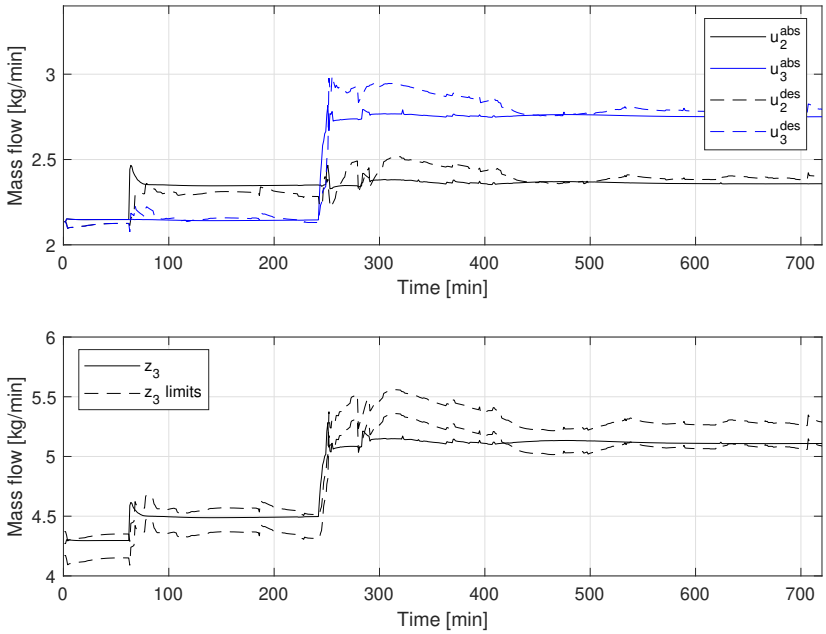



Figure 7.6: Comparison between the MVs calculated by NMPC abs and NMPC des when using control configuration 2. The upper plot shows the MVs, the solid lines are calculated by NMPC abs and the stippled lines are calculated by NMPC des. The lower plot shows $z_3$ in NMPC abs, the stippled lines are the limits based on the trajectory from NMPC des.

Figure 7.8 shows the loadings predicted by NMPC des and the ones measured from the plant. The upper plot shows the rich loading in the outlet flow coming from absorber column 1, the middle plot shows the rich loading in the outlet flow coming from absorber column 2. The lower plot shows the lean loading in the lean buffer tank. There is some deviation between the measured and predicted loading in absorber 1, but it is not that big. In absorber 2 on the other hand the difference is huge with the predicted rich loading being too small. The lean loading is pretty close apart from between 400 and 600 minutes where it deviates for a while.

Figure 7.7 shows the value of $z_3$ and its setpoint. The setpoint is followed closely. There is some deviation from the setpoint right when the disturbances occur, but not much.

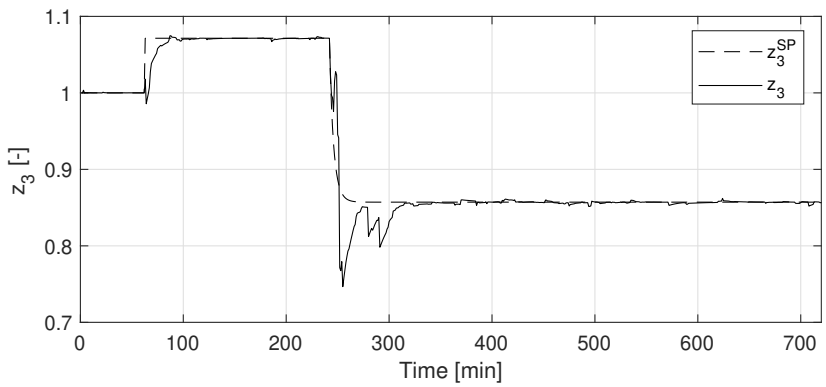



Figure 7.7: The CV $z_3$ and its setpoint in NMPC des when using control configuration 2

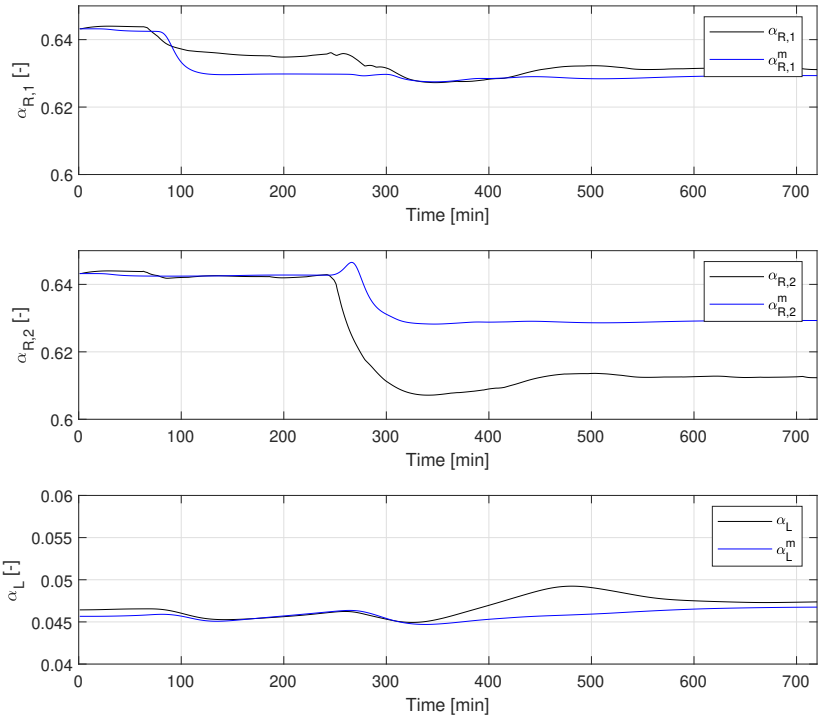



Figure 7.8: Lean and rich loading when using control configuration 2. The upper plot shows the rich loading in the outlet flows from absorber column 1. $\alpha_{R,1}$ is the value predicted by NMPC des and $\alpha^m_{R,1}$ is the measurement from the plant. The middle plot shows the rich loading in the outlet flows from absorber column 2, $\alpha_{R,2}$ is the value predicted by NMPC des and $\alpha^m_{R,2}$ is the measurement from the plant. The lower plot shows the lean loading in the lean buffer tank. $\alpha_L$ is the value predicted by NMPC des and $\alpha^m_L$ is the measurement from the plant.

### Two-NMPC control configuration with mole balance models

The CV setpoint tracking performance when using the two-NMPC configuration with mole balance models from Section 5.3, number 3 in Table 7.1, can be seen in Figure 7.9. The upper plot shows the CR in absorber column 1 while the middle plot shows the CR in absorber column 2. The bottom plot shows the CR in the desorber column. The setpoints are followed closely and the disturbances are rejected quickly. After about 500 minutes the measured CRs begin to drop and deviate from the prediction.

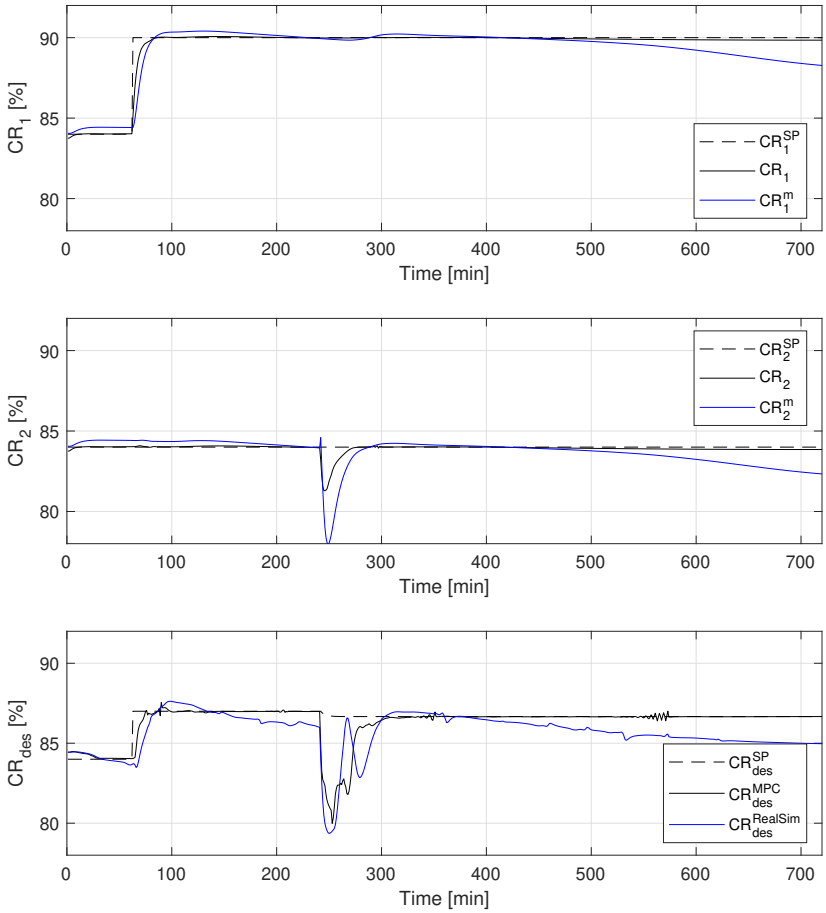



Figure 7.9: The CRs when using control configuration 3. The upper plot shows the CR in absorber column 1 and the middle plot shows the CRs in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted values in NMPC abs, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$. These are the CVs in NMPC abs. The bottom plot shows the CR in the desorber column, which is a CV in NMPC des.

The MVs can be seen in Figure 7.10. The upper plot shows the reboiler duty, $u_1$, while the lower plot shows the two lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$. Once again the lean flows are increased fast and remain stable after the initial period with quick change. The reboiler is used quite aggressively here as well, but it is reduced in the horizon.

A comparison between the lean flows found by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$, and the ones found by NMPC des, $u_2^{des}$ and $u_3^{des}$, is given in the upper plot in Figure 7.11. The lower plot shows the CV, $z_3$, in NMPC abs and the limits which it receives from NMPC des. In this configuration NMPC des consistently calculates lower lean flows than NMPC abs. Looking at the constraints on $z_3$ they seem to be violated almost the entire time.

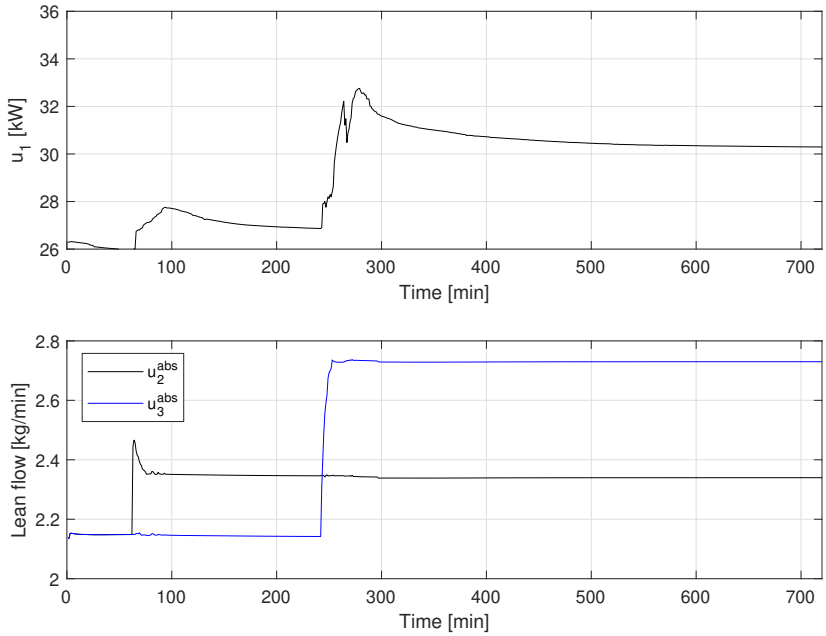



Figure 7.10: The MVs when using control configuration 3. The upper plot shows the reboiler duty, $u_1$. The lower plot shows the lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$.

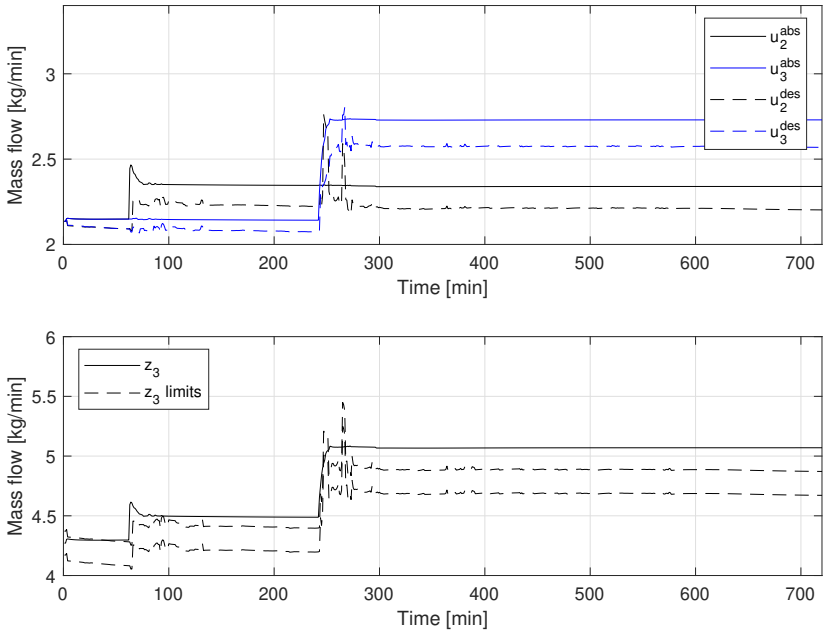



Figure 7.11: Comparison between the MVs calculated by NMPC abs and NMPC des when using control configuration 3. The upper plot shows the MVs, the solid lines are calculated by NMPC abs and the stippled lines are calculated by NMPC des. The lower plot shows $z_3$ in NMPC abs, the stippled lines are the limits.

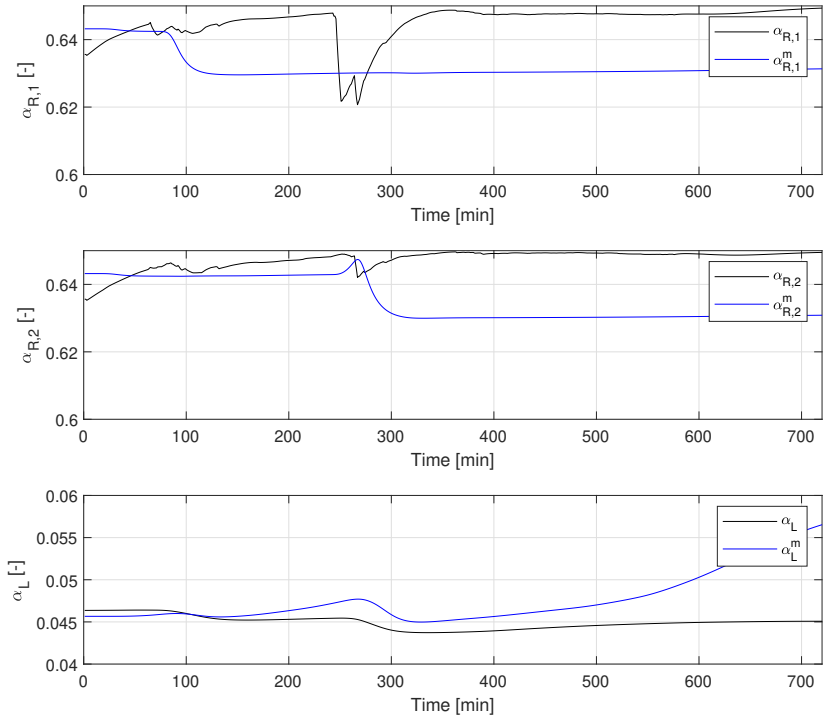



Figure 7.12: Lean and rich loading when using control configuration 3. The upper plot shows the rich loading in the outlet flows from absorber column 1. $\alpha_{R,1}$ is the value predicted by NMPC des and $\alpha^m_{R,1}$ is the measurement from the plant. The middle plot shows the rich loading in the outlet flows from absorber column 2, $\alpha_{R,2}$ is the value predicted by NMPC des and $\alpha^m_{R,2}$ is the measurement from the plant. The lower plot shows the lean loading in the lean buffer tank, $\alpha_L$ is the value predicted by NMPC des and $\alpha^m_L$ is the measurement from the plant.

Figure 7.12 shows the loadings predicted by NMPC des and the ones measured from the plant. The upper plot shows the rich loading in the outlet flow coming from absorber column 1, the middle plot shows the rich loading in the outlet flow coming from absorber column 2. The lower plot shows the lean loading in the lean buffer tank. Here the predicted rich loading is too high in both absorber columns. The lean loading is close up to around 300 minutes where it starts to increase until the end of the simulation, where the difference is quite large. Figure 7.13 shows the CV $z_3$ in NMPC des with its setpoint, which is followed closely.

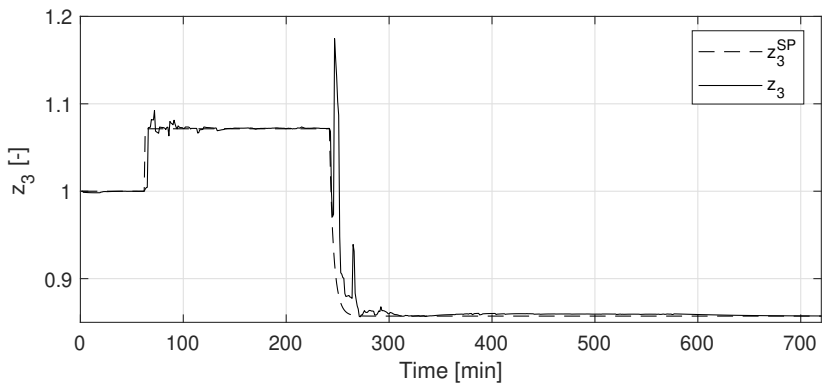



Figure 7.13: The CV $z_3$ and its setpoint in NMPC des when using control configuration 3

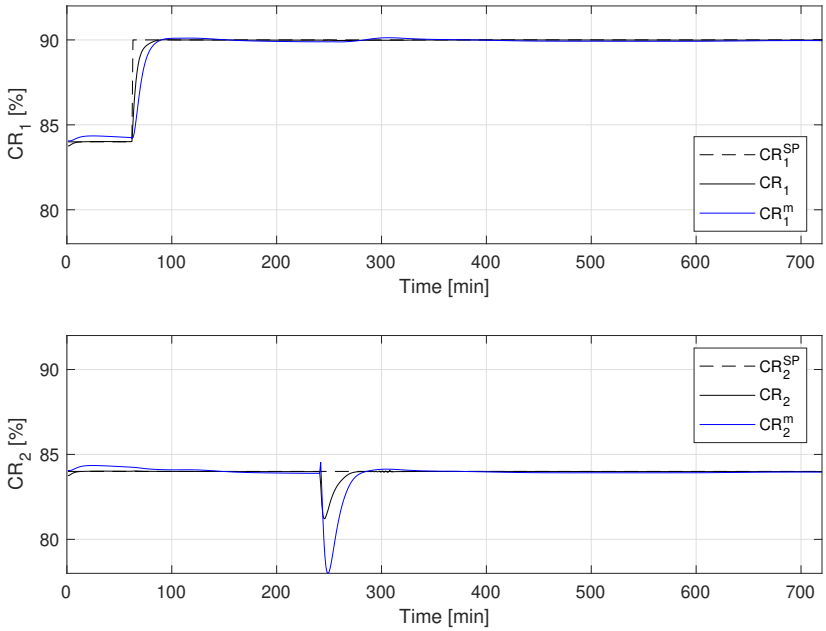



Figure 7.14: The CRs in NMPC abs when using control configuration 4. The upper plot shows the CR in absorber column 1 and the lower plot shows the CRs in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted values in NMPC abs, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$.

### Two-NMPC control configuration with bias estimation

The CV setpoint tracking performance when using the two-NMPC configuration from Section 6.1, number 4 in Table 7.1, can be seen in Figure 7.14. The upper plot shows the CR

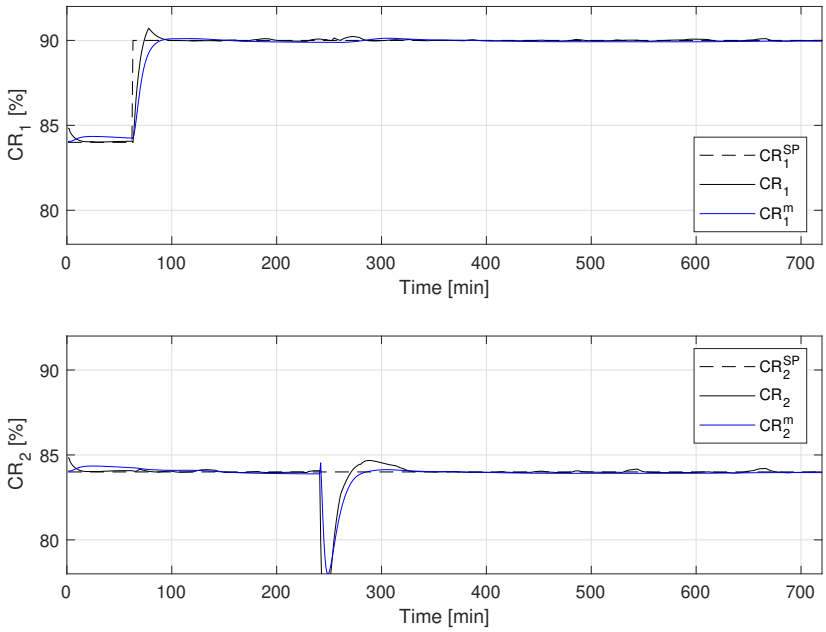



Figure 7.15: The CRs in NMPC des when using control configuration 4. The upper plot shows the CR in absorber column 1 and the lower plot shows the CRs in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted values in NMPC des, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$.

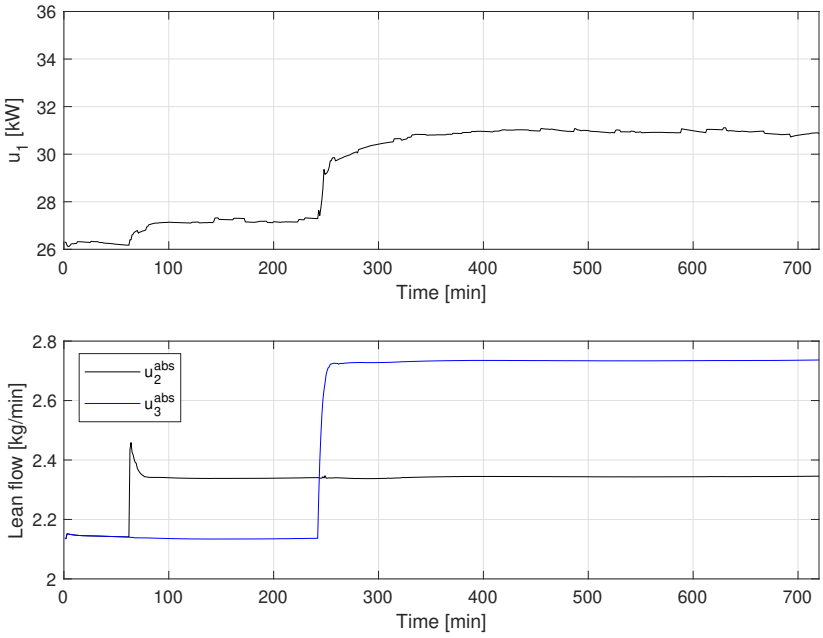



Figure 7.16: The MVs when using control configuration 4. The upper plot shows the reboiler duty, $u_1$. The lower plot shows the lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$.

in absorber column 1 while the lower plot shows the CR in absorber column 2. The setpoint is followed closely and the flue gas flow disturbance is accounted for quickly. There is very little deviation between prediction and measurement. Figure 7.14 is from NMPC abs while Figure 7.15 is the same plot from NMPC des. The CRs in the absorber columns are now the CVs in NMPC des instead of the desorber CR and the lean flow ratio. Figure 7.15 also shows that the setpoint is followed closely, the disturbance is rejected and there is little deviation between measurement and prediction.

The MVs can be seen in Figure 7.16. The upper plot shows the reboiler duty, $u_1$, while the lower plot shows the two lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$. The lean flows are similar to the rest of the two-NMPC configurations, but the reboiler duty strongly resembles the reboiler duty in configuration 1. The reboiler duty use is much less aggressive here. Instead it is increased slower towards its final value.

A comparison between the lean flows found by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$, and the ones found by NMPC des, $u_2^{des}$ and $u_3^{des}$, is given in the upper plot in Figure 7.17. The lower plot shows the CV, $z_3$, in NMPC abs and the limits which it receives from NMPC des. The lean flows found by NMPC des are much more aggressive and make a big jump before being quickly reduced again. $z_3$ is mostly within its constraints.

Figure 7.18 shows the loadings predicted by NMPC des and the ones measured from the plant. The upper plot shows the rich loading in the outlet flow coming from absorber column 1, the middle plot shows the rich loading in the outlet flow coming from absorber column 2. The lower plot shows the lean loading in the lean buffer tank.

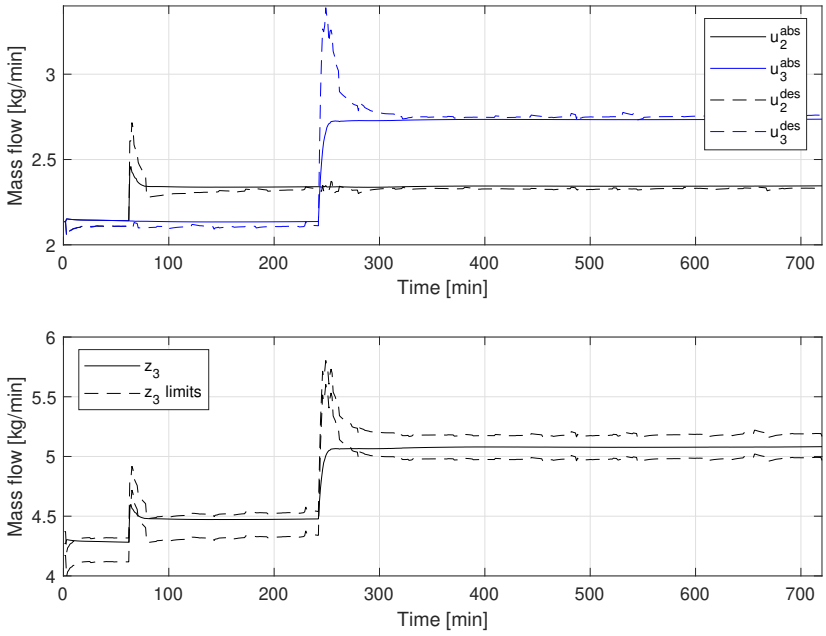



Figure 7.17: Comparison between the MVs calculated by NMPC abs and NMPC des when control configuration 4. The upper plot shows the MVs, the solid lines are calculated by NMPC abs and the stippled lines are calculated by NMPC des. The lower plot shows $z_3$ in NMPC abs, the stippled lines are the limits.

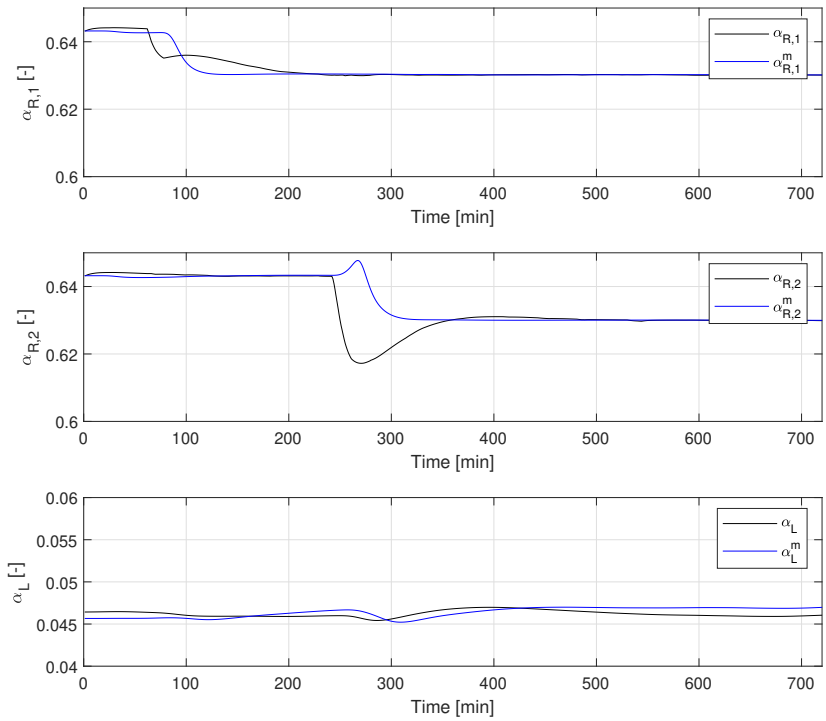



Figure 7.18: Lean and rich loading when using control configuration 4. The upper plot shows the rich loading in the outlet flows from absorber column 1. $\alpha_{R,1}$ is the value predicted by NMPC des and $\alpha^m_{R,1}$ is the measurement from the plant. The middle plot shows the rich loading in the outlet flows from absorber column 2, $\alpha_{R,2}$ is the value predicted by NMPC des and $\alpha^m_{R,2}$ is the measurement from the plant. The lower plot shows the lean loading in the lean buffer tank, $\alpha_L$ is the value predicted by NMPC des and $\alpha^m_L$ is the measurement from the plant.

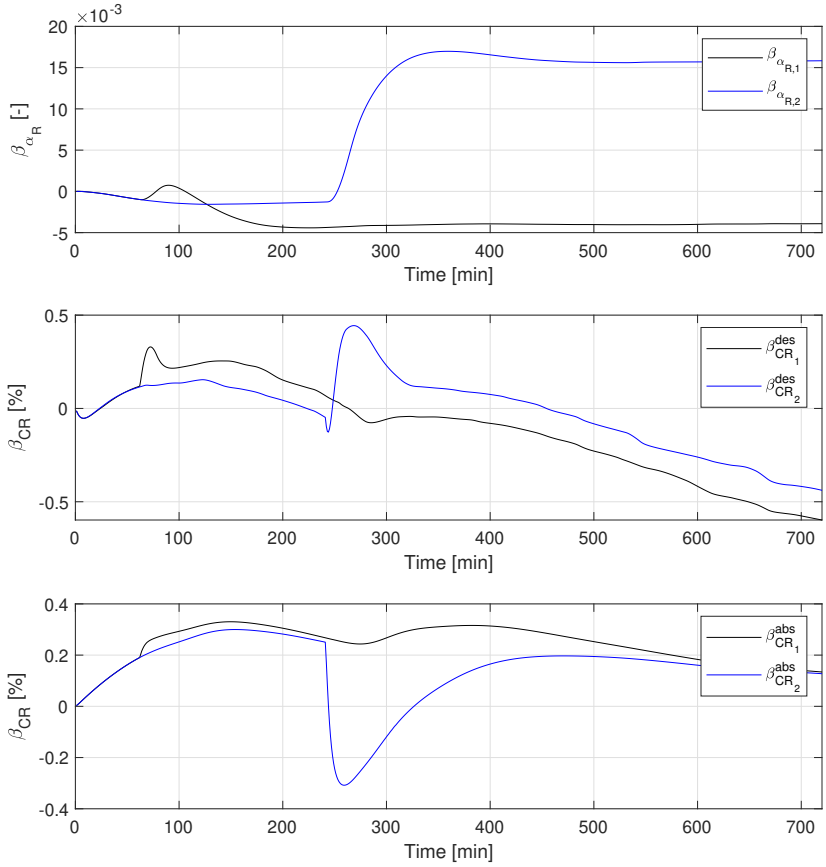



Figure 7.19: Bias variables from NMPC des and NMPC abs when using control configuration 4. The upper plot shows the rich loading bias variables from NMPC des for both absorbers, $\beta_{\alpha_{R,1}}$ and $\beta_{\alpha_{R,2}}$. The middle plot shows the CR bias variables in NMPC des, $\beta^{des}_{CR,1}$ and $\beta^{des}_{CR,2}$. The bottom plot shows the CR bias variables in NMPC abs, $\beta^{abs}_{CR,1}$ and $\beta^{abs}_{CR,2}$

The bias variables in NMPC des and NMPC abs can be seen in Figure 7.19. The upper plot shows the rich loading bias variables and the middle plot shows the CR bias variables from NMPC des. The bottom plot shows the CR bias variables from NMPC abs. The CR bias variables are quite small and pretty much settle at steady values towards the end of the horizon. The CR bias variables do not seem to settle at steady values.

### 7.1.3 Three-NMPC control configuration

The CV setpoint tracking performance when using the three-NMPC configuration from Section 6.2, number 5 in Table 7.1, can be seen in Figure 7.20. This plot is for NMPC abs. The upper plot shows the CR in absorber column 1 while the lower plot shows the CR in absorber column 2. The setpoint is followed closely and the flue gas disturbance is rejected quickly, there is however some deviation between measurements and prediction. Figure 7.21 shows the absorber CRs in NMPC des. The deviation between measurements and prediction is here as well. The CR predictions are a bit more noisy here. The setpoint is followed closely by the controller and the disturbance is rejected fast.

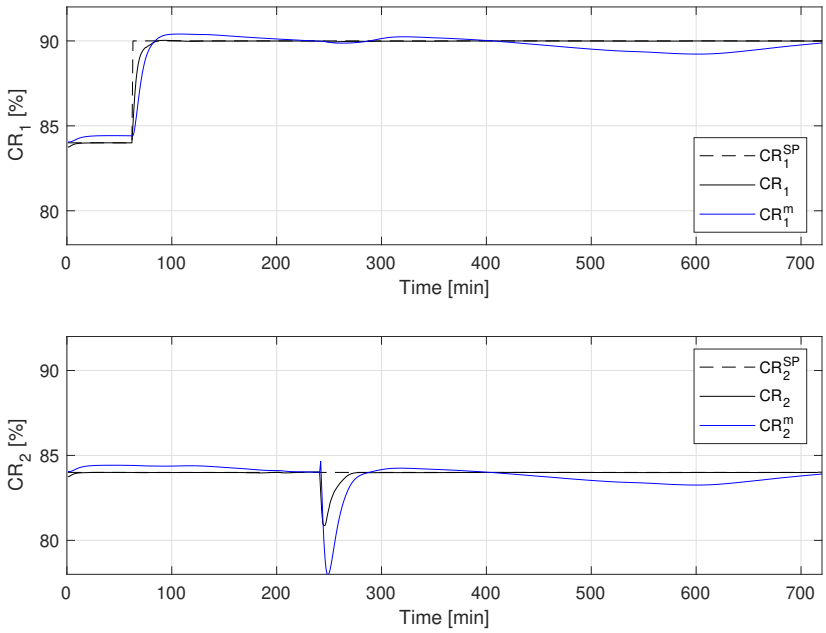



Figure 7.20: The CRs in NMPC abs when using control configuration 5. The upper plot shows the CR in absorber column 1 and the lower plot shows the CR in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted values in NMPC abs, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$.

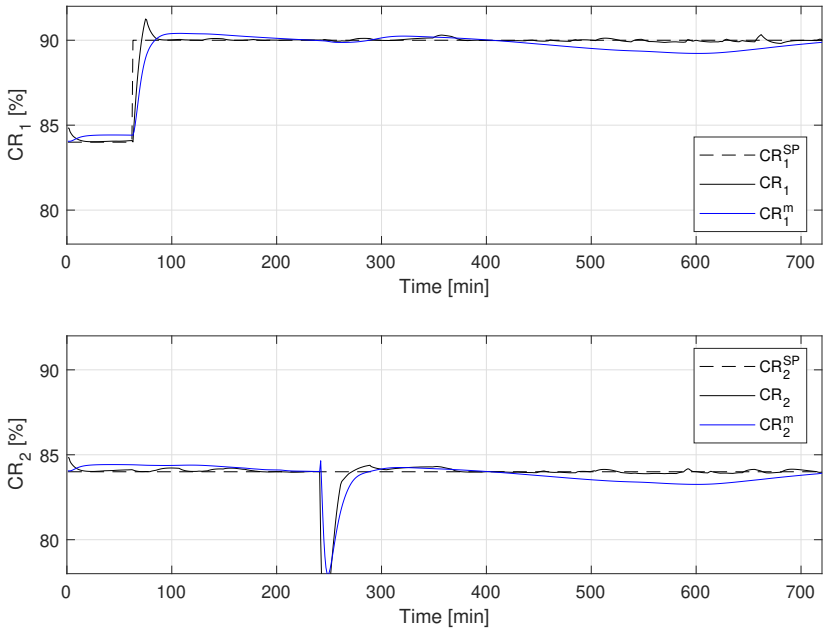



Figure 7.21: The CRs in NMPC des when using control configuration 5. The upper plot shows the CR in absorber column 1 and the lower plot shows the CR in absorber column 2. The stippled black lines are the setpoints, $CR_1^{sp}$ and $CR_2^{sp}$. The solid black lines are the predicted values in NMPC abs, $CR_1$ and $CR_2$. The blue lines are the measured CRs from the plant, $CR_1^m$ and $CR_2^m$.

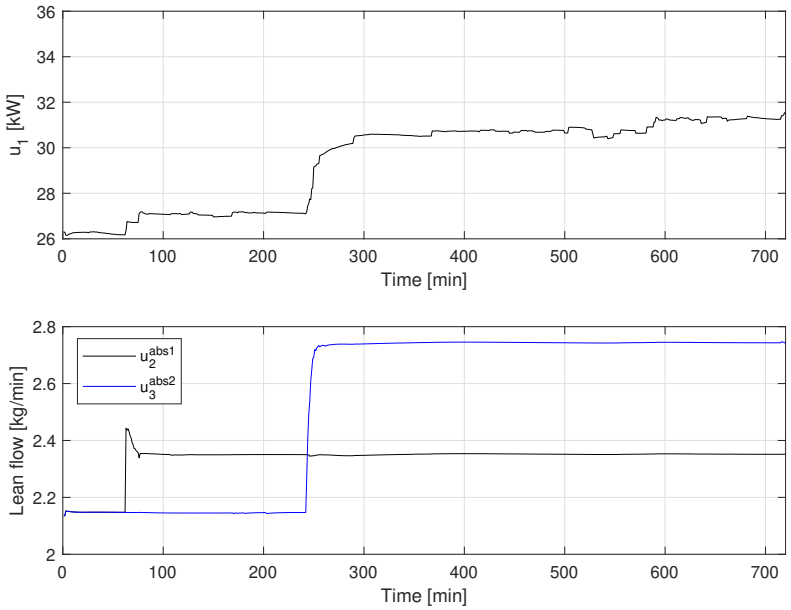



Figure 7.22: The MVs when using control configuration 5. The upper plot shows the reboiler duty, $u_1$. The lower plot shows the lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$.

The MVs can be seen in Figure 7.22. The upper plot shows the reboiler duty, $u_1$, while the lower plot shows the two lean flows calculated by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$. The lean flows look very similar to before. The reboiler duty is increased a little less after 240 minutes. It increases until the end of the horizon instead of being kept constant like in configuration 4.

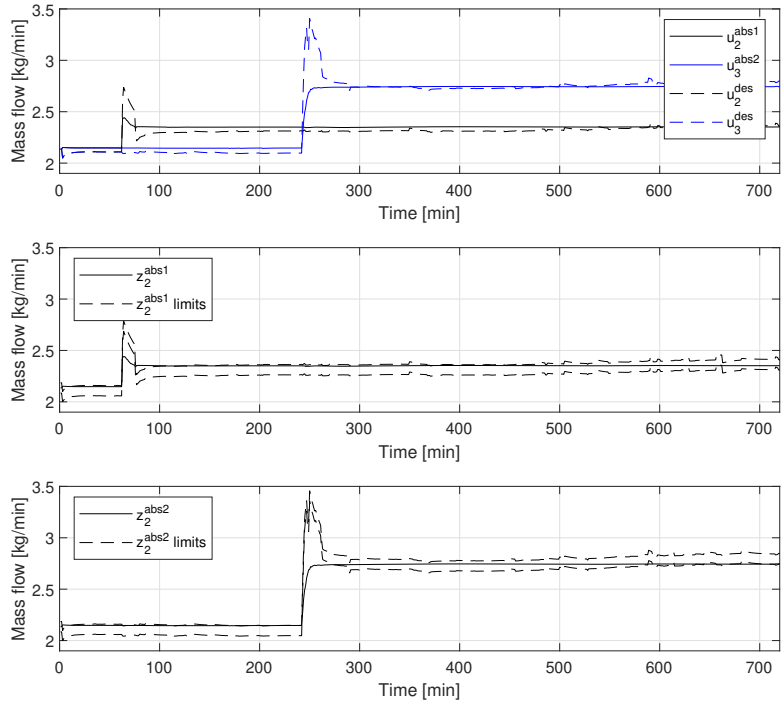



Figure 7.23: Comparison between the MVs calculated by NMPC abs and NMPC des when using control configuration 5. The upper plot shows the MVs, the solid lines are calculated by NMPC abs and the stippled lines are calculated by NMPC des. The middle and lower plots show $z_2$ in NMPC abs1 and NMPC abs2, respectively. The stippled lines are the limits.

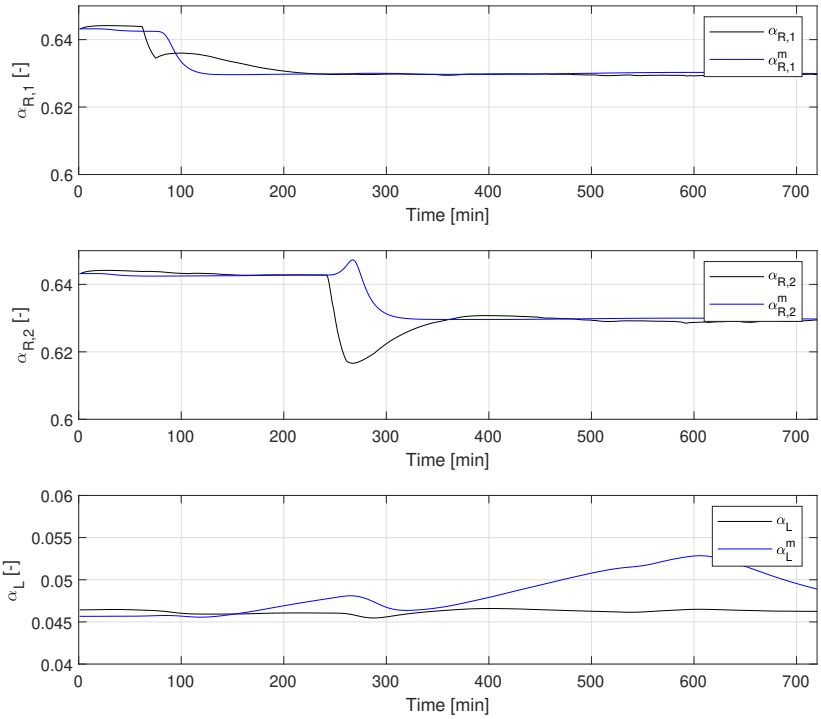



Figure 7.24: Lean and rich loading when using control configuration 5. The upper plot shows the rich loading in the outlet flows from absorber column 1. $\alpha_{R,1}$ is the value predicted by NMPC des and $\alpha^m_{R,1}$ is the measurement from the plant. The middle plot shows the rich loading in the outlet flows from absorber column 2, $\alpha_{R,2}$ is the value predicted by NMPC des and $\alpha^m_{R,2}$ is the measurement from the plant. The lower plot shows the lean loading in the lean buffer tank, $\alpha_L$ is the value predicted by NMPC des and $\alpha^m_L$ is the measurement from the plant.

A comparison between the lean flows found by NMPC abs, $u_2^{abs}$ and $u_3^{abs}$, and the ones found by NMPC des, $u_2^{des}$ and $u_3^{des}$, is given in the upper plot in Figure 7.23. The lower plot shows the CV, $z_3$, in NMPC abs and the limits which it receives from NMPC des. The lean flows found by NMPC des and NMPC abs1 and NMPC abs2 are pretty similar, but NMPC des chooses values that are a little lower and does bigger jumps at the setpoint change and the flue flow increase. The lean flows are mostly within their limits in both NMPC abs1 and NMPC abs2.

Figure 7.24 shows the loadings predicted by NMPC des and the ones measured from the plant. The upper plot shows the rich loading in the outlet flow coming from absorber column 1, the middle plot shows the rich loading in the outlet flow coming from absorber column 2. The lower plot shows the lean loading in the lean buffer tank. The rich loading predictions are good after the bias has updated them, but dynamically they are off. The lean loading measurement starts to drift off at 350 minutes, but returns towards the end of the horizon.

Figure 7.25 shows the bias variables. The upper plot shows the rich loading bias variables in NMPC des. The middle plot shows the CR bias variables in NMPC des and the bottom plot shows the CR bias variables in NMPC abs1 and NMPC abs2.

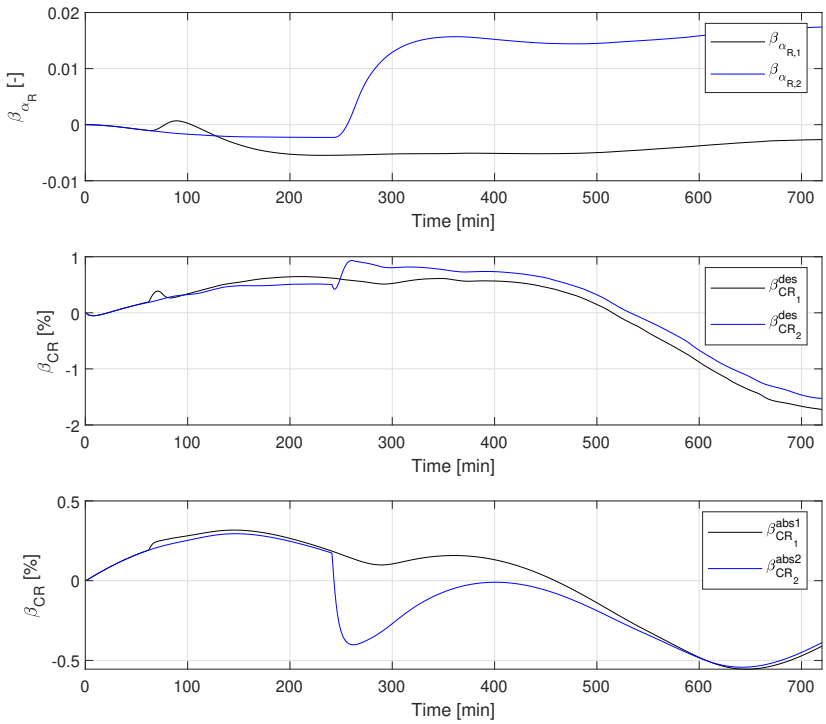



Figure 7.25: Bias variables from NMPC des and NMPC abs when using control configuration 5. The upper plot shows the rich loading bias variables from NMPC des for both absorbers, $\beta_{\alpha_{R,1}}$ and $\beta_{\alpha_{R,2}}$. The middle plot shows the CR bias variables in NMPC des, $\beta^{des}_{CR,1}$ and $\beta^{des}_{CR,2}$. The bottom plot shows the CR bias variables in NMPC abs1 and NMPC abs2, $\beta^{abs1}_{CR,1}$ and $\beta^{abs2}_{CR,2}$

A summary of the average computational time per sample for the different control configurations and their controllers is given in Table 7.2. The maximum sample time is also given. Splitting the optimisation problem into smaller pieces reduces the total computational time.

Table 7.2: Average time per sample and maximum sample time of controllers with a two absorber plant

| Configuration | Controller | Average time [s] | Max time [s] |
|---|---|---|---|
| 1 | NMPC | 22.59 | 77.67 |
| 2 | NMPC abs | 4.23 | 7.78 |
| | NMPC des | 2.09 | 10.52 |
| 3 | NMPC abs | 4.31 | 9.58 |
| | NMPC des | 1.56 | 6.18 |
| 4 | NMPC abs | 4.06 | 7.41 |
| | NMPC des | 1.89 | 3.12 |
| 5 | NMPC abs1 | 0.98 | 2.46 |
| | NMPC abs2 | 0.97 | 2.19 |
| | NMPC des | 1.87 | 5.55 |

## 7.2 $CO_2$ capture plant with three absorber columns

A test was done on a three absorber $CO_2$ capture plant. The test was just running the plant with the NMPCs switched on for 30 samples, with one sample being 60 seconds as before. No setpoint changes or disturbances were introduced. No plots are shown as the simulations are not interesting apart from the computational times. Control configuration 1, 4 and 5 from Table 7.1 were tested. The average computational times for different control configurations are given in Table 7.3. The maximum sample times are also given.

Table 7.3: Average time per sample and maximum sample time of controllers with a three absorber plant

| Configuration | Controller | Average time [s] | Max time [s] |
|---|---|---|---|
| 1 | NMPC | 67.03 | 94.84 |
| 4 | NMPC abs | 12.11 | 15.63 |
| | NMPC des | 1.62 | 2.02 |
| 5 | NMPC abs1 | 0.91 | 1.15 |
| | NMPC abs2 | 0.92 | 1.14 |
| | NMPC abs3 | 0.92 | 1.15 |
| | NMPC des | 1.5 | 2.24 |

Chapter 8

# Discussion

This chapter starts of with a discussion of some key decisions in the model and control configuration development. Then comes further discussion of the results and an evaluation of the final control configurations performance.

## 8.1 Modelling

Several simple absorber models were developed and implemented in the *desorber model*. In this section each of the simple absorber models are discussed. Both their performance and underlying assumptions. The conversion of the model from a one absorber to a multiple absorber $CO_2$ capture plant is also discussed

### 8.1.1 Step response models

The step response models in Section 4.2 fit well when the absorber column is at the conditions the step response models were found in, as can be seen in Figure 4.5. In this case that is with CR controlled at 84 % and 80 $m^3\,h^{-1}$ flue gas flow with 12.3 % $CO_2$. However, the dynamics of the step response models does not quite match the *absorber model*. In the step response model for rich loading there is some time delay which is not included in the step response model. Instead some more time was added to the time constant. Judging by Figure 4.5 there is inverse response in rich flow when increasing lean flow in the *absorber model* which also not included in the step response model. There is also some time delay which is not there in the step response model.

These dynamic differences are however not that critical as long as the steady state values are correct. The dynamics of the desorber column is slow compared to the absorber column and any NMPC which uses the step response models only controls the reboiler duty, which acts directly on the desorber column through the reboiler. Due to how the absorber columns are relatively faster than the desorber column the dynamic error of the step reponse models should not be a problem.

On the other hand, what can be a problem is when the conditions of the absorber column change. For instance a setpoint change will affect both the rich flow and the rich loading in the column. The values predicted by the models given a specific lean flow can then be wrong. Due to this it is common to use bias updating of the model when using step response models[19]. The plot in Figure 5.3 shows a comparison of the rich loading prediction with and without bias updating of the step response model. It can be seen that the models miss by quite a bit without bias updating, but with bias updating the error is removed.

### 8.1.2 Absorber model based on mass transfer

The absorber model based on mass transfer from Section 4.3 was by far the worst of the simple absorber models. Both in terms of prediction accuracy and general usability. There are many possible reasons for why the performance was so bad. One reason may be the fact that the temperature is assumed to be constant throughout the entire column. Looking at simulations using the *absorber model* this is certainly not the case. The temperature in the column varies quite a bit, increasing from the bottom towards the top until a drop in temperature before the lean solvent inlet at the top of the column. It was attempted to add a temperature dependence by linear interpolation of a temperature profile, but it was not successful.

Another assumption which likely caused error is the assumption that the total flows are constant due to the low $CO_2$ content. The $CO_2$ content in the flue gas in the simulations is 12.3 %, which is a likely too much for this assumption to hold. If the capture ratio is 84 % just over 10 % of the gas flow is removed. This would lead to changes in the mole fractions of the other components which are not negligible. The same applies to the liquid flow. The loading changes quite a bit from top to bottom. From simulating the *absorber model* a lean flow of $2.2\,\mathrm{kg\,min^{-1}}$ into the column gave a rich flow of $2.44\,\mathrm{kg\,min^{-1}}$ out of the column. This is a 9 % increase, which is too big to ignore.

Another possible issue with this model is the way it is solved. Shooting methods work well for linear models, but can fail if the model is too nonlinear. The algorithm for updating the initial value guess is also important. It was found that both secant method and Brent's method were unreliable and often failed. These method would often choose guesses which either lead to the model diverging towards infinity or returning NaN. The KINSOL solver was implemented and could solve the problem reliably.

The obvious solutions to these problems is to include temperature in the model, not assume that the flows are constant and finding a new way to solve the BVP. However, this would increase the complexity of the model and make it more computationally expensive. Adding temperature equations and more detailed flow equations would increase the size of the model and many parameters would have to be identified. Solving the model using either finite difference or a collocation method would also increase the complexity of the model to the point where it would be better to just use the original *absorber model*, which is tried and tested and known to be accurate.

### 8.1.3 Mole balance model

Since the dynamics of the absorber columns are much faster than the dynamics of the desorber column it was suggested that perhaps a steady state model of the absorber model could be good enough. The benefit of using a simple steady state mole balance model is that there is no need for thermodynamic equations, which can get complicated in an absorption process. The models in Section 4.4 were developed based on the assumption that a steady state model is good enough. First the model was formulated in a way that gave predictions of rich flow and rich loading, assuming the composition of the entering flue gas and lean amine are known. The CR must also be known. The second formulation assumes that rich flow and rich loading are known, they can then be used to calculate the CR in the column, as shown in Equation 4.20.

Both of these formulations are dependant on information predicted by other models. The first formulation needs to know the CR, which it receives from NMPC abs in the control structure in Section 5.3, test results of which can be seen in Section 7.1.2. A possible issue with this is that there may be differences between the actual CR in the plant and the one predicted by NMPC abs. This will lead to errors in the predictions of rich flow and rich loading by the model, even if the model is perfect. Similarly for the second formulation, the rich flow and rich loading must come from somewhere else. This formulation is used in the control configuration in Section 6.1, where rich flow and rich loading are predicted using a step response model. If the predictions from the step response models are wrong the CR will also be wrong. This will be discussed further later in the chapter in the sections of the relevant control configurations.

### 8.1.4 Assumptions when expanding to multiple absorbers

When the model is expanded to include multiple absorber columns it is assumed that it does not matter if the flow goes to one or several columns from the absorbers perspective. The only thing that matters in the lean buffer tank is the total amount of lean solvent leaving the tank. This should be a reasonable assumption. The lean solvent flows out of the tank and into the absorber columns, which may have different temperatures and compositions. The differences between the columns does not matter, as the contents of the lean buffer tank are not affected by what happens downstream.

In the heat exchanger it is assumed that the molar flows from each absorber column can be added to total molar flows entering the heat exchanger. Due to the conservation of mass this assumption holds. Temperature on the other hand is an intensive quantity, hence it is not possible to add the temperatures together like with the rich flows. To calculate he temperature as shown in Equation 4.30d, a few assumptions have to be made. It is assumed that there is no mixing enthalpy and that the flows have the same heat capacity. This assumption should hold as the rich flow coming from the different columns are similar. The rich loading varies a bit between the columns, same with the temperature. These properties affect both the heat capacity and the mixing enthalpy. The differences however are quite

small so these assumptions should be fine.

## 8.2 Control configurations

The development of the control configurations happened hand in hand with the model development. The first step was expanding the models to multiple absorber columns and adapting the control structure developed in the project thesis[17] to a multiple absorber plant, as explained in Section 5.1. This is the first of the two-NMPC configurations developed during this thesis. As explained in Chapter 7 the configurations were tested using the same test.

### 8.2.1 Configuration 1

Test results of control configuration 1 in Table 7.1 are given in Section 7.1. This is the benchmark control configuration with no model mismatch between plant and NMPC. Figure 7.1 shows the setpoint tracking performance of this control configuration. It performs very well, which is expected when there is no model mismatch. The desorber capture ratio is not a CV, it is only added for comparison with other configurations where it is a CV.

Figure 7.2 shows the MVs. It can be seen that the lean flows are changed quickly while the reboiler duty is changed a bit slower in comparison. This is likely due to how the inputs affect the plant and the CVs. Dynamically the CRs in the absorber columns are strongly dependant on the lean flows. An increase in lean flow will cause an immediate increase in CR, which explains why they are changed so quickly when either the setpoint changes or the flue flow increases. They are not very dependant on the reboiler duty dynamically, as the effects of changing it is a change in loading which in turn affects the CRs. The desorber is slow compared to the absorbers so a sudden increase in reboiler duty would not cause the same immediate increase in CR. That is not to say that the reboiler duty is not important, it just uses longer time to affect the plant.

Figure 7.3 shows the loadings in the plant and as expected there is almost no deviation between the measurements and the values from the NMPC. Overall this control structure performs very well with regards to how good the control is. It is far from perfect though, as solving the whole plant in one NMPC is very time consuming. The average computational time per sample and the maximum sample computational time are given in Table 7.2, for this control configuration and all the others later in the chapter. This control configuration is significantly slower than the rest with an average sample time of 22.59 seconds. The maximum computational time of a sample was 77.67 seconds, and considering that the sampling time of the controller is 60 seconds that is not good enough.

### 8.2.2 Configuration 2

The control configuration given in Section 5.1 used step response models to predict rich flow and rich loading. It is number 2 in Table 7.1. The results of this controller to the test are shown in Section 7.1.2. Looking at the CRs in Figure 7.4 it can be seen that this

configuration works quite well. There is some deviation between the measurements and the values predicted in NMPC abs, but it is a good starting point. The desorber CR is also controlled nicely, though there is some deviation here as well.

The MVs can be seen in Figure 7.5. Comparing with the MVs from control configuration 1, shown in Figure 7.2 it can be seen that the lean flows are quite similar. The lean flows in Figure 7.5 are calculated by NMPC abs and as long as the lean loading prediction received from NMPC des is close to the actual value in the plant they should be pretty similar, as the models in the *absorber model* are the same as in the *original model*.

The reboiler duty on the other hand is calculated by NMPC des. There is more difference in reboiler duty than in the lean flows. In control configuration 2 the reboiler duty is used quite aggressively and is increased much more before being reduced again. In control configuration 1 the reboiler is increased slower and does not have the same pattern of large increase before a decrease. The main contributing factor to this is likely the difference in CVs between the NMPCs. In configuration 1 the CVs are the CRs in the absorber columns which are quickly affected by the lean flows, while the reboiler duty takes a longer time to affect them. In configuration 2 on the other hand the CVs are the CR in the desorber column and the ratio between the lean flows. The reboiler duty has a large effect on the desorber CR as an increase in reboiler duty will lead to an increased amount of $CO_2$ leaving the reboiler and the desorber CR increasing. When either the CR setpoint is increased or the $CO_2$ flow into the absorbers increases the only way for NMPC des to respond quickly is by increasing the reboiler duty. Due to the locations of the evaluation points for $z_2 = u_1$ in NMPC des reboiler duty usage is only penalised after 100 minutes into the horizon, while deviations in CR are penalised after 10 minutes. Due to this NMPC des will choose to quickly increase the reboiler duty if it can help satisfy the CR setpoint.

A comparison between the lean flows calculated by NMPC abs and NMPC des is given in Figure 7.6. They are similar, but the ones calculated by NMPC des are a bit more changing up and down. NMPC abs finds the correct value and pretty much keeps it there. The lower plot shows $z_3$ from NMPC abs with its limits. The limits are violated momentarily, but NMPC abs ends up keeping them within the limits most of the time. Figure 7.8 shows the loadings in the columns and the lean buffer tank. It can be seen that the rich loadings predicted by NMPC des are lower in absorber 2 and a little larger in absorber 1. As the flow to absorber 2 is bigger the net result of this is likely that the rich loading entering the desorber column is smaller in NMPC des than in the plant, which can explain why the lean flows calculated by NMPC des are larger than the ones calculated by NMPC abs.

There is also some deviation between the reboiler duty in configuration 1 and 2 towards the end of the simulation. This might be due to the difference in rich loading. Figure 7.8 shows the loadings in configuration 2. It can be seen that the loading predicted by NMPC des in absorber column 2 is much lower than the actual loading in the plant. This will lead NMPC des to have less $CO_2$ in the desorber than is actually the case in the plant. The reboiler duty needed for NMPC des to fulfill the desorber CR setpoint will then be larger than what

is needed in the plant. This explains why the measured CRs are higher than the predicted values. Figure 7.7 shows $z_3$ in NMPC des. The setpoint is followed closely. Judging by the upper plot in Figure 7.6 the distribution of the flows in NMPC des seem to be correct, as it matches well with the distribution in NMPC abs.

### 8.2.3 Configuration 3

The test results for a controller using this simple absorber model is given in Section 7.1.2, this configuration is number 3 in Table 7.1. Figure 7.9 shows the CRs in the absorbers and in the desorber. It can be seen that the controller performs very well initially, but it starts to deviate at about 400-500 minutes into the simulation. This applies to both the absorber CRs and the desorber CR.

The MVs can be seen in Figure 7.10. NMPC abs is identical in control configuration 2 and 3, so the fact that the lean flows found by NMPC abs resemble the ones found by the NMPC in control configuration 1 is expected for the same reasons as before. The reboiler duty is not that similar to the one found buy configuration 1, but it is similar to the one found by configuration 2. This is because the reboiler duty is found by NMPC des which once again uses the desorber CR, which is more affected by the reboiler duty than the CRs in the absorber columns.

Figure 7.11 shows a comparison between the lean flows calculated by NMPC abs and by NMPC des. In this configuration NMPC des chooses smaller lean flows than NMPC abs. Looking at the loadings in Figure 7.12 might explain this. The rich loadings predicted in NMPC des in this configuration are larger than the actual loading in the absorber columns. This means that less rich flow is needed into the desorber to get the same amount of $CO_2$ and the lean flows becomes lower than what they should be. This is the opposite of what happens in configuration 2, where the rich loading is too low and the lean flows become higher than they should be.

Looking at the bottom plot in Figure 7.11 it can be seen that NMPC abs chooses to violate the constraints all the time in order to satisfy the CR setpoints. The difference between the flows calculated by NMPC abs and NMPC des is an issue probably from the errors in rich loading prediction.

After about 450 minutes the CRs measured from the plant starts to drop. The explanation for this can once again be found by looking at the loadings in Figure 7.12. At about the same time as the CRs begin to drop the lean loading measured from the plant starts to increase. The lean loading predicted in the model on the other hand stays the same place. NMPC abs uses the lean loading trajectory it receives from NMPC des, and when this is wrong the capture ratio predicted in NMPC abs will also be wrong. If the lean loading is small there is need for less lean flow than when the lean loading is high. This means that when NMPC abs receives lean loading predictions that are lower than the lean loading in the plant it will choose lean flows which are too small to capture the desired amount of $CO_2$ in the plant.

The explanation for why the lean loading in the plant starts to increase can be found by comparing the reboiler duties in Figure 7.10 and Figure 7.2. Towards the end of the horizon NMPC des in control configuration 3 chooses a lower reboiler duty than configuration 1, while the lean flows chosen by NMPC abs are almost identical to the ones in configuration 1. The consequence of this is naturally that the lean loading increases as less $CO_2$ is stripped away when the reboiler duty is lower. The root cause of the issues is the error in rich loading predictions. By predicting larger rich loading than what is actually the case it allows NMPC des to choose lower lean flows and reboiler duty than what is actually needed in the plant while still achieving its desorber CR setpoint.

Stripping $CO_2$ is easier the higher the $CO_2$ loading is. This explains why NMPC des can remove enough $CO_2$ while using lower reboiler duty. The reboiler duty required per kg $CO_2$ removed is known as specific reboiler duty (SRD). SRD is lower when the rich loading is higher, hence less reboiler duty is required to remove the same amount of $CO_2$.

In NMPC des. the value of the CR in the simple model in Subsection 4.4.1, $c$, is received in the form of a predicted trajectory from NMPC abs. When there is a difference between the prediction and the actual value in the plant this will lead to error in the predicted rich flow and rich loading in NMPC des. These errors affect NMPC des and the inputs and predictions it produces. The lean loading trajectory sent from NMPC des to NMPC abs will then also be wrong which leads to a larger error in the CR prediction and the cycle continues. It seems that sending predicted trajectories both ways is not wise at it can lead to a snowballing effect where the deviation grows.

### 8.2.4 Configuration 4

Based on the results of configurations 2 and 3 it was concluded that the predicted rich loading had to be correct for NMPC des to be able to choose the correct reboiler duty. The way this was achieved was by combining the two previous control configurations and adding bias updating based on measurements from the plant. Based on the results of configuration 2 it seemed that the step response models did a better job of predicting rich flow and loading than the mole balance model in configuration 3. The mole balance in configuration 3 was not good for predicting rich loading, but it was decided to change it so that it could be used to predict CR in the absorbers instead. This allows NMPC des to use the absorber CRs as CVs instead of the desorber CR and the ratio of the lean flows. This is better because the absorber CRs is what we actually want to control. We now use the step response models to predict rich flow and rich loading and the mole balance model to calculate the absorber CRs based on these predictions.

Besides giving the correct CVs in NMPC des there is also one more advantage with this configuration. It is now possible to have bias updating on the CVs in NMPC des, which ensures offset free control through integral action. Bias updating is also done on teh CVs in NMPC abs. In NMPC des there is also bias updating on the step response models to make it predict the correct rich loading. This was important to achieve as correct rich loading is

essential for choosing the correct reboiler duty.

The results of this controller to the test can be seen in Section 7.1.2. Comparing the CRs in Figure 7.14 with the absorber CRs in Figure 7.1 there is very little difference. Both have almost no difference between measured and predicted values and the setpoints are followed fast and closely. Figure 7.15 shows the CRs from NMPC des. There is not much difference between NMPC abs and NMPC des, but NMPC des's predictions are perhaps a bit more noisy.

The MVs in Figure 7.16 and Figure 7.2 are also very similar. Interestingly the jump in reboiler duty observed in configuration 2 and 3 is now gone, this is because the desorber CR is no longer a CV, instead the absorber CRs are. The final value towards the end of the horizon is also similar. Figure 7.17 shows the lean flows calculated by both NMPC abs and NMPC des. The jump in reboiler duty in NMPC des has instead been replaced by a jump in the lean flows. See that after the jumps NMPC des settles down at just about the same value as NMPC abs. Of all the two-NMPC configurations this is the one which most resembles the inputs from configuration 1, and the one where the lean flows found by NMPC abs and NMPC des are the most similar.

The rich and lean loading can be seen in Figure 7.18. From the rich loading plots it can be seen that the bias updating does what it is supposed to do. The deviation between prediction is completely removed after some time. Dynamically there is some difference though. The huge drops in the predicted rich loadings, which occur at both 60 and 240 minutes does not happen in the actual plant. This combined with Equation 4.20 can explain why NMPC des chooses the large jumps in lean flow. When the rich loading drops $x_{0,\mathrm{CO}_2}$ in Equation 4.20 also decreases. This causes the CR to decrease and the easiest way to counteract it is by increasing the lean flow which makes first $L_1$ then $L_0$ increase. The lean loading prediction is also better here than in the previous configurations. The improved accuracy of the predicted loadings also leads to the reboiler duty being closer to the reboiler duty from configuration 1.

The bias variables are shown in Figure 7.19. The rich loading bias variables seem to settle at steady values, which combined with the rich loading plots in Figure 7.18 shows that the bias variables manage to update the step response models such that they predict the correct rich loading. The bias variables for CR does not seem to settle at steady values, neither in NMPC abs nor in NMPC des. Perhaps the simulations are not long enough and they settle in the horizon.

Overall this control configuration performed excellently. The differences between the performance of this configuration and configuration 1 are small with regards to the how well they control the plant. Looking at the times in table Table 7.2 however, it is a clear advantage for configuration 4. The average computational time per sample in configuration 1 is 22.59 seconds, while the average computational time in configuration 4 is 4.06 seconds in NMPC abs and 1.89 seconds in NMPC des. In total that is 5.95 seconds per sample. This is a

reduction in computational time of 73.6 %. The maximum sample time was reduced from 77.67 seconds to 10.53 seconds, an 86.4 % reduction.

### 8.2.5 Configuration 5

As explained in Section 5.6 and Section 6.2, NMPC abs can be split into an individual NMPC for each absorber column. This resulted in control configuration 5, the test results of which are given in Subsection 7.1.3. This configuration is identical to configuration 4, except NMPC abs is split into two NMPCs and NMPC des now sends individual lean flow trajectories to each absorber instead of the sum.

It was expected that this configuration would behave similarly to configuration 4. The CRs can be seen in Figure 7.20. This is similar, but there are some key differences. There is more deviation between the measurements and the predictions, even though the bias updating works in the same way with the same gain. This applies to the CR in both columns. Figure 7.21 shows the CRs in NMPC des, the plots look similar to the plots from NMPC abs, but they are perhaps a bit more noisy here as well. The MVs can be seen in Figure 7.22. The lean flows look very similar to the ones calculated by configuration 4, but the reboiler duty is a bit different. In configuration 5 NMPC des for some reason uses less reboiler duty initially, but has to increase it later, while in configuration 4 NMPC des chooses a higher value straight away and keeps it quite steady.

The upper plot in Figure 7.23 shows a comparison between the lean flows calculated by NMPC abs and NMPC des. The flows calculated by NMPC abs are a bit higher than the ones calculated by NMPC des, but the difference is small. The difference is however a bit bigger in this configuration than in configuration 4. The middle and bottom plot show $z_2$ in NMPC abs1 and NMPC abs2, respectively. The variables are kept within their constraints most of the time, but the constraints are violated when the large jumps in flow occur in NMPC des.

The loadings are shown in Figure 7.24. The bias updating does it job here as well, but there seems to be a bit more error here than in configuration 4, seen in Figure 7.18. The difference in lean loading however is easy to spot. The lean flows seem to be almost identical in configuration 4 and 5, so the explanation for this deviation in lean loading is probably the difference in reboiler duty. The reboiler duty is a bit lower in configuration 5 which explains why the lean loading increases. Towards the end of the simulation the reboiler duty is increased which leads to the lean loading sinking again.

It is difficult to explain why there are these differences between configuration 4 and 5. There are only tiny differences between the lean flows found by NMPC abs in configuration 4 and the lean flows found by NMPC abs1 and NMPC abs2 in configuration 5. There are bigger differences when looking at the inputs from NMPC des. The lean flows seem to increase a bit slower here than in configuration 4. The biggest difference is in the reboiler duty, which is increased less after the second disturbance than in configuration 4, at least until nearer the

end where NMPC des in configuration 5 increases the reboiler duty a bit more. The reboiler duty in configuration 5 is also perhaps going a bit more up and down. Why NMPC des does this is difficult to say. The only difference between NMPC des in configuration 4 and 5 is that in 4 it sends the total lean flow to NMPC abs, while in 5 it sends lean flow 1 to NMPC abs1 and lean flow 2 to NMPC abs2. Why this affects NMPC des in this way is unknown.

The performance is perhaps not as good as in configuration 4 with regards to how well the plant is controlled, but it is an improvement with regards to computational time. As can be seen in Table 7.2 the average sample computational time in configuration 4 is 1.89, while in configuration 5 it is 1.87. This reduction is small and likely just a coincidence. When it comes to NMPC abs however the difference is bigger. The sum of the computational time in NMPC abs1 and NMPC abs2 is 1.95 seconds, while NMPC abs in 4 used 4.06 seconds. The total time reduction is 35.8 % compared to configuration 4 and 83.1 % compared to configuration 1.

## 8.3 Three column $CO_2$ capture plant

The test run with the three absorber $CO_2$ capture plant was very simple. It consisted of just running the plant at steady state for 30 samples with the controllers switched on. The average computational times per sample and the maximum sample times are given in Table 7.3. The advantage in terms of computational time reduction when splitting the problem into smaller pieces becomes even clearer here. The average time for configuration 1 is 67.03 seconds. When the sample rate from the plant is 60 seconds it is obvious that this configuration is not viable without either changing the sample rate or the tuning of the controller in some way.

For configuration 4 the average computational time per sample is 13.73 in total, a 79.5 % decrease compared to configuration 1. The average computational time in configuration 5 is 4.25 seconds, a 93.7 % decrease compared to configuration 1. This is a huge reduction of the computational time and really illustrates the effect of splitting the problem into smaller pieces. Judging by the times for NMPCs running only one absorber column in Table 7.2 and Table 7.3 it seems that they use just under a seconds on average. If there are individual NMPCs for each column there is no reason for this number to go up as they are solved independently.

The maximum times are not that relevant here. In the test of the two absorber plant the maximum times happen just after the disturbance or setpoint change, but here there are no disturbances. The maximum times in Table 7.3 are likely from the first sample when the NMPCs are run for the first time.

## 8.4 General discussion

Based on just the computational times there is no doubt which configuration is the best. There are however some differences in how NMPC des behaves in configuration 5 which

makes its performance worse than in configuration 4. The reason behind these differences is yet to be identified. A priority in future work with these configurations would be to get to the bottom of this and remove the problem. Because of the significant reduction in computational time it is possible to experiment with different aspects of the controller. For instance the CV or MV parameterisation could be changed to give more accurate control. The prediction horizon can be expanded and it could be possible to use an EKF instead of just bias updating. This particularly applies to configuration 5, but also to configuration 4.

Another advantage of splitting NMPC abs into individual NMPCs for each absorber column is that it is more decentralised. It allows for different companies running their own absorber column while the desorber column is shared among them. It also allows parallel computing. The absorber column NMPCs are independent of each other, hence they can be run at the same time. This means that the total time of configuration 4 could be reduced to the time for NMPC des plus the time of whatever NMPC abs which used the longest time.

The test run with a three column plant were just meant to test how adding an extra absorber column affects the computational time, not how well they control the plant. One problem that appeared when expanding the model is the fact that the columns are designed to be in a $CO_2$ capture plant with one absorber and one desorber. This means that in a multi-absorber plant either the desorber must be operated near the limits of its capacity or the absorbers have to have very low flue flow compared to what they can handle. For a proper study time should be spent redimensionalising the columns. In this thesis the absorbers are run with low flue gas amounts to not be too close to the limits of the desorber. This seemed to work fine, but the results may not be representative for a plant which is run at different conditions where the dimensions are more balanced. The step response models fit very well to the column when the flow was low, perhaps they are not as linear closer to their capacity limit. The $CO_2$ concentration can also change and could affect the validity of a linear model.

Chapter 9

# Conclusion

The foundations of the work in this thesis were laid during a summer internship and a subsequent specialisation project in collaboration in Cybernetica AS[17]. In the specialisation project it was concluded that splitting the $CO_2$ capture plant into smaller pieces could significantly reduce the computational times while offering acceptable control performance. Based on the results in the project it was decided that one of the control configurations developed there would be further developed during the master's thesis.

The configuration from the specialisation project was a two-NMPC configuration, one of the NMPCs controlled the capture ratio in the controller by manipulating the lean flow and the other controlled the capture ratio in the desorber by manipulating the reboiler duty and a local lean flow. This configuration was first expanded to two absorber columns. The performance was quite good, but there were some deviation between measurements and predictions which were thought to be due to errors introduced by the step-response models in the desorber NMPC.

Several different simple absorber models were developed and tested during the development process. Some of the models were implemented in different control configurations and their performance was tested. The development was an iterative design process where the knowledge gained from one implementation was used to decide the next step. The work culminated in two final control configurations. The first was a two-NMPC configuration which offered great control of the plant, comparable with the benchmark controller, while also being a significant improvement with regards to computational time. In the two-absorber test it gave a 73.6 % reduction in average computational time per sample. The second was a multi-NMPC configuration with individual NMPCs for each absorber column. The control was a little worse than the two-NMPC configuration, but it gave a 83.1 % reduction in average computational time per sample in the same test. In the three-absorber test the time reduction was 79.5 % for the two-NMPC configuration and 83.1 % for the multi-NMPC configuration. The control configurations were able to control the CRs at their setpoint while steering the plant towards minimum reboiler duty.

## 9.1 Future work

As mentioned in the introduction, the work in this master's thesis is related to the EU project REALISE CCUS, which brings together partners from science and industry to demonstrate the full $CO_2$ chain for industrial clusters centered on refineries. The aim of REALISE is to formulate a sustainable, cost-effective CCUS process that can be widely replicated. The technologies will be demonstrated at Irving Oil's Whitegate refinery in Ireland. The work in this thesis could prove useful for Cybernetica and their role in the REALISE project.

Future work could include expanding the models to include even more absorber columns. It could also be interesting to implement larger buffer tanks for both rich and lean amine, which can be utilised in a more long term optimisation where the energy price varies, similar to what Cybernetica have done before[8].

As mentioned in the discussion the absorber and desorber columns in this thesis are designed to be in a $CO_2$ capture plant with one of each column. Hence either the desorber has to work close to the limit or the absorbers have to have very low $CO_2$ amounts in a multi-absorber plant. To get more accurate results either the desorber should be redimensionalised to be able to handle larger $CO_2$ amounts, or the absorbers should be redimensionalised such that they are operated at reasonable percentage of their capacity. It is not given that the performance of the control configurations developed in this thesis will be the same then. Different $CO_2$ sources also have different concentrations of $CO_2$, this should also be experimented with to see how it affects the control.

# Bibliography


[1] Mann, M. E. Greenhouse gas. https://www.britannica.com/science/greenhouse-gas.

[2] Karbondioksid (CO2) (2020). URL https://miljostatus.miljodirektoratet.no/tema/klima/norske-utslipp-av-klimagasser/co2/.

[3] Global greenhouse gas emissions data. https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data.

[4] Carbon dioxide levels continue at record levels, despite COVID-19 lockdown (2020). URL https://public.wmo.int/en/media/press-release/carbon-dioxide-levels-continue-record-levels-despite-covid-19-lockdown.

[5] Jansen, D., Gazzani, M., Manzolini, G., van Dijk, E. & Carbo, M. Pre-combustion $CO_2$ capture. *International Journal of Greenhouse Gas Control* **40**, 167 – 187 (2015). URL http://www.sciencedirect.com/science/article/pii/S1750583615001917.

[6] Flø, N. E. *Post-combustion absorption based $CO_2$ capture: modeling, validation and analysis of process dynamics*. Ph.D. thesis, Norwegian University of Science and Technology, Trondheim (2015).

[7] Demonstration of Optimal Control of Post-Combustion Capture Processes (DOCPCC). URL https://tcmda.com/no/demonstration-of-optimal-control-of-post-combustion-capture-processes-docpc

[8] Kvamsdal, H. M. *et al.* Demonstration of two-level nonlinear model predictive control of $CO_2$ capture plants. *GHGT-14* (2018).

[9] Hauger, S. O. *et al.* Demonstration of non-linear model predictive control of post-combustion $CO_2$ capture processes. *Computers and Chemical Engineering* **123**, 184–195 (2019).

[10] Panahi, M. & Skogestad, S. Economically efficient operation of $CO_2$ capturing process part I: Self-optimizing procedure for selecting the best controlled variables. *Chemical*

*Engineering and Processing: Process Intensification* **50**, 247–253 (2011). URL https://www.sciencedirect.com/science/article/pii/S0255270111000444.

[11] Panahi, M. & Skogestad, S. Economically efficient operation of $CO_2$ capturing process. Part II. Design of control layer. *Chemical Engineering and Processing: Process Intensification* **52**, 112–124 (2012). URL https://www.sciencedirect.com/science/article/pii/S0255270111002406.

[12] He, X., Wang, Y., Bhattacharyya, D., Lima, F. V. & Turton, R. Dynamic modeling and advanced control of post-combustion $CO_2$ capture plants. *Chemical Engineering Research and Design* **131**, 430–439 (2018). URL https://www.sciencedirect.com/science/article/pii/S0263876217306937. Energy Systems Engineering.

[13] Sultan, T., Zabiri, H., Ali Ammar Taqvi, S. & Shahbaz, M. Plant-wide MPC control scheme for $CO_2$ absorption/stripping system. *Materials Today: Proceedings* **42**, 191–200 (2021). URL https://www.sciencedirect.com/science/article/pii/S2214785320391318. International Conference of Chemical Engineering & Industrial Biotechnology.

[14] Demonstrating a refinery-adapted clusterintegrated strategy to enable full-chain CCUS implementation. URL https://realiseccus.eu/sites/default/files/realise_briefing_document_final.pdf.

[15] Foss, B. & Heirung, T. A. N. *Merging Optimization and Control* (2016).

[16] Strand, S. & Sagli, J. R. MPC in Statoil – Advantages with In-House Technology. *IFAC Proceedings Volumes* **37**, 97–103 (2004). URL https://www.sciencedirect.com/science/article/pii/S1474667017387153. 7th International Symposium on Advanced Control of Chemical Processes (ADCHEM 2003), Hong-Kong, 11-14 January 2004.

[17] Tjessem, V. Simplified absorber model for use with model predictive control of a $CO_2$ capture plant (2020).

[18] Rawlings, J. B., Mayne, D. Q. & Diehl, M. M. *Model Predictive Control: Theory, Computation, and Design* (Nob Hill Publishing, 2019), 2 edn.

[19] Qin, S. J. & Badgwell, T. A. A survey of industrial model predictive control technology. *Control Engineering Practice* **11**, 733–764 (2003).

[20] Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* **13**, 291–309 (2003). URL https://www.sciencedirect.com/science/article/pii/S0959152402000628.

[21] Ziegler, J. G. & Nichols, N. B. Optimum settings for automatic controllers. *Trans. A.S.M.E* **64**, 759–768 (1942).

[22] Xie, H.-B., Zhou, Y., Zhang, Y. & Johnson, J. K. Reaction Mechanism of Mo-

noethanolamine with $CO_2$ in Aqueous Solution from Molecular Modeling. *Journal of Physical Chemistry A* **114**, 11844–11852 (2010).

[23] van Os, P. Final publishable summary report. Tech. Rep. (2011). URL https://cordis.europa.eu/docs/results/213/213569/134654861-8_en.zip.

[24] Brúder, P., Lauritsen, K. G., Mejdell, T. & Svendsen, H. F. $CO_2$ capture into aqueous solutions of 3-methylaminopropylamine activated dimethyl-monoethanolamine. *Chemical Engineering Science* **75**, 28–37 (2012). URL https://www.sciencedirect.com/science/article/pii/S0009250912001686.

[25] Aarseth, J. B. Numeriske beregninger Kompendium (2014). URL https://folk.ntnu.no/leifh/teaching/tkt4140/NumeriskeBeregninger.pdf.

[26] Nocedal, J. & Wright, S. J. *Numerical Optimization* (Springer, 2006), 2 edn.

[27] Brent, R. P. *Algorithms for minimization without derivatives* (Prentice-Hall, 1973).

Appendix A

# Constant values

Table A.1: Constants in the absorber column

| Description | Value | Unit |
|---|---|---|
| Inside diameter | 0.2032 | m |
| Cross sectional area | 0.032429 | $m^2$ |
| Packing height | 19.418 | m |
| Gas flow rate reference | 0.0617 | $kmol\,m^{-2}\,s^{-1}$ |
| Liquid flow rate reference | 0.0308 | $kmol\,m^{-2}\,s^{-1}$ |

Table A.2: Constants in the desorber column

| Description | Value | Unit |
|---|---|---|
| Inside diameter | 0.163 | m |
| Cross sectional area | 0.020867 | $m^2$ |
| Packing height | 13.6 | m |
| Gas flow rate reference | 0.0376 | $kmol\,m^{-2}\,s^{-1}$ |
| Liquid flow rate reference | 0.0537 | $kmol\,m^{-2}\,s^{-1}$ |

Table A.3: Constants in the absorber sump

| Description | Value | Unit |
|---|---|---|
| Inside diameter | 0.2032 | m |
| Tank height | 1 | m |
| Cross sectional area | 0.032429 | $m^2$ |
| Tank volume | 0.032429 | $m^2$ |
| Valve constant for liquid outflow | 0.00015 | $m^3\,kPa^{-1}\,s^{-1}$ |
| Gain, outlet liquid flow | 1 | - |
| Pressure drop | 1 | kPa |
| Default level setpoint | 320 | mm |
| Flow rate reference | 0.001 | $kmol\,s^{-1}$ |
| Mol amount reference | 0.7653 | kmol |

Table A.4: Constants in the heat exchanger

| Description | Value | Unit |
|---|---|---|
| Inner diameter, inner tube | 0.025 | m |
| Inner diameter, outer tube | 0.0396 | m |
| Tube length | 74.5 | m |
| Total cross sectional area, outer tube | 0.00049 | $m^2$ |
| Total cross sectional area, inner tube | 0.00074 | $m^2$ |
| Nominal rich flow | 0.1167 | $kg\,s^{-1}$ |
| Heat transfer coefficient | 0.3788 | - |
| Flow rate reference, hot side | 2.0372 | $kmol\,s^{-1}$ |
| Flow rate reference, cold side | 1.35 | $kmol\,s^{-1}$ |

Table A.5: Constants in the lean buffer tank

| Description | Value | Unit |
|---|---|---|
| Inside diameter | 0.3 | m |
| Tank height | 2 | m |
| Cross sectional area | 0.0707 | $m^2$ |
| Tank volume | 0.1414 | $m^3$ |
| Level setpoint | 126.26 | mm |
| Mol amount reference | 2.5803 | kmol |

Table A.6: Constants in the reboiler

| Description | Value | Unit |
|---|---|---|
| Gas flow rate reference | 0.0007 | $\mathrm{kmol\,s^{-1}}$ |
| Liquid flow rate reference | 0.001 | $\mathrm{kmol\,s^{-1}}$ |
| Time constant for inter-phase fluxes | 120 | s |
| Total volume | 0.43 | $\mathrm{m^3}$ |
| Base area | 0.6 | $\mathrm{m^2}$ |
| Valve constant for outlet gas flow | 0.01 | $\mathrm{m^3\,kPa^{-1}\,s^{-1}}$ |
| Valve constant for outlet liquid flow | 0.025 | $\mathrm{m^3\,kPa^{-1}\,s^{-1}}$ |
| Gain for outlet liquid flow | 1 | - |
| Pressure drop | 1 | kPa |
| Default level setpoint | 450 | mm |
| Liquid amount reference | 4.929 | kmol |
| Gas amount reference | 0.0165 | kmol |
| Interfacial heat transfer coefficient | 0.1 | $\mathrm{kW\,kmol^{-1}}$ |
| Heat transfer coefficient to ambient | 0.0 | $\mathrm{kW\,kmol^{-1}}$ |

Table A.7: Constants in the condenser

| Description | Value | Unit |
|---|---|---|
| Condenser volume | 0.1 | $\mathrm{m^3}$ |
| Base area | 0.2 | $\mathrm{m^2}$ |
| Valve constant for outlet gas flow | 0.00012 | $\mathrm{m^3\,kPa^{-1}\,s^{-1}}$ |
| Valve constant for outlet liquid flow | $3.67 \times 10^{-5}$ | $\mathrm{m^3\,kPa^{-1}\,s^{-1}}$ |
| Gain for outlet liquid flow | 1 | - |
| Pressure drop | 1 | kPa |
| Gain, energy | 0.2 | - |
| Volume setpoint | 0.01 | $\mathrm{m^3}$ |
| Liquid amount reference | 0.5477 | kmol |
| Gas amount reference | 0.0069 | kmol |
| Gas flow rate reference | 0.0007 | $\mathrm{kmol\,s^{-1}}$ |
| Liquid flow rate reference | 0.0007 | $\mathrm{kmol\,s^{-1}}$ |
| Time constant for inter-phase fluxes | 40 | s |

# Appendix B

# Simulation tools

An important part of the work in this master's thesis consisted of implementing nonlinear model predictive controllers (NMPC) in Cybernetica AS's software. Software tools developed by Cybernetica were made available and assistance was given by the co-supervisors from Cybernetica AS. The software used in this master's thesis work include Cybernetica RealSim and Cybernetica CENIT.

Cybernetica RealSim is a process simulator and was used as a plant replacement for testing of the developed NMPCs. When using RealSim the modules are run consecutively starting with the process simulator before one or more NMPCs. Figure B.1 shows an example of what

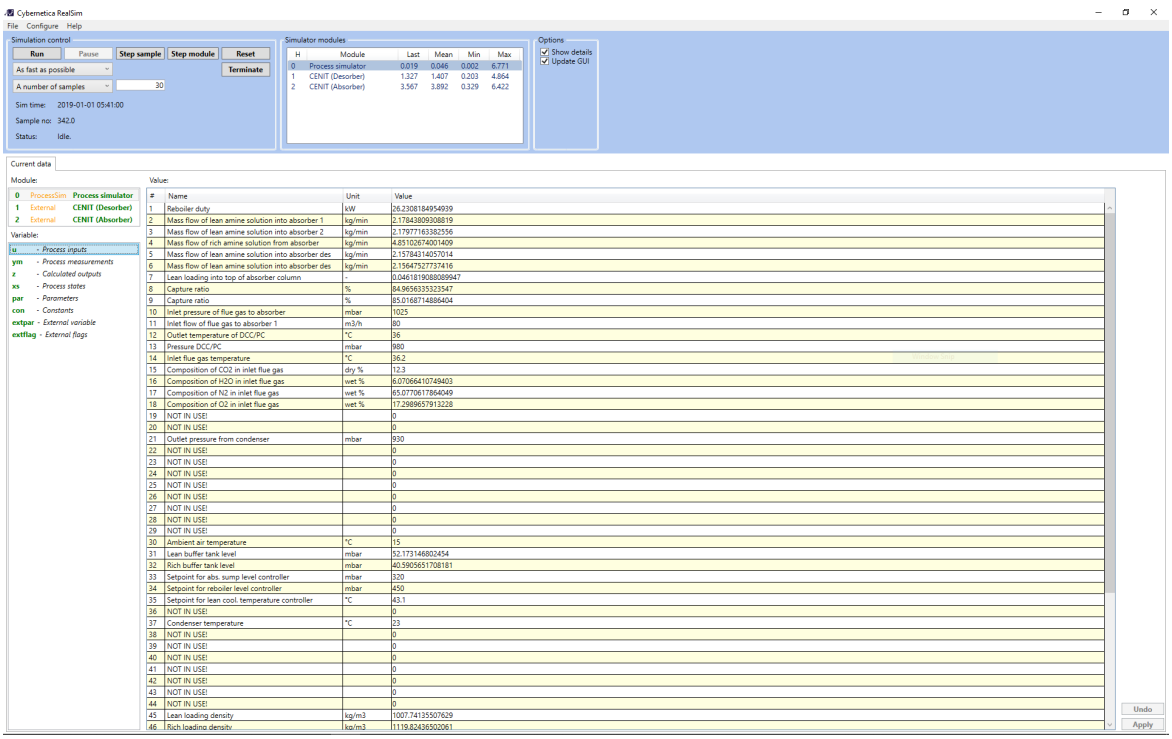
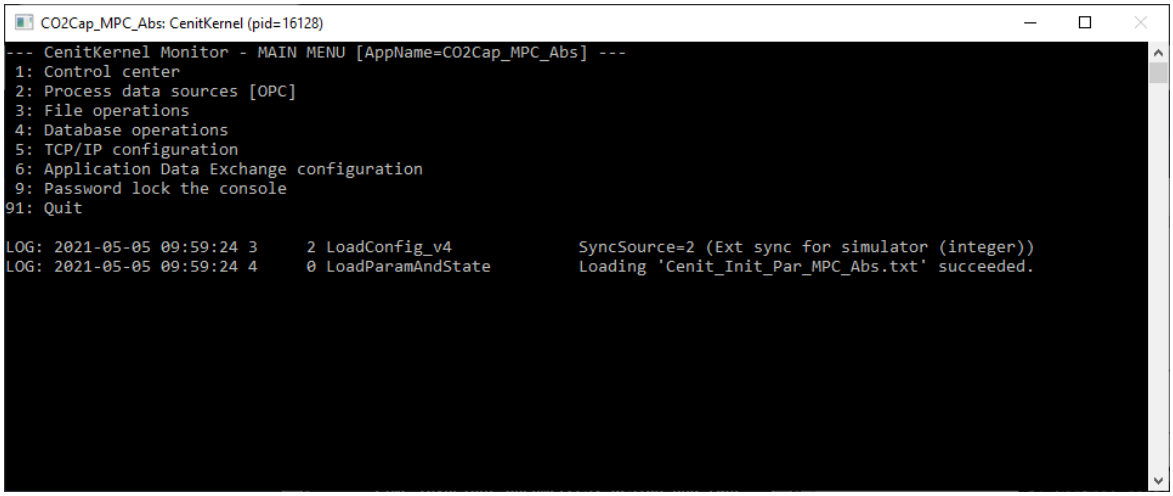
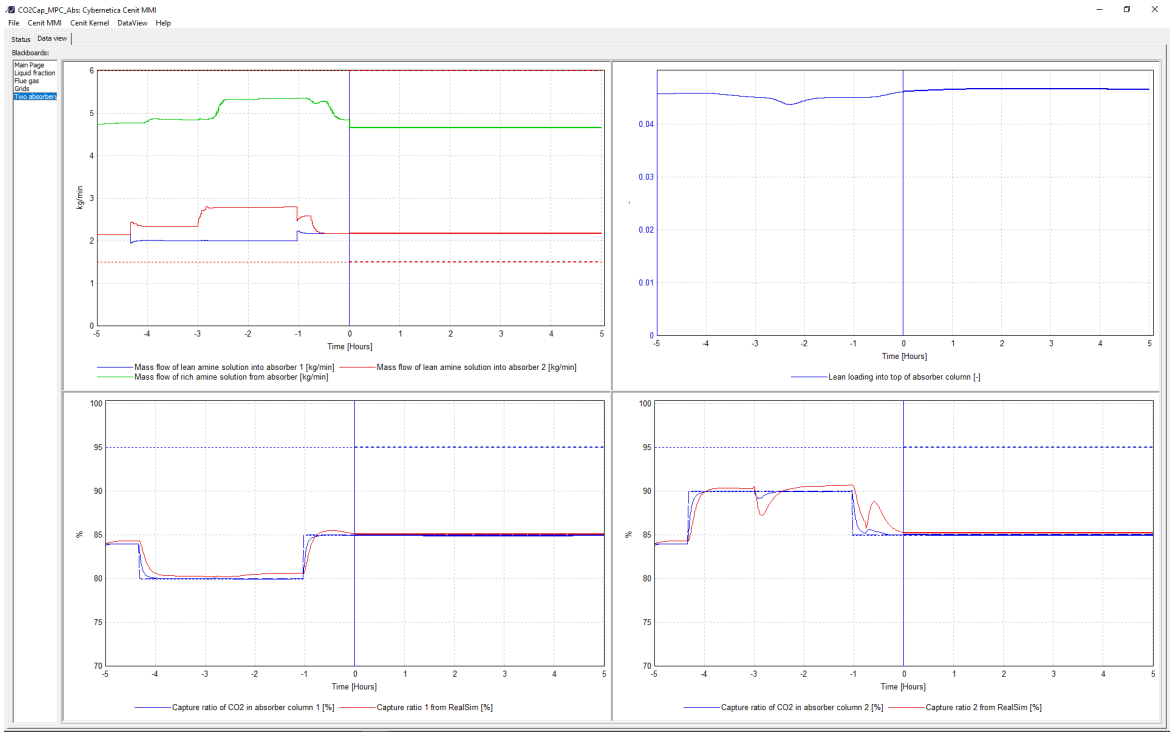



Figure B.1: The RealSim interface which is used to control the simulation




(a) The CENIT Kernel command line interface. This is the actual NMPC, but is not usually interacted with




(b) CENIT MMI is the interface for interacting with the NMPC

Figure B.2: The two components of CENIT

the RealSim interface looks like when having two NMPCs. In this particular instance there are two NMPCs, where NMPC des is run before NMPC abs. The simulation is controlled in the upper left part of the interface. Here the simulation can be started, stopped and run a given amount of samples. It is also possible to run one sample or one module at a time. The window in the upper center part of the interface shows an overview of the interfaces. It also shows the time used in the last sample, the mean time used per sample and the maximum and minimum time used for a sample. Through the menus on the left it is possible to see and change parameters and values relevant to the simulation.

Cybernetica CENIT is a software suite for NMPC which consists of two parts, CENIT Ker-

nel and CENIT MMI. Figure B.2a shows the CENIT kernel command line interface and Figure B.2b shows CENIT MMI. The CENIT Kernel is the actual NMPC, but is rarely interacted with. CENIT MMI on the other hand is made with the purpose of human interaction. A lot of information related to the controller is available in CENIT MMI. For instance states, measurements, controlled variables, manipulated variables and their predicted trajectories are available. It is also possible to change some settings in the NMPC, for instance the weights on setpoint violations. Communication between RealSim and CENIT goes through an open platform communications (OPC) server through tags. The tags were added to the Matrikon OPC server, shown in Figure B.3. The tags for parameters, inputs and measurements entering the CENIT controllers are in the folder CENIT/in. The tags for the MVs leaving the controller are in CENIT/out and the tags for telling the controllers to start and RealSim that they have finished are in RealSim/Manager.

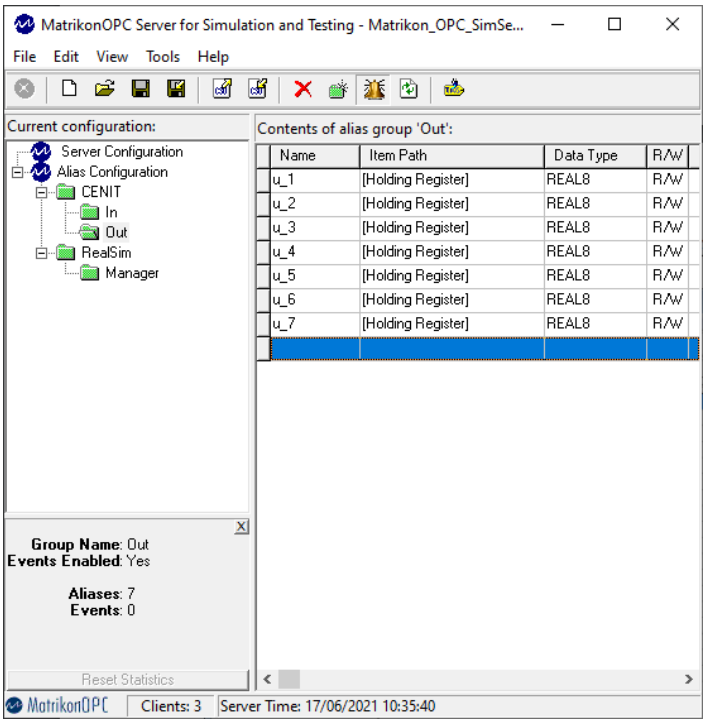



Figure B.3: The tags of the MVs in the Matrikon OPC server

Appendix C

# Exchanging information between NMPCs

## C.1 Exporting trajectories from NMPC des

To exchange information between the NMPCs data was stored in .mat files by NMPC des and the files were read by NMPC abs. First the dimension of the data structure used for exporting the data had to be defined. This is done by calling the function `ModelAppDataExchange_NmpcExportDim()`, shown in Code box C.1. In this function the variables `nFloat` and `nInt` are given their values. `pVar` is a data structure storing data related to the NMPC. `pVar->Z.nc` gives the number of columns in the matrix `pVar->Z.re`, which is a matrix storing the z-variables trajectories, where each row corresponds to a z-variable and the columns are predicted values increasingly far into the prediction horizon.

Code box C.1: C code for creating the exported data structure from NMPC des

```
ADXCH_API int ModelAppDataExchange_NmpcExportDim(
    int* nFloat,            // Out: Number of floats to export
    int* nInt,              // Out: Number of ints to export
    struct NmpcVar* pVar    // In:  Nmpc data structure
)
{
    int r = 0;

    // Number of floats to export
    *nFloat = 0;
    *nFloat += 1;           // Current value of z_LeanLoading
    *nFloat += pVar->Z.nc;  // z_LeanLoading trajectory
    *nFloat += 1;           // Current value of z_Fl_Lean_Abs_Mass
    *nFloat += pVar->Z.nc;  // z_Fl_Lean_Abs_Mass trajectory

    // Number of integers to export
    *nInt = 0;
    *nInt += 1;             // Synchronization of data (ok flag)
    *nInt += 1;             // z_LeanLoading setpoint trajectory
    *nInt += 1;             // z_Fl_Lean_Abs_Mass setpoint trajectory
```

```
21
22      return r;
23  }
```

After the dimensions of the datastructure had been defined, it had to be filled with data. This is done using the function `ModelAppDataExchange_NmpcExportData()` as shown in Code box C.2. The variable `dFloatData` is used to store the current value and predictions, `nIntData` is used to store the dimensions of the exported data. Both are written to the .mat file. `pVar->z.v` is a vector with the current values of the z-variables.
`pVar->z.v[1 + i_z_LeanLoading]` gives the current value of the z-variable `z_LeanLoading`, while `pVar->Z.re[1 + i_z_LeanLoading][i]` gives the predicted value of `z_LeanLoading` in column `i`. Similar for `z_Fl_Lean_Abs_Mass`. The for loops fill `dFloatData` with predicted values increasingly far into the prediction horizon.

Code box C.2: C code for filling the datastructure with data from NMPC des

```
1   ADXCH_API int ModelAppDataExchange_NmpcExportData(
2       double* dFloatData,             // Out: Float data to export
3       int* nIntData,                  // Out: Integer data to export
4       int nFloat,                     // In:  Number of floats to export
5       int nInt,                       // In:  Number of ints to export
6       struct NmpcVar* pVar            // In:  Nmpc data structure
7   )
8   {
9       int i, j, k, r = 0;
10      int DataIsOK = 1;
11
12      // Export setpoint trajectory for z_LeanLoading
13      j = 0;
14      nIntData[1] = 1 + pVar->Z.nc;
15      dFloatData[j] = pVar->z.v[1 + i_z_LeanLoading];
16      for (i = 1; i <= pVar->Z.nc; i++) {
17          k = j + i;
18          dFloatData[k] = pVar->Z.re[1 + i_z_LeanLoading][i];
19      }
20
21      // Export setpoint trajectory for z_Fl_Lean_Abs_Mass
22      j += nIntData[1];
23      nIntData[2] = 1 + pVar->Z.nc;
24      dFloatData[j] = pVar->z.v[1 + i_z_Fl_Lean_Abs_Mass];
25      for (i = 1; i <= pVar->Z.nc; i++) {
26          k = j + i;
27          dFloatData[k] = pVar->Z.re[1 + i_z_Fl_Lean_Abs_Mass][i];
28      }
29
30      // Set ID/sync
31      nIntData[0] = DataIsOK;
32      pVar->ConVar.v[1 + i_cv_RTO_ExportSuccessful] = DataIsOK;
33
34      return r;
35  }
```

## C.2 Importing data in NMPC abs

NMPC abs has to read the values stored in the .mat file by NMPC des and store them in a data structure which can be accessed elsewhere in the code. The definition of the data structure is given in Code box C.3.

Code box C.3: Type definition for data exchange between NMPC des and NMPC abs

```
// TYPES FOR DATA EXCHANGE BETWEEN MPC_ABS AND MPC_DES
typedef struct sImportData_MPC {
    nrVEC LeanLoading;          // Vector of predicted lean loading
    nrVEC Fl_Lean_Abs_Mass;     // Vector of predicted lean flow
    int dataAvailable;          // Flag to indicate whether the data
                                // is usable or not     [1/0]
    double timeOffset;          // Time offset between point in time
                                // when trajectories were generated
                                // vs. when they were loaded [s]
} ImportData_MPC;
ImportData_MPC MPCData;
```

The values of `MPCData` are filled in using the function `ModelAppDataExchange_NmpcImportData()` shown in Code box C.4. First there are some conditions which have to be met to ensure that the data is fine to use. If the data is acceptable the flag in `NMPCData` will be set equal to 1 and the variables are filled with the trajectories and time offset.

Code box C.4: C code for filling the datastructure with data in NMPC abs

```
ADXCH_API int ModelAppDataExchange_NmpcImportData(
    double* dFloatData,     // In: Float data from external application
    int* nIntData,          // In: Integer data from external application
    int nFloat,             // In: Number of floats to export
    int nInt,               // In: Number of ints to export
    double timeImportData,  // In: Time stamp of imported data
    double timeCurrent,     // In: Time stamp for current application
    int nID,                // In: Import id
    struct NmpcVar* pVar    // In: Nmpc data structure
)
{
    int i, j, r = 0;
    int DataIsOK;

    // Do not update if current time stamp older than imported data
    // (happens due to simulated time)
    if (timeCurrent - timeImportData < 0.0) {
        MPCData.dataAvailable = 0;
        EventLog(EvTypeWarning, 0, __FUNCTION__,
            "MPC ABS data from file has invalid timestamp", pEvLog);
        return r;
    }

    // Check if the data is marked as okay
    DataIsOK = nIntData[0];
```

```
26      if (DataIsOK == 0) {        // The loaded data is not ok,
27                                  // according to the file
28          MPCData.dataAvailable = 0;
29          EventLog(EvTypeWarning, 0, __FUNCTION__,
30              "MPC ABS data from file not marked as ok", pEvLog);
31          return r;
32      }
33
34      // If data is okay, copy it to the NMPC data struct
35      else {
36          MPCData.dataAvailable = 1;
37          pVar->ConVar.v[1 + i_cv_MPC_ImportSuccessful] = 1;
38          // Set time offset
39          MPCData.timeOffset = timeCurrent - timeImportData;
40
41          // Obtain setpoint trajectory for z_btl_nCO2
42          j = 0;
43          for (i = 0; i < nIntData[1]; i++) {
44              MPCData.LeanLoading.v[1 + i] = dFloatData[j + i];
45          }
46
47          // Obtain setpoint trajectory for z_Fl_Lean_Abs_Mass
48          j += nIntData[1];
49          for (i = 0; i < nIntData[2]; i++) {
50              MPCData.Fl_Lean_Abs_Mass.v[1 + i] = dFloatData[j + i];
51          }
52      }
53      return r;
54  }
```

The code in Code box C.5 shows how the reference trajectory from NMPC des is implemented as reference trajectory in NMPC abs, in this case it is the trajectory of lean mass flow. `interp_linear()` is a function written by Fredrik Gjertsen at Cybernetica AS, which performs linear interpolation. This is necessary if the two controllers have different parameterisation of the prediction horizon. `Zref` is a matrix structure which is used to store the reference trajectories of all the z-variables.

Code box C.5: C code for defining the reference trajectory of `z_Fl_Lean_Abs_Mass` with data imported from NMPC des

```
1   if (i == i_z_Fl_Lean_Abs_Mass) {
2       // Trajectory for z_Fl_Lean_Abs_Mass
3       for (j = 1; j <= Zref->nc; j++) {
4           Zref->re[1 + i][j] = interp_linear((j*NDT_MPC*DT +
5               MPCData.timeOffset), &(PredHorSamplesTime.v[1]),
6               &(MPCData.Fl_Lean_Abs_Mass.v[1]), PredHorSamplesTime.n);
7       }
8   }
```

After the reference trajectory was defined the constraints had to be implemented as well. This was done as shown in Code box C.6. The reference trajectory is used to create the constraints. The minimum constraint is the reference trajectory minus $0.1\,\mathrm{kg\,min^{-1}}$ while

the maximum constraint is the reference trajectory plus $0.1\,\mathrm{kg\,min^{-1}}$.

Code box C.6: C code for defining the maximum and mimimum constraints of `z_Fl_Lean_Abs_Mass` using the reference trajectory from NMPC des

```
if (i == i_z_Fl_Lean_Abs_Mass) {
    /* Set Zmin             */
    for (j = 1; j <= Zmin->nc; j++)
        Zmin->re[1 + i][j] = Zref->re[1 + i][j] - 0.1;
    /* Set Zmax             */
    for (j = 1; j <= Zmax->nc; j++)
        Zmax->re[1 + i][j] = Zref->re[1 + i][j] + 0.1;
}
```

Appendix D

# Model code

## D.1 Step response model

Code box D.1 shows the code for the step response models with bias updating of the rich loading model. `dxs_RichLoading_as_1` is the derivative of the state `xs_RichLoading_as_1`. It is similar for the three other states where the derivatives are `dxs_RichLoading_as_2`, `dxs_RichFlow_as_1` and `dxs_RichFlow_as_2`, and the states are `xs_RichLoading_as_2`, `xs_RichFlow_as_1` and `xs_RichFlow_as_2`.

Code box D.1: C code for the step response models

```
int model_dx_step_response(     // Ret: Error code (0 = OK!)
    double *dxs,                // Out: State vector derivative
    double *xs,                 // In:  State vector
    double *par,                // In:  Parameter vector
    double *con,                // In:  Constant vector
    double *U_INT,              // In:  Internal process input
    double *au,                 // In:  Vector of additional/calculated inputs
    double *dcalcvar,           // In:  Internal variable calculated by model
    int    *icalcvar,           // In:  Internal variable calculated by model
    double  dti,                // In:  Integration time step
    double *v,                  // In:  Process disturbance vector
    int     intf                // In:  ID of calling interface
    )
{
    int r = 0;
    dxs_RichLoading_as_1    =   1.0 / ( 60.0 * 17.5 ) * (-0.0461 *
                                (U_LeanFlow_des_1 * 60 - 2.5) -
                                (xs_RichLoading_as_1 - 0.62719));
    dxs_RichFlow_as_1       =   1.0 / ( 60.0 * 10   )*((1.0004 + 0.4 *
                                dc_Bias_RichFlow_1) * (U_LeanFlow_des_1 * 60.0 -
                                2.5) - (xs_RichFlow_as_1 - 2.74046));
    dxs_RichLoading_as_2    =   1.0 / ( 60.0 * 17.5 ) * (-0.0461 *
                                (U_LeanFlow_des_2 * 60 - 2.5) -
                                (xs_RichLoading_as_2 - 0.62719));
    dxs_RichFlow_as_2       =   1.0 / ( 60.0 * 10   )*((1.0004 + 0.4 *
```

```
                                dc_Bias_RichFlow_2)*(U_LeanFlow_des_2 * 60.0 -
                                2.5) - (xs_RichFlow_as_2 - 2.74046));
    return r;
}
```

## D.2 Mole balance model

The mole balance model was used for predicting rich flow and rich loading and for calculating the capture ratio. The code for each case is given here.

### D.2.1 Predicting rich flow and rich loading

The code for predicting rich flow and rich loading with the mole balance model is given in Code box D.2.

Code box D.2: C code for calculating rich flow and rich loading based on mole balances

```
void MB_model(
    double *rl,                 // Out: Rich loading
    double *rf,                 // Out: Rich flow
    double *xs,                 // In:  State vector
    double *par,                // In:  Parameter vector
    double *con,                // In:  Constant vector
    double *U_INT,              // In:  Internal process input
    int     abs_1               // In:  Flag to set wether it is column 1 or 2
)
{
    double V0, L0, L1, F, nTotV1, xL[NC_LIQ], L1_vec[NC_LIQ], L0_vec[NC_LIQ];

    double InletFlow_Flue, Fl_Lean_Abs_Mass, CR, yG_CO2;
    if (abs_1) {
        InletFlow_Flue      =   U_InletFlow_Flue;
        Fl_Lean_Abs_Mass    =   U_LeanFlow_des_1;
        CR                  =   U_CR_1/100.0;
        yG_CO2              =   U_yG_CO2_Flue_in;
    }
    else {
        InletFlow_Flue      =   U_InletFlow_Flue_2;
        Fl_Lean_Abs_Mass    =   U_LeanFlow_des_2;
        CR                  =   U_CR_2/100.0;
        yG_CO2              =   U_yG_CO2_Flue_in_2;
    }

    // V0 and F
    V0                      =   U_InletPress_Flue * InletFlow_Flue /
                                (RGAS * U_InletTemp_Flue);      // [kmol/s]
    F                       =   CR * yG_CO2 * V0;               // [kmol CO2/s]

    // L1
    nTotV1                  =   (xs_btl_nCO2 + xs_btl_nH2O + xs_btl_nAmine);
    xL[i_CO2]               =   xs_btl_nCO2 / nTotV1;
    xL[i_H2O]               =   xs_btl_nH2O / nTotV1;
```

```
    xL[i_AMINE]             =   xs_btl_nAmine / nTotV1;
    L1                      =   Fl_Lean_Abs_Mass / (xL[i_CO2] * MW_CO2 +
                                xL[i_H2O] * MW_H2O + xL[i_AMINE] * MW_AMINE);
                                // [kmol/s]
    L1_vec[i_CO2]           =   xL[i_CO2] * L1;
    L1_vec[i_H2O]           =   xL[i_H2O] * L1;
    L1_vec[i_AMINE]         =   xL[i_AMINE] * L1;

    // L0
    L0_vec[i_CO2]           =   L1_vec[i_CO2] + F;
    L0_vec[i_H2O]           =   L1_vec[i_H2O];
    L0_vec[i_AMINE]         =   L1_vec[i_AMINE];

    double MWs[]            =   { MW_CO2, MW_H2O, MW_AMINE };

    L0                      =   0;
    for (int i = 0; i < NC_LIQ; i++) L0 += L0_vec[i] * MWs[i];

    *rl                     =   L0_vec[i_CO2] / L0_vec[i_AMINE];
    *rf                     =   L0;
}
```

The code for calculating the derivatives of the rich flow and rich loading states is given in Code box D.3.

Code box D.3: C code for the mole balance model

```
int model_dx_MB_model(      // Ret: Error code (0 = OK!)
    double *dxs,                // Out: State vector derivative
    double *xs,                 // In:  State vector
    double *par,                // In:  Parameter vector
    double *con,                // In:  Constant vector
    double *U_INT,              // In:  Internal process input
    double *au,                 // In:  Vector of additional/calculated inputs
    double *dcalcvar,           // In:  Internal variable calculated by model
    int    *icalcvar,           // In:  Internal variable calculated by model
    double  dti,                // In:  Integration time step
    double *v,                  // In:  Process disturbance vector
    int     intf                // In:  ID of calling interface
    )
{
    int r = 0;
    double RichLoading_1, RichFlow_1, RichLoading_2, RichFlow_2, tauRL, tauRF;

    MB_model(&RichLoading_1, &RichFlow_1, xs, par, con, U_INT, 1);
    MB_model(&RichLoading_2, &RichFlow_2, xs, par, con, U_INT, 0);

    tauRL = 18.0;           // Time constant for rich loading
    tauRF = 10.0;           // Time constant for rich flow

    dxs_RichLoading_as_1    =   1 / (60.0 * tauRL) * (RichLoading_1
                                - xs_RichLoading_as_1);
    dxs_RichFlow_as_1       =   1 / (60.0 * tauRF) * (RichFlow_1 * 60.0 -
                                xs_RichFlow_as_1);
```

```
28
29      dxs_RichLoading_as_2    =   1 / (60.0 * tauRL) * (RichLoading_2
30                                  - xs_RichLoading_as_2);
31      dxs_RichFlow_as_2       =   1 / (60.0 * tauRF) * (RichFlow_2 * 60.0
32                                  - xs_RichFlow_as_2);
33      return r;
34  }
```

### D.2.2 Calculating capture ratio

When the mole balance model is used for calculating the capture ratio in absorber column 1 the code is as shown in Code box D.4.

Code box D.4: C code for calculating the capture ratio based on mole balances

```
1   double R, L, F, nTotV1, xL[NC_LIQ], L1, xCO2, R1, xR[NC_LIQ];
2
3   F               =   U_yG_CO2_Flue_in * U_InletPress_Flue * U_InletFlow_Flue
4                       / (RGAS * U_InletTemp_Flue);            // [kmol CO2/s]
5   nTotV1          =   (xs_btl_nCO2 + xs_btl_nH2O + xs_btl_nAmine);
6   xL[i_CO2]       =   xs_btl_nCO2 / nTotV1;
7   xL[i_H2O]       =   xs_btl_nH2O / nTotV1;
8   xL[i_AMINE]     =   xs_btl_nAmine / nTotV1;
9   L1              =   U_Fl_Lean_Abs_Mass / (xL[i_CO2] * MW_CO2 + xL[i_H2O]
10                      * MW_H2O + xL[i_AMINE] * MW_AMINE);
11      L           =   xL[i_CO2] * L1;                         // [kmol/s]
12
13  xR[i_CO2]       =   xs_RichLoading_as_1 / (1.0 + xs_RichLoading_as_1
14                      + xs_btl_nH2O / xs_btl_nAmine);
15  xR[i_H2O]       =   xs_btl_nH2O / xs_btl_nAmine / (1.0 +
16                      xs_RichLoading_as_1 + xs_btl_nH2O / xs_btl_nAmine);
17  xR[i_AMINE]     =   1.0 / (1.0 + xs_RichLoading_as_1 + xs_btl_nH2O
18                      / xs_btl_nAmine);
19
20  R1              =   xs_RichFlow_as_1 / 60.0 / (xR[i_CO2] * MW_CO2
21                      + xR[i_H2O] * MW_H2O + xR[i_AMINE] * MW_AMINE);
22  R               =   xR[i_CO2] * R1;                 // [kmol/s]
23
24  z_CaptureRatio_1 = 100.0 * (R - L) / F;
```