CONFIDENTIAL

Switched Linear MPC to Minimize the Power Consumption of a Portable Oxygen Concentrator

Nathan Looye

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

MSCCONFIDENTIAL

Switched Linear MPC to Minimize the Power Consumption of a Portable Oxygen Concentrator

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Nathan Looye

April 9, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Philips. Their coorperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Switched Linear MPC to Minimize the Power Consumption of a Portable Oxygen Concentrator

by

Nathan Looye

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: April 9, 2020

Supervisor(s):

Dr.ir. E. Steur

Dr.ir. D.J.H. Bruijnen (Philips Innovation Services)

Reader(s):

Prof.dr.ir. B. de Schutter

Dr. M. Mazo Espinosa

Abstract

Portable Oxygen Concentrators (POCs) are devices that produce oxygen-enriched air, by selectively filtering nitrogen out of ambient air with a cyclic process called Pressure Swing Adsorption (PSA). The current control method is to adjust the timings of the process by means of lookup tables, such that the POC operates as efficient as possible. The aim of this thesis is to determine whether Model Predictive Control (MPC) is a viable alternative to control the POC, and is able to cope with the constraints and variations of the system. First, a high-fidelity model has been made of the POC, used for simulation of the device and controller design. Comparisons with other suitable models of POCs have shown that the dynamics inside the POC have been modeled correctly. Because this model is too complex to serve as a predictive model, a simplified batch model has been created for that purpose. This hybrid automaton consists of 13 linear models, and encompasses the cycle-to-cycle dynamics of the plant. Finally, a switched linear MPC strategy has been designed and implemented on the high-fidelity model. Simulations show that this control strategy is suited to control the POC, although further research is needed to cope with system degradation.

Table of Contents

List of Figures

List of Tables

Preface

This thesis marks the end of my journey in obtaining a master's degree in Systems & Control at the Delft University of Technology. This would have not been possible without the help of the following people, which I would like to thank.

I would like to express my gratitude to my two supervisors, Dennis Bruijnen and Erik Steur, for the fruitful discussions and constructive feedback during the course of my graduation internship at Philips. Also, thanks to my fellow interns Esau and Rens, for the company and advice during my stay at Philips.

And finally, I would like to thank Anna Roseboom, my family, and my friends in Delft and Eindhoven for supporting me and giving me the motivation to finish this thesis.

Nathan Looye
Delft, april 9 2020

"The only difference between science and screwing around is writing it down."
— *Adam Savage*

Chapter 1

Introduction

Oxygen therapy is the use of supplemental oxygen as a medical treatment. For people suffering from e.g., Chronic Obstructive Pulmonary Disease (COPD), this treatment greatly improves their quality of life [4]. Its use is widespread in the world; in 2008 it was estimated 800.000 people in the USA received long-term oxygen therapy [5]. One method of providing oxygen is with an oxygen concentrator: these devices selectively remove nitrogen from the air using a process called Pressure Swing Adsorption (PSA), to supply oxygen-rich air (ca. 90% oxygen) to the user [6, 7]. These devices do not use compressed gas storage like regular O_2 cylinders, which makes them safer to use at home.

PSA, the process used to separate nitrogen and oxygen, is based on adsorption selectivity [8]. An oxygen concentrator works as follows: compressed air is pumped into sieve beds, cylinders filed with a material that adsorbs nitrogen. As a result, nitrogen is adsorbed and oxygen flows through into an oxygen tank, from where the desired flow of oxygen-enriched air is delivered to the user.

A growing market is that of Portable Oxygen Concentrators (POCs), which unlike conventional oxygen concentrators, carry their own power supply. This means users are no longer confined to their stationary source of oxygen, giving them more mobility. POCs have even been approved by the FAA for on-board aircraft usage [9]. These developments have led to the desire for lightweight, long-lasting and efficient devices.

1-1 Research Context

This research is aimed towards reducing the power consumption of a POC, by increasing its efficiency. This way the device could either be used for a longer period after a battery charge, or smaller batteries can be used, reducing the weight of the POC. Possible ways to increase the efficiency are e.g., improving the batteries or the compressor (the main power user [10]), but this requires a change of hardware. Another option is to improve the efficiency of the PSA process inside the POC. The PSA process is typically controlled in open-loop, where the (valve) timings of the system, which control the flow of air in the POC, are calibrated

in the factory. System degradation needs to be taken into account, resulting in conservative timings and consequently a reduced efficiency.

A better (more advanced) control strategy might be better suited to handle this system degradation, and therefore control the POC more efficiently. In this thesis, the feasibility of applying Model Predictive Control (MPC) as a control method for the POC will be investigated. MPC is based on finite-horizon optimization of a plant model: every time step an optimal control move is calculated by minimizing a suitable cost function, possibly subject to constraints [11]. There are multiple reasons why MPC could be an improvement over the current control method. First of all, if the system degradation can be captured in the predictive model for the controller, MPC is able to predict this system behavior and adjust the timings accordingly. Secondly, because MPC uses instantaneous feedback, it is more robust to (small) disturbances (such as system degradation), compared to open-loop strategies. If these disturbances become more severe, it is also possible to use hybrid- or adaptive MPC to ensure adequate performance.

Another argument for MPC, not related to degradation, is its ability to deal with constraints. The objective of the controller is to minimize the power consumption, but there are constraints on the flow of oxygen-enriched air to the user: one, a certain flow rate is needed, and two, a minimum oxygen concentration of this air is required.

Because of these two constraints (flow rate and O_2 concentration) on the outgoing air, a minimum flow rate of ambient air to the compressor is needed. This in turn sets a lower limit on the power needed for the compressor, i.e. minimizing the power consumption is only possible to a certain extent. Therefore, the efficiency of the PSA process becomes important: the more efficient, the less air is needed. Increasing the efficiency of a PSA process is done by operating the sieve beds as close to their saturation capacities as possible, which is a challenge, because these capacities change over time due to sieve bed degradation [12]. So, in order to optimize the operation of a POC, the controller should have knowledge of these system dynamics.

To limit the scope of this research, sieve bed degradation will not be incorporated into the controller (i.e. the predictive model); the main focus will be to optimize the power consumption of the POC under normal operating conditions. However, a case study on sieve bed degradation will also be done, in which the POC will be simulated with degraded sieve beds. In this case study, the performance of the derived MPC strategy will be compared with open-loop strategies, and with the case without degradation. After that, an assessment can be made whether further development of the MPC strategy is required to handle sieve bed degradation.

For this purpose, two models are developed. One will act as the plant (the POC), and is used for simulation, data generation and control verification. The other is the predictive model, and is used by the MPC strategy.

1-2 Research Objective

The goal of the research in this thesis is to develop an MPC-based control method that minimizes the power consumption of a POC, subject to constraints on the flow of oxygen-enriched air to the user. To (further) limit the scope of this research, some assumptions

and simplifications will be defined, of which one has already been discussed in the previous section. These are:

- Sieve bed degradation will not be incorporated into the predictive model of the controller (like discussed in Section 1-1).
- Only the power consumption of the compressor, the main power consumer, will be considered.
- The molar flow rate of air to the user and to the compressor are both constant, leaving the timings of the process as the inputs to the system.

Given these assumptions, the main research objective is the following:

> *Develop a model predictive controller that minimizes the power consumption of the POC, while keeping the oxygen concentration in the tank above a certain limit.*

To do so, sub-objectives will also be defined. First of all, a dynamic model is needed to simulate the behavior of the POC. This model needs to capture the intricate dynamics that make up a PSA process. Secondly, the new controller should ideally work on the existing hardware of the POC. This limits the complexity of the controller and the predictive model that is used for optimization. Thus, these two sub-objectives are to:

1. *Develop a model that is capable of accurately simulating the dynamics of a POC.*
2. *Develop a simplified predictive model for the controller, that is simple enough such that it could be implemented on a POC.*

1-3 Main Contributions

The research in this thesis has the following main contributions:

- Design and implementation of a high-fidelity model of the POC.
  A first-principles based high-fidelity model has been created to simulate the POC in the absence of the real device. Comparisons with other studies have shown that the important dynamics of the PSA process are modeled correctly.
- Design and implementation of a simplified (predictive) batch model.
  As a predictive model for MPC, a hybrid linear model has been identified that captures the cycle-to-cycle dynamics of the POC.
- Development of a MPC strategy to control the POC.
  This thesis shows that a POC can be controlled by MPC. Although detailed modeling and optimization is applied, the MPC strategy can be improved further.

1-4 Thesis Outline

The outline of this thesis is as follows. First, in Chapter 2, some background information will be given, needed for the modeling and control of a POC. A schematic overview of the device is shown, and its components will be discussed. Thereafter, the process behind the separation of nitrogen and oxygen, PSA, is explained. Next, Chapter 3 will discuss the high-fidelity modeling of the POC. In the absence of a real POC, this first principles model will be used to simulate the plant, and design the controller. In Chapter 4 the derived high-fidelity model is simulated in open-loop, to show the dynamics of the POC. Comparable studies in the literature are used to validate the model.

Chapter 5 will discuss the development of a so-called batch model. A hybrid linear framework is used to model the cycle-to-cycle dynamics of the plant, and this model is used as a predictive model for the controller. Then, in Chapter 6, the MPC strategy is derived. Cost functions will be defined, and the design choices will be motivated. Chapter 7 will then discuss the simulation results of the controller, and in Chapter 8 the results of this thesis are summarized, the research objective will be reviewed, and recommendations for future work will be provided.

Chapter 2

Background

This chapter covers the necessary background needed for the modeling and control of a Portable Oxygen Concentrator (POC). The first section is about the device itself: its components will be discussed, as well as the current control method. In Section 2-2, the chemical process behind the purification of oxygen will be explained: Pressure Swing Adsorption (PSA).

2-1 Portable Oxygen Concentrators

A POC is a device that provides air with greater oxygen content (ca. 90%) than ambient levels by filtering out the nitrogen in the air. Unlike their stationary counterparts, which have external power supplies, POCs are equipped with batteries. This, in combination with their relatively low size and weight, means that users are able to carry their devices with them, providing mobility.

The setup of a POC is best explained by a schematic overview, see Figure 2-1. Air, consisting of 78% nitrogen, 21% oxygen and trace amounts of other gases (mostly argon), is fed into the system via the compressor, where it is pressurized. From here, pressurized air flows to the sieve beds, where the nitrogen is extracted from the air. Oxygen-rich air exits these sieve beds, and flows to the oxygen tank. From there it can be fed to the user at the desired flow rate. The oxygen tank acts as a buffer, because there are moments there is no oxygen production from either sieve bed. This way a continuous flow of oxygen is still achieved.

The two sieve beds (or adsorbers) are the lungs of the POC. These cylinders are filled with a material that adsorbs nitrogen. By pumping compressed air through a sieve bed, nitrogen is captured in the adsorber and oxygen-enriched air flows out the adsorber. They can get saturated with nitrogen, and that is why two sieve beds are used: if one is producing oxygen, the other is regenerating, i.e. exhausting the nitrogen. This will be explained in-depth in Section 2-2.

There are three different kinds of valves inside the device, all used to control the direction of the airflow. Some are passive, like the orifice and the check valves. The latter are used to

prevent oxygen-rich air from flowing back into the sieve beds from the oxygen tank, and the need for the orifice will be explained in Section 2-2-3. The active valves, together with the RPM of the compressor, are the inputs of the system and will be discussed in Section 2-1-1.

There are two kinds of POCs: devices that deliver a pulsed flow of oxygen-enriched air, and devices that deliver a continuous flow. The POC in this thesis delivers a continuous flow, and recalling Section 1-2, this flow will be constant, since also manipulating this variable is not in the scope of this research. Therefore a valve controlling the flow of oxygen to the user has been left out of the model.

One component that is not described in Figure 2-1 is the battery. Although the goal of this study is to minimize the power consumption, including the battery is also not in the scope of this thesis. It could be included in future improvements of this model, but for now the focus is on modeling and subsequently controlling the other components of the POC.

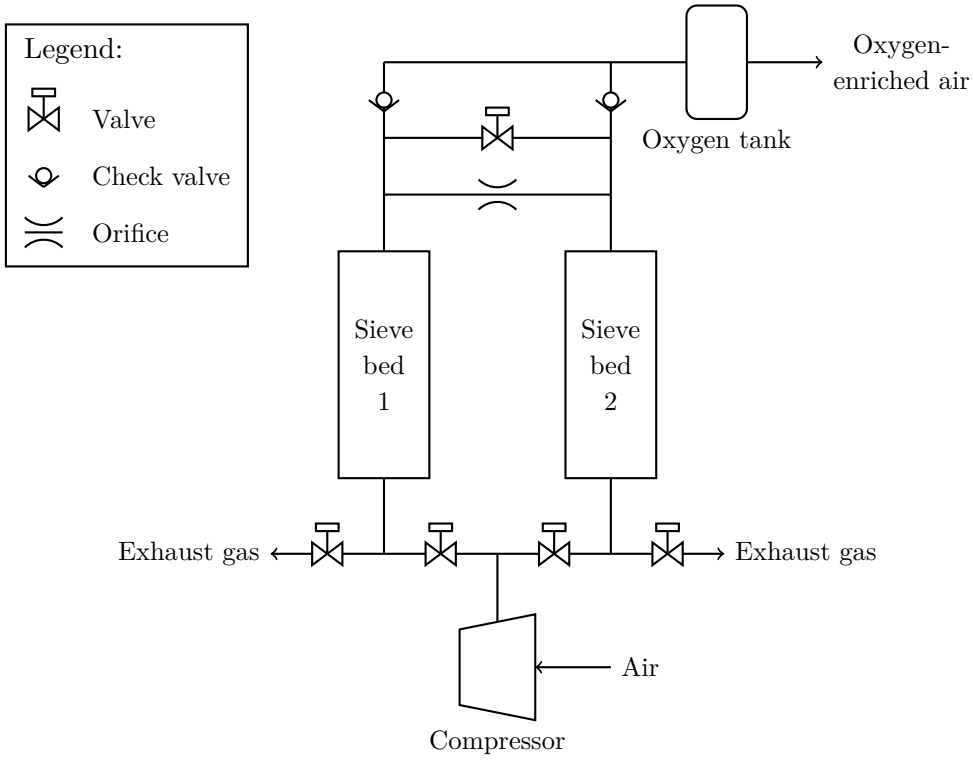

Figure 2-1: Schematic overview of the POC.

2-1-1 Current Control Method

Like said before, there are two types of inputs: the valves and the RPM of the compressor. The main purpose of the controller is to keep the energy consumption of the POC to a minimum, whilst keeping the oxygen concentration of the air flow to the user above a certain limit. It is important that a sieve bed is exhausted before it is completely saturated: otherwise nitrogen will leak through into the oxygen tank. This phenomenon is called nitrogen breakthrough. As a consequence, the oxygen concentration in the tank can drop below the minimal allowable value. To prevent this from happening, the valves are used to change the flow in the system:

by carefully timing the switching of valves, the sieve beds are utilized to their maximum. The other input, the compressor RPM, is used to change the flow of air into the system when needed. However, the dynamical model described in this thesis will only use the valves as input. This has been done to reduce the complexity of the system.

A physical POC has a limited amount of sensors, for economical reasons. Usually, the flow rate and the oxygen purity are measured, but any states like pressure or oxygen concentration in the sieve beds are not. For the dynamical model, another assumption is made, namely that all states (e.g., pressure, oxygen concentration) inside the system are measurable.

The current method of controlling the POC is to tune the valve timings based on factory calibration. Over time, small adjustments to these timings are needed, due to the degradation of the sieve beds (see Section 2-2-4). Determining these adjustments is done by using lookup tables, and the hardware on the POC is tailored for this.

2-2 Pressure Swing Adsorption

The process used inside the POC to extract the nitrogen from the air is called pressure swing adsorption. PSA is a cyclic process, widely used in the industry to separate or purify gasses [6, 8]. It is based on the affinity of specific gasses to an adsorbent material, which highly depends on pressure. By increasing the pressure in a sieve bed, more and more of the target gas (in this case nitrogen) is adsorbed, whereas oxygen is largely unaffected and can flow through the sieve bed. If the pressure is then reduced, the adsorption capacity of the adsorbent is also reduced, nitrogen is released and flows out of the other end of the sieve bed, meaning the sieve bed is effectively regenerated. PSA is not only used for O_2, N_2 separation; other usages are e.g., industrial separation of H_2 and CH_4 [13], or the capture of CO_2 [14].

In order to produce the near-continuous stream of oxygen, two adsorbers are used in parallel. In the first half of a cycle, the first adsorber is pressurized such that it produces oxygen, and the second adsorber is depressurized, exhausting the nitrogen. During the second half of the cycle the adsorbers switch roles, and after one cycle the first adsorber begins pressurization again. This so-called PSA cycle will be discussed in Section 2-2-3.

2-2-1 Adsorption

Before the PSA cycle is explained, the phenomenon of adsorption will be discussed. Adsorption is the name of the attraction that a molecule (adsorbate) experiences when it is close to the surface of a solid (adsorbent) [8].

The separation of a gas mixture by adsorption depends on both equilibrium and kinetic factors, but the relative importance varies greatly for different systems [15]. Kinetic separation is achieved by virtue of the differences in diffusion rates of different molecules, while equilibrium-based separation is based on differences in equilibrium affinity, i.e. preferential adsorption of certain components. The adsorbent used in this POC is a zeolite, and in this material diffusion of both oxygen and nitrogen is rapid and thus the separation depends on the preferential (equilibrium) adsorption of nitrogen [16].

Zeolites are micro-porous, crystalline minerals that are well suited as adsorbent; they are highly porous and therefore have a high surface-to-volume ratio. The zeolite itself is produced

as a fine powder, where each grain is crystalline structure made up of zeolite molecules joined together. These structures are then held together by a porous clay binder material to form pellets, which consist of 80% zeolite crystals and 20% binder material. Both sieve beds are filled with these pellets. The reason for this bound form is to reduce the pressure drop in the column [1].

Adsorption Equilibrium

An important aspect in the design and modeling of adsorption processes is knowledge of the adsorption equilibrium. This equilibrium mathematically defines the relationship between the concentration of a given component in the gas phase with its concentration in the adsorbed phase, and depends on pressure, temperature and phase composition [8]. These equilibria are graphically depicted as isotherms, showing the maximal adsorbed phase concentration (amount adsorbed per mass adsorbent) at various pressures. In Figure 2-2 isotherms of some zeolites are shown. The greater the adsorption capacity difference for each gas, the greater the driving force to separate those gasses. Different models exist that can represent these isotherms. This will be discussed in Chapter 3.

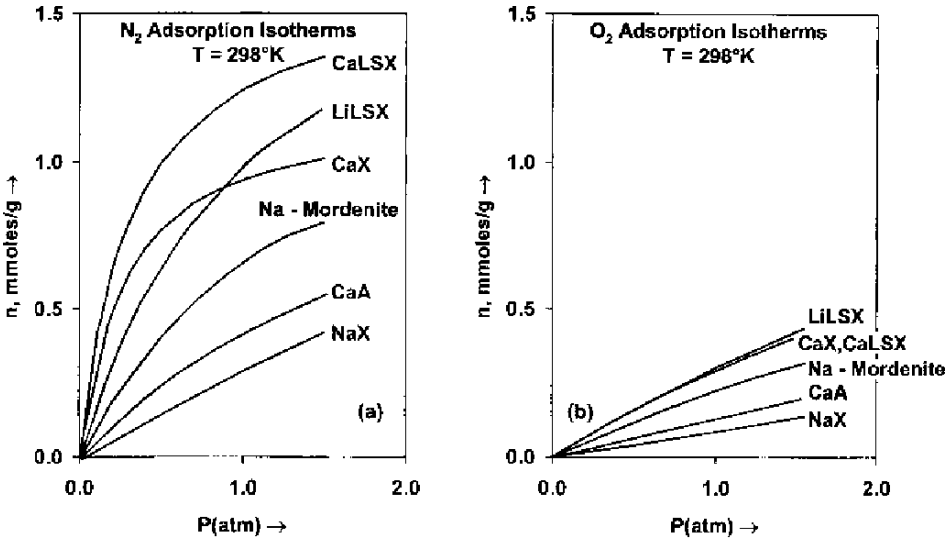

Figure 2-2: Adsorption isotherms of nitrogen (left) and oxygen (right) on various zeolites at 25°C. Retrieved from [1].

2-2-2 Sieve Bed Behavior

Sieve beds have already been mentioned in this chapter as the zeolite-filled cylinders where nitrogen is adsorbed. In this section, the adsorption dynamics inside the sieve beds will be discussed. A sieve bed has a feed end, closest to the compressor, and a product end, closest to the oxygen tank. The variable z will be used to denote the axial location in the sieve bed with $z = 0$ the feed end and $z = L$ the product end.

The (ideal) adsorption and desorption dynamics inside a sieve bed can be seen as traveling concentration waves. During production, compressed air enters the column at the feed end,

and nitrogen is adsorbed. The zeolite closest to the feed end gets saturated first, since it is first to come in contract with the nitrogen. As time continues, more and more zeolite gets saturated, and this adsorption front propagates through the sieve bed. A widely used name for this front is the Mass Transfer Zone (MTZ) [8], and in Figure 2-3 a visual representation of the MTZ can be seen.

If the feed is not stopped before the leading end of the MTZ reaches the end of the sieve bed, adsorbate (nitrogen) leaks through. The result is a decrease in oxygen concentration of the air leaving the sieve bed, and eventually also a decrease of oxygen concentration of the air going to the user. This should obviously be avoided. On the other hand, if the feed is stopped prematurely, some unused bed will remain containing oxygen-enriched air. This lowers the efficiency of the process. It is therefore paramount to detect this MTZ and adjust the feed accordingly for optimal PSA process design.

The dynamics of a sieve bed depend on the adsorption equilibrium, adsorption kinetics and fluid dynamics. Modeling of these sieve beds, among the other components, will be discussed in the next chapter.

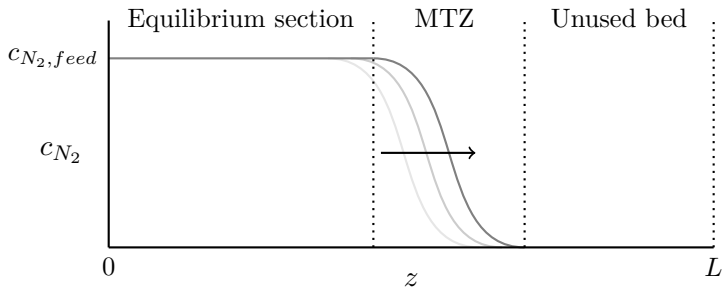

Figure 2-3: Manifestation of the concentration waves inside a sieve bed. Nitrogen enters the sieve bed at $z = 0$ with a concentration $c_{N_2} = c_{N_2,feed}$, and in the MTZ the nitrogen is adsorbed such that $c_{N_2} \to 0$. This MTZ propagates from $z = 0$ to $z = L$ or vice versa, depending on the current PSA cycle step. Adapted from [8].

2-2-3 Cycle Steps

A PSA cycle can be divided into a number of steps. The PSA cycle of the POC is based on a modified Skarstrom cycle [17], and consists of four steps: feed, pressure equalization I, exhaust, and pressure equalization II. Each sieve bed undergoes these four steps in a cyclic manner, and in an alternating way: if sieve bed 1 is at the 1$^{\text{st}}$ step, sieve bed 2 undergoes the 3$^{\text{rd}}$ step, and vice versa. The same holds for the two pressure equalization steps. Switching between consecutive steps is done by opening and closing different valves.

The original Skarstrom cycle did not have the two pressure equalization steps; they were added later [18]. Their only purpose is to increase the energetic efficiency of the process by conserving energy that would otherwise be lost during the exhaust step. For that reasons this PSA cycle can be found in other studies with [19, 20] and without [21, 2, 22] these two extra steps.

The functionality of each cycle step is explained below, shown from the point of view of sieve bed 1. In Figure 2-4, a visual explanation is provided.

1. **Feed**
   Compressed air enters sieve bed 1 through the feed end, and as a result the bed is pressurized to a higher operating pressure. Nitrogen is captured in the sieve bed, and as soon as the pressure in the sieve bed exceeds the pressure in the oxygen tank, a check valve opens and oxygen flows from the product end to the oxygen tank. A fraction of the product gas is bled of via the purge orifice to purge the other sieve bed, as will be explained in step 3.

2. **Pressure equalization I**
   During this step, instead of directly exhausting the sieve bed, the active valve connecting the two sieve beds at their product ends is opened. The high pressure of sieve bed 1 is used to partially pressurize sieve bed 2, which has just finished the exhaust step.

3. **Exhaust**
   The feed end of sieve bed 1 is opened and the pressure is further reduced in the sieve bed. Nitrogen is released by the adsorbent and the bed is regenerated. At the same time, a fraction of the product gas from sieve bed 2 (which is now at step 1) is used to flush away the remaining nitrogen molecules still present in the sieve bed.

4. **Pressure equalization II**
   This is the same as the second step, but with the roles reversed: sieve bed 1 now receives pressure from sieve bed 2 and the compressor feed stream.

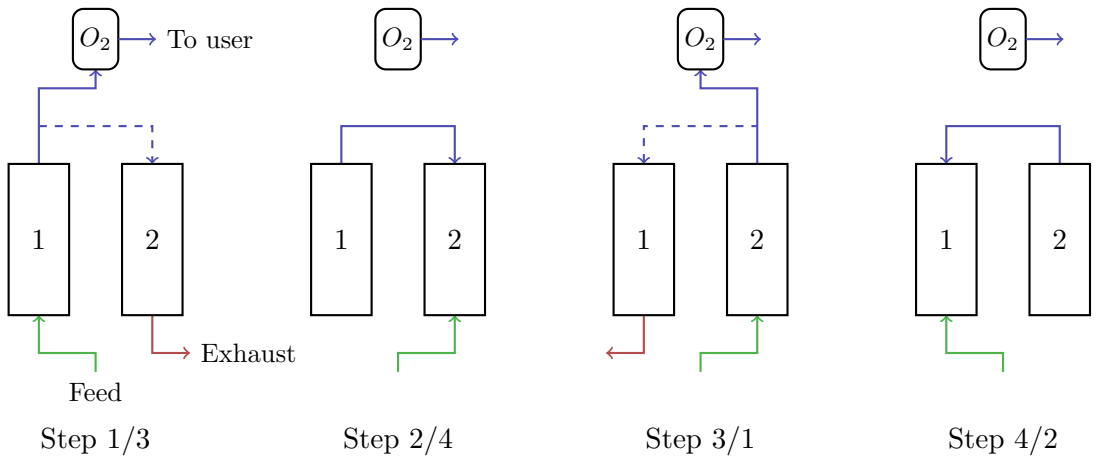

Figure 2-4: Visualization of the 4 PSA steps. Blue lines denote flows of oxygen-enriched air, red lines of nitrogen-rich air, and green lines of compressed ambient air. The sieve beds are denoted with "1" and "2", and the oxygen tank by "O_2". The sieve beds alternate between producing oxygen and exhausting nitrogen. Note that the oxygen tank provides a continuous stream of oxygen-enriched air to the user.

It is worth pointing out that the purge orifice is always open. This means that there is always a flow from the high-pressure sieve bed to the low-pressure sieve bed through this orifice, also during the pressure equalization steps.

2-2-4 Sieve bed Degradation

One unwanted side-effect of the oxygen separation process is the adsorption of water vapor inside a sieve bed. Although steps are usually taken to remove moisture from the feed stream, some moisture will pass through into the sieve beds [1, 23]. The attractive forces between the water molecules and the adsorbent are so strong that desorption is not possible under normal operating conditions. This means that this accumulation of water vapor inside the beds is permanent, reducing the adsorption capacity of the adsorbent inside the beds [12]. Fortunately, the adsorption kinetics of water are slow, so most of the water molecules will be purged out before they can adsorb into the zeolite pellets. Nevertheless, the small accumulation of these water molecules in the adsorbent bed will eventually result in sieve bed degradation, noticeable after months of use.

2-3 Concluding Remarks

In this chapter, the functioning of the POC has been explained. It is shown that the POC operates in a cyclic manner, where oxygen is produced in one sieve bed and the other sieve bed is exhausted. This cycle can be divided into four steps, which the sieve beds undergo in a phase-shifted manner. These steps should be carefully timed to prevent a sieve bed from saturating completely, at which nitrogen breakthrough occurs and the oxygen concentration of the air to the user is lowered.

The dynamics in the sieve beds have also been discussed: the concept of the MTZ has been introduced, where the adsorption front inside the sieve bed can be represented by a traveling concentration wave.

Chapter 3

High-Fidelity Modeling

Since a real POC cannot be used for control design, a dynamical model will be created. The choice has been made to develop a first-principles based, high-fidelity model to capture the dynamics of a POC, which is the subject of this chapter. These models can be complex and computationally expensive, but are needed to capture all the nonlinear dynamics of a PSA process, such as breakthrough and saturation. For this reason these kinds of models are also used in comparable studies on PSA plants [7, 13, 20, 24].

The components that make up a POC have already been introduced in Section 2-1: a POC consists of multiple valves, two sieve beds, a compressor and an oxygen tank. Because the nonlinear dynamics (i.e. adsorption) take place in the sieve beds, the main focus of this chapter will be on modeling these components.

The outline of this chapter is as follows: first, in Section 3-1, a short over view will be given of the system and its components, and the assumptions and simplifications made in this study will be explained. After that, the peripheral components will be discussed in Section 3-2, and the sieve beds in Section 3-3. The choice of inputs and outputs to this system will be explained in Section 3-4, and the chapter will conclude with a summary found in Section 3-5.

3-1 System Overview and Assumptions

Before the individual components will be discussed in the next sections, a short overview of the model is given here. The same schematic overview that was given in Section 2-1 will be used, now also showing all the states of the dynamic model. It can be found in Figure 3-1. The meaning of all the mentioned states will be explained below.

The gas flowing through the POC is described by the following state variables: molar flux Q, in [mol s^{-1}], pressure p in [Pa], and nitrogen and oxygen mole fractions γ_{N_2} and γ_{O_2}. Inside the sieve beds, q_{N_2} and q_{O_2} denote the adsorbed phase concentration of nitrogen and oxygen, respectively (both in [mol kg^{-1}]). The flow through a valve is given by Q_v, also in [mol s^{-1}]. Note that some states also have a spatial dependence, this is denoted by the suffix

(z). Superscripts are used to denote to which component a state belongs: $j \in \{1, 2\}$ for the two sieve beds, t for the oxygen tank, and c for the compressor.

In the following sections, the suffixes (z) and (t) will be left out of the equations, unless they are necessary to clarify an equation.

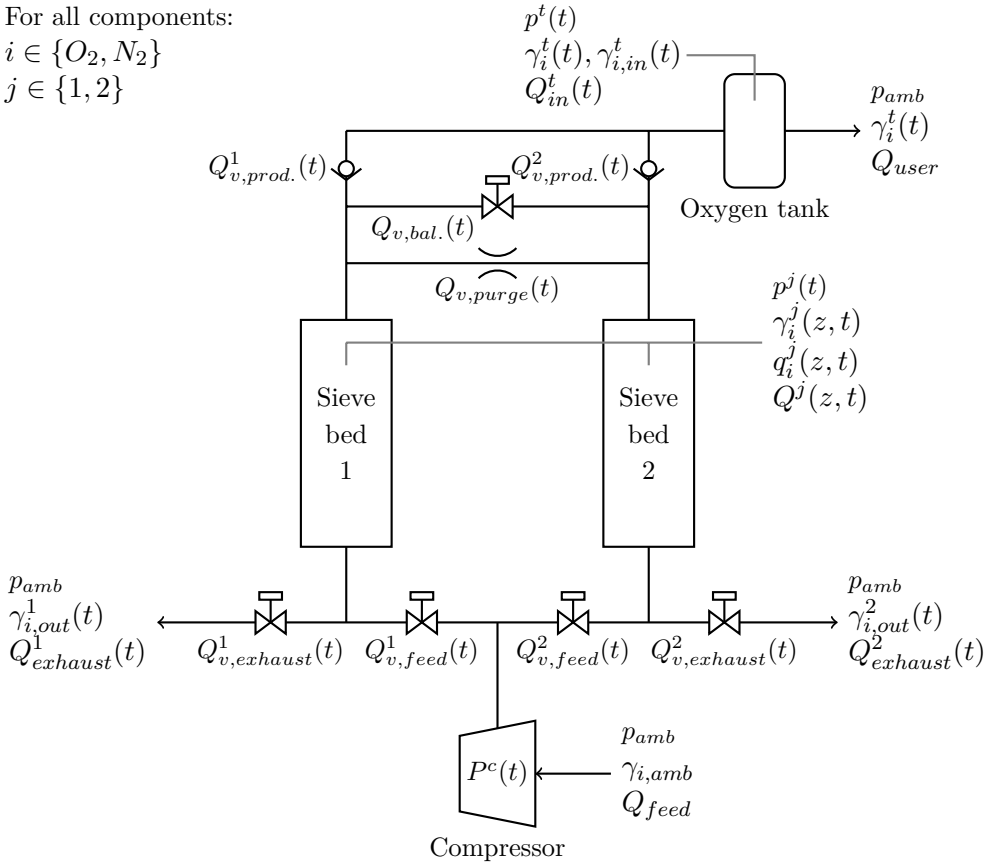

Figure 3-1: Schematic overview of the POC, with the states of the model. Some constants are also shown, i.e. the parameters without a temporal or spatial dependence. The subscript amb stands for ambient conditions. Two valve names are abbreviated here, namely prod.: production, and bal.: balance.

3-1-1 Assumptions and Simplifications

The following assumptions are made for the POC model:

1. The gas behaves as an ideal gas mixture, i.e. the ideal gas law,

$$p = cRT \quad \text{or equivalently} \quad p\gamma_i = c_i RT, \tag{3-1}$$

holds [8]. Here c is the gas-phase concentration in [mol m^{-3}], either of the total gas mixture or of component i in the gas mixture. T in [K] is the temperature, and R the gas constant, in [J K^{-1} mol^{-1}].

2. The pressure drop in the adsorption column, due to viscous energy losses and drop in kinetic energy, is considered negligible [24, 25].

3. The PSA process (i.e. the adsorption) is considered to be isothermal. In [26] it is reasoned that this is a reasonable assumption for oxygen production from air. For the compressor, adiabatic compression is assumed.

4. Radial variations in concentration and pressure inside the adsorbers are considered negligible compared to the axial variations. This assumption is found throughout the literature [7, 13, 16, 20].

5. The composition of air in this model is 78% nitrogen, 21% oxygen, and 1% argon. However, to reduce the computational burden (i.e. one less mass balance, see Section 3-3-1), argon and oxygen will be grouped together in the sieve beds. This can be done because these two gasses have almost identical adsorption properties [21, 27]. In the oxygen tank, the two gasses will be split up again. Other studies also use this approach [27], or just ignore argon altogether [20, 22].

3-2 Modeling of Peripheral Components

In this section the peripheral components are discussed. These are the valves and orifices, the oxygen tank, and the compressor.

3-2-1 Valves and Orifices

The flows between components of the POC are controlled by valves and orifices, and also need to be modeled. This is done with valve equations. These equations are adapted from [28], and relate the molar flow rate Q_v through a valve to the up- and downstream pressure p_u and p_d respectively, and the valve input u_v:

$$Q_v = F_v(p_u, p_d)u_v, \tag{3-2}$$

with F_v the actual valve equation:

$$F_v = \begin{cases} 85.6\frac{p_{stp}}{RT_{stp}}k_v\sqrt{\frac{(p_u-p_d)p_d}{\rho_{air,stp}T}} & \text{if} \quad p_d > \frac{p_u}{2}, \\ 42.8\frac{p_{stp}}{RT_{stp}}k_v\frac{p_u}{\sqrt{\rho_{air,stp}T}} & \text{else.} \end{cases} \tag{3-3}$$

In this equation, k_v is the flow coefficient in [sl/min] (standard liters per minute), which denotes the flow through a valve at standard conditions[1] and a pressure drop of 1 bar, and depends on the valve. $\rho_{air,stp}$ is the density of air at standard conditions (in [kg m^{-3}]), and p_{stp} is the pressure at standard conditions.

The active valves (feed, balance, exhaust) are controlled by a binary valve input, so for these valves $u_v \in \{0, 1\}$. The purge orifice does not have an input and is always open, so for convenience sake $u_{v,purge} = 1$. The check valves between the sieve beds and the oxygen tank

[1] Standard conditions for Temperature and Pressure (STP): 0 °C and 10^5 Pa [29].

only permit flow in one direction (towards the oxygen tank), hence their state depends on the pressure in the corresponding sieve bed:

$$u^j_{v,production} = \begin{cases} 1 & \text{if} \quad p^j \geq p^t, \\ 0 & \text{else.} \end{cases} \tag{3-4}$$

3-2-2 Oxygen Tank

In the oxygen tank, oxygen-rich air from the sieve beds is stored until it can be fed to the user. The states that are needed to describe the dynamics have already been introduced in Figure 3-1: the tank pressure p^t, the N_2 mole fraction in the tank $\gamma^t_{N_2}$, the molar flux entering the tank Q^t_{in}, and the corresponding N_2 mole fraction of the air entering the tank, $\gamma^t_{N_2,in}$. Both Q^t_{in} and $\gamma^t_{N_2,in}$ depend on the states in the sieve bed and the current PSA cycle step. In this study, Q^t_{user}, the flow rate of oxygen-rich air to the user of the POC, is a constant.

Two mass balances have been derived from the ideal gas law (Eq. (3-1)), namely the total mass balance and component mass balance:

$$\frac{dp^t}{dt} = \frac{RT}{V^t}\left(Q^t_{in} - Q^t_{user}\right), \tag{3-5a}$$

$$\frac{d\gamma^t_{N_2}}{dt} = \frac{RT}{p^t V^t}\left(Q^t_{in}\gamma^t_{N_2,in} - Q^t_{user}\gamma^t_{N_2}\right) - \frac{\gamma^t_{N_2}}{p^t}\frac{dp^t}{dt}. \tag{3-5b}$$

Here V^t is the volume of the tank, in [m^3]. Recall from Section 3-1-1, that argon and oxygen have been grouped together in the sieve beds, but split up again in the oxygen tank. These two mole fractions are calculated using $\gamma^t_{N_2}$:

$$\gamma^t_{O_2} = (1 - \gamma^t_{N_2}) \cdot \frac{21}{22}, \tag{3-6a}$$

$$\gamma^t_{Ar} = (1 - \gamma^t_{N_2}) \cdot \frac{1}{22}. \tag{3-6b}$$

Finally, the molar flux entering the oxygen tank and its corresponding mole fraction are defined as:

$$Q^t_{in} = Q^1_{v,production} + Q^2_{v,production}, \tag{3-7}$$

$$\gamma^t_{N_2,in} = \begin{cases} \gamma^1_{N_2}(z = L) & \text{if} \quad Q^1_{v,production} > 0, \\ \gamma^2_{N_2}(z = L) & \text{if} \quad Q^2_{v,production} > 0. \end{cases} \tag{3-8}$$

3-2-3 Compressor

In a real POC, the role of the compressor is to pressurize the air going into the sieve beds. The mass flow Q_{feed} and pressure increase are controlled by adjusting the RPM of the compressor. In this model, the pressure increase is calculated in the sieve beds, and Q_{feed} is set constant. Therefore a compressor is not needed to simulate the PSA process.

The reason a compressor is included in Figure 3-1, is because an estimate of its power consumption P^c (in [W]) is needed. This will be estimated by calculating the adiabatic power

needed to compress the air from p_{amb} to either p^1 or p^2, depending on the current PSA cycle step. First the adiabatic compression work is needed [30]:

$$H^c = \frac{\kappa}{\kappa - 1} R T_{amb} \left(\left(\frac{p^c}{p_{amb}} \right)^{\frac{1-\kappa}{\kappa}} - 1 \right), \tag{3-9}$$

with H^c, also called the head, in [J/mol], κ the heat capacity ratio of air, and p^c the compressor pressure. Note that p^c is equal to either p^1 or p^2.

Subsequently, the compressor power is estimated as

$$P^c = Q_{feed} \frac{H^c}{\eta^c}, \tag{3-10}$$

where Q_{feed} is the molar flow entering the compressor, and η^c the adiabatic compressor efficiency.

3-3 Modeling of Sieve Beds

Now that all the other components have been described, the modeling of the sieve beds can be discussed. What sets these sieve beds apart from the other components, is the fact that coupled Partial Differential Equations (PDEs) are needed to describe all the necessary dynamics [16].

Given the assumptions made in this study, the dynamics of a sieve bed are modeled by mass balances, rate equations and adsorption isotherms.

3-3-1 Mass Balance

The governing equation for a PSA system is the mass balance, which describes the exchange between the gaseous concentration c and the adsorbed concentration q, over a small axial element of the adsorption column.

The component mass balance is taken from [8, 16], and holds for each sieve bed $j \in \{1, 2\}$:

$$\epsilon \frac{\partial c_i^j}{\partial t} + \epsilon \frac{\partial (v c_i)^j}{\partial z} + (1 - \epsilon) \rho_p \frac{\partial q_i^j}{\partial t} = \epsilon D_L \frac{\partial}{\partial z} \left(c^j \frac{\partial \gamma_i^j}{\partial z} \right). \tag{3-11}$$

Here ϵ is the voidage of the sieve bed, i.e. the fraction of unoccupied space in the adsorption column. v is the interstitial velocity of the gas in [m s^{-1}], ρ_p the density of the adsorbent pellets, and z the axial (spatial) coordinate. Lastly, D_L, in [m^2 s^{-1}], is a coefficient for the axial dispersion.

The term on the right hand side of the equation, describing the axial dispersion, is sometimes omitted. Opinions differ in the literature whether this is a sound decision; Ruthven [16] states that for smaller PSA units this axial dispersion is more significant, and advises to keep it. On the other hand, many studies on (portable) oxygen concentrators leave this term out of their mass balances [7, 20, 31]. Since the important dynamics (adsorption, MTZ) are captured with the other terms, it will be removed for this study.

Likewise with the oxygen tank, two mass balances are also needed for the sieve bed to fully describe the dynamics. Which components are used is trivial, since the other component (or the total concentration) can be calculated using the ideal gas law. In this thesis, the choice has been made to use the component mass balance of nitrogen and the total mass balance, and calculate the mole fraction of oxygen accordingly.

Since the models of the peripheral components all use the mole fraction γ and the molar flow rate Q, Eq. (3-11) must be slightly rewritten. The ideal gas law (Eq. (3-1)) is again used, as well as the following Eqs. (3-12) and (3-13):

$$cv = \frac{Q}{\epsilon A_b}, \tag{3-12}$$

with A_b (in $[m^2]$) the cross sectional area of the adsorption bed. Furthermore,

$$\rho_b = (1-\epsilon)\rho_p. \tag{3-13}$$

Here ρ_b is the bulk density, i.e. the density of the sieve bed. Then the total mass balance for each sieve bed $j \in \{1,2\}$ with a N_2, O_2 gas mixture becomes:

$$\frac{\partial p^j}{\partial t} = -\frac{RT}{\epsilon}\left(\frac{1}{A_b}\frac{\partial Q^j}{\partial z} + \rho_b\left(\frac{\partial q^j_{N_2}}{\partial t} + \frac{\partial q^j_{O_2}}{\partial t}\right)\right), \tag{3-14}$$

and the component mass balance:

$$\frac{\partial \gamma^j_{N_2}}{\partial t} = -\frac{RT}{p^j\epsilon}\left(\frac{1}{A_b}\frac{\partial (Q^j\gamma^j_{N_2})}{\partial z} + \rho_b\frac{\partial q^j_{N_2}}{\partial t}\right) - \frac{\gamma^j_{N_2}}{p^j}\frac{\partial p^j}{\partial t}. \tag{3-15}$$

3-3-2 Rate Equations & Adsorption Equilibria

Describing the rate at which gas is adsorbed inside an adsorption column is an important aspect of modeling of PSA systems. Two fundamental properties need to be captured by these equations: firstly, that mass transfer between the adsorbed state and the gaseous state is not instantaneous, and secondly, that the adsorbate has a saturation limit. The following is a brief summary on these topics, and is extracted from [8, 13, 24]. The interested reader is referred to these works for more details.

Rate Equations

Rate equations, sometimes called mass transfer kinetics, are used to describe the interphase mass transfer in e.g., adsorption columns. To fully describe all the mass transfer resistances the adsorbate experiences would require a rigorous model, which increases computational complexity.

However, throughout the literature, it is stated that these diffusion rates can be approximated by the Linear Driving Force (LDF) model, which given in Eq. (3-16):

$$\frac{dq_i}{dt} = K_{LDF_i}(q_i^* - q_i), \tag{3-16}$$

with K_{LDF} a mass transfer coefficient in [s^{-1}], and q^* the equilibrium value for the adsorbed amount of component i (at current temperature and pressure). Higher order approximations of the diffusion model have also been used in the literature, but only sporadically [19]. It should be noted that Khajuria [13] states that for "very fast" PSA processes (i.e. with cycle lengths comparable to this POC), corrective terms are needed to ensure the validity of the LDF model. Santos [7] also speaks about this in a study about small oxygen PSA units, but states that for equilibrium-based separations (as is the case for O_2, N_2 separation), the use of the LDF model is valid. So, because of this, and its widespread use in comparable studies (see also [22, 32]), the use of this simplification is deemed valid.

Adsorption Equilibria

So-called isotherm equations relate q^*, the current equilibrium value, to q^s, the saturation value of the adsorbent for component i. Different equations can be used for this, and one of the most popular isotherms is the Langmuir isotherm [8]:

$$\frac{q_i^*}{q_i^s} = \frac{k_i p}{1 + p k_i}. \tag{3-17}$$

Here k is the Langmuir equilibrium constant in [Pa^{-1}]. An overview of other (basic) isotherms can be found in [8].

Usually the adsorbate consists of multiple components, as is the case with air, and multi-component isotherm equations are needed. The multi-component Langmuir isotherm is:

$$\frac{q_i^*}{q_i^s} = \frac{k_i p}{1 + p \sum_j k_j \gamma_j}, \tag{3-18}$$

which adequately describes the adsorption equilibrium. It should be noted that the equilibrium constants of single-component and multi-component isotherms differ: e.g. k_{O_2} of pure oxygen will not be the same as k_{O_2} of an air mixture.

Naturally, more complex isotherm equations exist. Examples are multi-site isotherms [33], and the Ideal Adsorbed Solution Theory (IAST) [34], which permits the calculation of adsorption equilibria for components in a gaseous mixture using only pure-component data [20]. The downside of the latter method is that it is an implicit approach; implicit algebraic equations have to be solved for q^*. The choice for the ordinary multi-component Langmuir isotherm in Eq. 3-18 was made because its suitability for the POC and this research.

3-3-3 Spatial Discretization

As can be seen in the previous section, the dynamics of an adsorption column are made up by three sets of equations in this thesis: mass balances (Eqs. (3-14) and (3-15)), a rate equation (Eq. (3-16)) and adsorption isotherms (Eq. (3-18)). This system of coupled PDEs cannot be solved by the built-in PDE solver of MATLAB, since that requires that all equations are PDEs [35]. Both the adsorption isotherms and the rate equation are "merely" Ordinary Differential Equations (ODEs), so this system of equations has to be solved in another way.

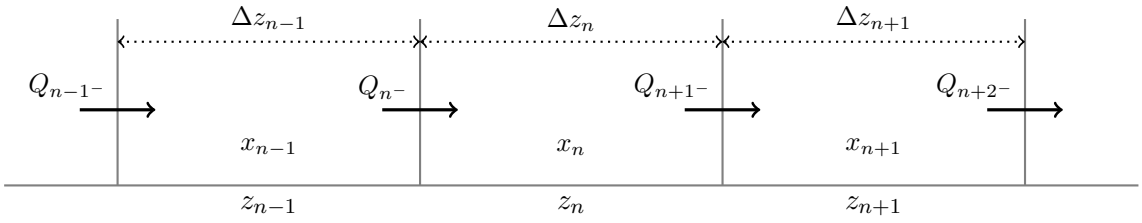

Figure 3-2: Visual explanation of the spatial discretization. The adsorption column is discretized in N compartments (or finite volumes) $z_n, n \in \{1, \ldots, N\}$. x_n is the value of the state x at the center of compartment z_n, and Δz_n is the length of compartment n. Note that Q_{n^-} represents the molar flux from compartment z_{n-1} to z_n (or vice versa). Obviously, $Q_{n-1^+} = Q_{n^-}$, but for clarity only the latter is used.

Finite Volume Method

An approach that is found often in the literature is to discretize the spatial domain of the sieve beds manually, such that the PDEs are changed into a set of ODEs. This is done with the Finite Volume Method (FVM) [25], which is the topic of this section. A visual explanation can be found in Figure 3-2. For clarity, the superscript j for the sieve beds will be dropped temporarily.

The FVM is a conservative discretization method, which implies conservation of mass (and energy) [36, 37]. For that reason it is widely used in fluid dynamics, and modeling of PSA processes as well [20, 24]. The spatial domain is divided into a discrete number N of volume elements, and the respective equations are integrated over a volume element n. Take e.g. the total mass balance, Eq. (3-14), integrated over such a volume element with boundaries at z_{n^-} and $z_{n^+} = z_{n+1^-}$:

$$\int_{z_{n^-}}^{z_{n+1^-}} \frac{\partial p}{\partial t} dz = -\int_{z_{n^-}}^{z_{n+1^-}} \frac{RT}{\epsilon} \left(\frac{1}{A_b} \frac{\partial Q}{\partial z} + \rho_b \left(\frac{\partial q_{N_2}}{\partial t} + \frac{\partial q_{O_2}}{\partial t} \right) \right) dz, \tag{3-19a}$$

$$\frac{\partial p}{\partial t} \, z \Big|_{z_{n^-}}^{z_{n+1^-}} = -\frac{RT}{\epsilon} \left(\frac{1}{A_b} \, Q \Big|_{z_{n^-}}^{z_{n+1^-}} + \rho_b \left(\frac{\partial \bar{q}_{N_2}}{\partial t} + \frac{\partial \bar{q}_{O_2}}{\partial t} \right) z \Big|_{z_{n^-}}^{z_{n+1^-}} \right). \tag{3-19b}$$

Note that $\bar{x}_n$ are the time- and position-dependent states $x(z,t) = [\gamma_{N_2}, q_{N_2}, q_{O_2}]$, spatially averaged over a control volume z_n. The assumption will be made that for large enough N, $\bar{x}_n = x_n$, where x_n is the value of the state in the center of the control volume. Furthermore, $\Delta z_n = z_{n+1^-} - z_{n^-}$, and although the finite volumes could theoretically be of different sizes, in this thesis this is not the case, so $\Delta z_n = \Delta z, \forall n$. Then Eq. (3-19b) becomes:

$$\frac{dp}{dt} = -\frac{RT}{\epsilon} \left(\frac{1}{A_b \Delta z} \left(Q_{n+1^-} - Q_{n^-} \right) + \rho_b \left(\frac{dq_{N_2,n}}{dt} + \frac{dq_{O_2,n}}{dt} \right) \right), \qquad n \in \{1, \ldots, N\}. \tag{3-20}$$

In this equation, Q_{n^-} is the molar flux evaluated at the left boundary of the control volume z_n. Likewise, Eq. (3-15) can be rewritten as:

$$\frac{d\gamma_{N_2,n}}{dt} = -\frac{RT}{p\epsilon} \left(\frac{1}{A_b \Delta z} \left(Q_{n+1^-} \gamma_{N_2,n+1^-} - Q_{n^-} \gamma_{N_2,n^-} \right) + \rho \frac{dq_{N_2,n}}{dt} \right) - \frac{\gamma_{N_2,n}}{p} \frac{dp}{dt}, \quad n \in \{1, \ldots, N\}. \tag{3-21}$$

Here γ_{N_2,n^-} is the mole fraction of nitrogen at the left side wall of the finite volume n. How the value of this state is calculated will be discussed in the next section. Since the molar flux now only appears at the boundaries of the finite volumes, the minus sign on Q will be dropped from the equations from now on: $Q_{n^-} = Q_n$.

With the two mass balances rewritten as two sets of coupled ODEs, the equations are now solvable by any MATLAB ODE solver. An overview of all the equations and how they relate to each other can be found in Section 3-5.

Interpolation Schemes

As can be seen in Section 3-3-3, some states also appear at the boundaries of the cells. The molar flux Q_n can be derived from Eq.(3-20), as will be explained in Section 3-5, but to calculate the nitrogen mole fraction γ_{N_2,n^-} at the cell boundaries, interpolation is required.

However, it is known in the literature that using spatial discretization methods like the FVM with standard linear interpolation can yield unnatural oscillations. This is due to shocks, discontinuities and steep gradients in the solution domain, which can also occur in PSA processes. To counter these effects, flux limiters are used [38].

The main idea of flux limiters is to limit the spatial derivatives to realistic values. They only come into operation when sharp wave fronts are present, and do not operate for smoothly changing waves. To do so, first the ratio between successive gradients of the state variable x_n, for each n is defined:

$$r(n) = \begin{cases} \frac{x_{n-1}-x_{n-2}}{x_n-x_{n-1}} & \text{if} \quad Q_n \geq 0, \\ \frac{x_n-x_{n+1}}{x_{n-1}-x_n} & \text{if} \quad Q_n < 0, \end{cases} \tag{3-22}$$

which is used to calculate the limiter function. Different limiter functions exist (for examples see [38]), and in this thesis the Van Leer flux limiter is used. The reason for this is the fact that it produces favorable results, and its use in comparable studies [24, 39]. It is defined as

$$\Phi(n) = \frac{r(n) + |r(n)|}{1 + |r(n)|}. \tag{3-23}$$

By design, the limiter function is constrained to be greater or equal than zero, i.e. $\Phi(n) \geq 0$. Its maximum value is $\lim_{r\to\infty} \Phi(n) = 2$.

With both these equations, the state variables at the boundaries are subsequently calculated with:

$$x_{n^-} = \begin{cases} x_{n-1} + \frac{1}{2}(x_n - x_{n-1})\,\Phi(n) & \text{if} \quad Q_n \geq 0, \\ x_n + \frac{1}{2}(x_{n-1} - x_n)\,\Phi(n) & \text{if} \quad Q_n < 0. \end{cases} \tag{3-24}$$

This interpolation scheme is also taken from [24, 39]. What this all means in practice, is that the flux limiter switches to a higher order interpolation scheme when the difference in gradients is small ($\Phi(n) \approx 1$), and a low order interpolation scheme when the difference in gradients is large (e.g. steep shock waves are present, $\Phi(n) \to 0 \vee \Phi(n) \to 2$). This low order scheme does not have the problem of oscillations. In Chapter 4 the effect this flux limiter will be evaluated.

Boundary Conditions

The final step in describing the dynamics of the sieve beds is setting up the Boundary Conditions (BCs) at the ends of the sieve beds, i.e. at z_{1^-} and z_{N^+}. These can be defined using the valve equations from Section 3-2-1. For each adsorption column $j = \{1, 2\}$, the BCs for the molar flow rate are given as (adapted from [20]):

$$\begin{aligned} Q^j(0,t) = Q^j_1 &= Q^j_{v,feed} + Q^j_{v,exhaust}, \\ &= F_v(p_{atm}, p^j)u^j_{v,feed} + F_v(p_{atm}, p^j)u^j_{v,exhaust}, \\ Q^j(L,t) = Q^j_{N+1} &= Q^j_{v,production} + (-1)^{j-1}Q_{v,balance} + (-1)^{j-1}Q_{v,purge}, \\ &= F_v(p^j, p^t)u^j_{v,production} + (-1)^{j-1}F_v(p^1, p^2)u_{v,balance} \\ &\quad + (-1)^{j-1}F_v(p^1, p^2)u_{v,purge}. \end{aligned} \tag{3-25}$$

Note that the pressure p and valve states u also depend on (t), but that this suffix has been dropped from the equations for clarity.

The BCs for the nitrogen mole fraction, that is, $\gamma^j_{N_2,1^-}$ and $\gamma^j_{N_2,N^+}$, are calculated using ghost cells. What this means is that on each side of the spatial domain, a new cell is added, such that Eq. (3-24) can be used normally at every finite volume n. These are:

$$\begin{aligned} \gamma^j_{N_2,0} &= \gamma_{N_2,amb}, \\ \gamma^1_{N_2,N+1} &= \gamma^2_{N_2,N}, \\ \gamma^2_{N_2,N+1} &= \gamma^1_{N_2,N}. \end{aligned} \tag{3-26}$$

3-4 Inputs and Outputs

Although it is theoretically possible to control a PSA process by adjusting every valve individually, this is a very impractical approach. Doing so would require setting constraints on the valves, since e.g. the feed valve and exhaust valve of a sieve bed cannot be open at the same time (this is highly inefficient). An easier option, also often used in the literature [13, 20, 24], is to use lengths of the PSA cycle steps as input(s). The valve states $u_v \in \{0, 1\}$ are subsequently retrieved from a lookup table.

With that in mind, in this thesis the input will be the feed time τ_f, the duration of the first step. This feed time will be identical for both sieve beds during a cycle, so τ_f also denotes the length of the mirroring step, i.e. step 3 (exhaust). The duration of the balance step, τ_b, is constant. The reason for this is that this steps only purpose is to balance the pressures, and the time that this takes is more or less the same for the operating range of this POC. Any increase in performance in terms of mole fraction or power consumption is not really achieved by also manipulating this value. The lookup table for this POC can be found in Table 3-1.

The outputs of the POC are the oxygen mole fraction in the tank, $\gamma^t_{O_2}$, and the power usage by the compressor, P^c. As will be shown in the next chapter, these two outputs oscillate during a cycle, whereas the input is constant. Therefore it is convenient to also define outputs that are also constant during a cycle. This way the dynamics of one cycle k can be captured by three variables: the input $\tau_{f,k}$ and the two corresponding outputs. These two outputs are the

	$u^1_{v,feed}$	$u^2_{v,feed}$	$u^1_{v,exhaust}$	$u^2_{v,exhaust}$	$u_{v,balance}$
Step 1/3 (τ_f)	1	0	0	1	0
Step 2/4 (τ_b)	0	1	0	0	1
Step 3/1 (τ_f)	0	1	1	0	0
Step 4/2 (τ_b)	1	0	0	0	1

Table 3-1: Binary values for the active valves in relation to the PSA steps.

cycle-averaged O_2 mole fraction in the tank, also called the purity $\mathcal{P}_k$, and the cycle-averaged compressor power $\bar{P}^c_k$:

$$\mathcal{P}_k = \frac{100}{\tau_{t,k}} \int_{t_k}^{t_k+\tau_{t,k}} \gamma^t_{O_2} dt, \tag{3-27}$$

$$\bar{P}^c_k = \frac{1}{\tau_{t,k}} \int_{t_k}^{t_k+\tau_{t,k}} P^c dt. \tag{3-28}$$

Here t_k denotes the time at which cycle k starts and $\tau_{t,k}$ denotes the length of this cycle: $\tau_{t,k} = 2(\tau_{f,k} + \tau_b)$. The reason cycle-averaged outputs are used and not the value of the time-domain outputs ($\gamma^t_{O_2}$ and P^c) at e.g., the end of a cycle, is because these cycle-averaged outputs paint a better picture of the dynamics of that cycle, especially for the transient behavior. For that reason cycle-averaged outputs are also used in other studies [7, 13, 20]. Their use will be highlighted in the next chapters.

3-5 Concluding Remarks

In this chapter the modeling decisions for a high-fidelity model of a POC have been discussed. Because part of the equations are coupled PDEs, they have been spatially discretized to ODEs to allow for calculation by any ODE solver. The resulting 2-bed model consists of $6N + 4$ ODEs: $p^t, p^1, p^2, \gamma^t_{N_2}, \gamma^j_{N_2,1}, \ldots, \gamma^j_{N_2,N}, q^j_{N_2,1}, \ldots, q^j_{N_2,N}, q^j_{O_2,1}, \ldots, q^j_{O_2,N}, j \in \{1, 2\}$.

In Tables 3-2 and 3-3, a summary of all the equations can be found.

Sieve bed model for each of the two adsorbers $\mathbf{j} \in \{1, 2\}$

Boundary values for molar flux Q^j (Section 3-3-3):

$$\begin{aligned} Q_1^j &= Q_{v,feed}^j + Q_{v,exhaust}^j, \\ Q_{N+1}^j &= Q_{v,production}^j + (-1)^{j-1} Q_{v,balance} + (-1)^{j-1} Q_{v,purge}. \end{aligned} \tag{3-29}$$

Adsorbed state concentrations $q_i^j, i \in \{N_2, O_2\}$ (Section 3-3-2):

$$\frac{dq_{i,n}^j}{dt} = k_{LDF,i} \left(\frac{q_i^s k_i \gamma_{i,n}^j p^j}{1 + p^j \left(k_{O_2} \gamma_{O_2,n}^j + k_{N_2} \gamma_{N_2,n}^j \right)} - q_{i,n}^j \right), \qquad n \in \{1, \ldots, N\}. \tag{3-30}$$

Pressure dp^j/dt (Sections 3-3-1 and 3-3-3):

$$\frac{dp^j}{dt} = -\frac{RT}{\epsilon L} \left(\frac{1}{A_b} \left(Q_{N+1}^j - Q_1^j \right) + \rho_b \Delta z \sum_{n=1}^{N} \left(\frac{dq_{N_2,n}^j}{dt} + \frac{dq_{O_2,n}^j}{dt} \right) \right). \tag{3-31}$$

Non-boundary values of Q_n^j (Sections 3-3-1 and 3-3-3):

$$Q_{n+1}^j = Q_n^j - \left(\frac{\epsilon}{RT} \frac{dp^j}{dt} + \rho_b \left(\frac{dq_{N_2,n}^j}{dt} + \frac{dq_{O_2,n}^j}{dt} \right) \right) A_b \Delta z, \qquad n \in \{1, \ldots, N-1\}. \tag{3-32}$$

Nitrogen mole fraction $d\gamma_{N_2,n}^j/dt$ (Sections 3-3-1, 3-3-3 and 3-3-3):

$$\frac{d\gamma_{N_2,n}^j}{dt} = -\frac{RT}{p^j \epsilon} \left(\frac{1}{A_b \Delta z} \left(Q_{n+1}^j \gamma_{N_2,n+1^-}^j - Q_n^j \gamma_{N_2,n^-}^j \right) + \rho_b \frac{dq_{N_2,n}^j}{dt} \right) - \frac{\gamma_{N_2,n}^j}{p^j} \frac{dp^j}{dt}, \qquad n \in \{1, \ldots, N\}, \tag{3-33}$$

with (Section 3-3-3):

$$\gamma_{N_2,n^-}^j = \begin{cases} \gamma_{N_2,n-1}^j + \frac{1}{2} \left(\gamma_{N_2,n}^j - \gamma_{N_2,n-1}^j \right) \Phi(n) & \text{if} \quad Q_n^j \geq 0, \\ \gamma_{N_2,n}^j + \frac{1}{2} \left(\gamma_{N_2,n-1}^j - \gamma_{N_2,n}^j \right) \Phi(n) & \text{if} \quad Q_n^j < 0. \end{cases} \tag{3-34}$$

Table 3-2: Sieve bed equations for POC model.

Valve models

Molar flow:

$$Q_v = F_v(p_u, p_d)u_v, \tag{3-35a}$$

$$F_v = \begin{cases} 85.6\frac{p_{stp}}{RT_{stp}}k_v\sqrt{\frac{(p_u-p_d)p_d}{\rho_{air,stp}T}} & \text{if} \quad p_d > \frac{p_u}{2}, \\ 42.8\frac{p_{stp}}{RT_{stp}}k_v\frac{p_u}{\sqrt{\rho_{air,stp}T}} & \text{else.} \end{cases} \tag{3-35b}$$

Oxygen tank model

Mass balances:

$$\frac{dp^t}{dt} = \frac{RT}{V^t}\left(Q^t_{in} - Q^t_{user}\right), \tag{3-36a}$$

$$\frac{d\gamma^t_{N_2}}{dt} = \frac{RT}{p^tV^t}\left(Q^t_{in}\gamma^t_{N_2,in} - Q^t_{user}\gamma^t_{N_2}\right) - \frac{\gamma^t_{N_2}}{p^t}\frac{dp^t}{dt}, \tag{3-36b}$$

$$\gamma^t_{O_2} = (1-\gamma^t_{N_2})\frac{21}{22}, \tag{3-36c}$$

$$\gamma^t_{Ar} = (1-\gamma^t_{N_2})\frac{1}{22}, \tag{3-36d}$$

$$Q^t_{in} = Q^1_{v,production} + Q^2_{v,production}, \tag{3-36e}$$

$$\gamma^t_{N_2,in} = \begin{cases} \gamma^1_{N_2,N+1^-} & \text{if} \quad Q^1_{v,production} > 0, \\ \gamma^2_{N_2,N+1^-} & \text{if} \quad Q^2_{v,production} > 0. \end{cases} \tag{3-36f}$$

Compressor model

Power consumption:

$$H = \frac{\kappa}{\kappa-1}RT_{amb}\left(\left(\frac{p^c}{p_{amb}}\right)^{\frac{1-\kappa}{\kappa}} - 1\right), \tag{3-37a}$$

$$P_{comp} = Q_{feed}\frac{H}{\eta^c}. \tag{3-37b}$$

Table 3-3: Equations for peripheral components.

Chapter 4

Model Validation

Ideally, a dynamical model of a real-life system is validated with experiments. In the case of the POC, this is not possible due to a lack of sensors. Therefore, validation is done by comparing the POC with other comparable studies in the literature. Although these will not be 100% identical, these comparisons serve to prove that the important dynamics of a PSA process (i.e. breakthrough, saturation) are modeled correctly.

The aim of this chapter is twofold: show the behavior of the POC and compare it with studies in the literature. First, in Section 4-1, some modeling decisions will be substantiated with simulation results. The dynamics of the POC are subsequently discussed in two sections: in Section 4-2, steady state behavior will be discussed, and in Section 4-3 the transient dynamics are analyzed. After that, the comparison with literature will be made in Section 4-4, and the chapter will end with a conclusion.

4-1 Choice of Model Parameters

Most of the model parameters have been obtained from internal documents and discussions at Philips, and because of their confidentiality cannot be included in this report. In this section two modeling choices will be discussed that do not have this restriction: the first is the degree of spatial discretization, i.e. the choice of N, and the second is the effect of the flux limiter (for both see Section 3-3-3).

4-1-1 Choice of N

Choosing the right amount spatial volumes N is important: for $N \to \infty$, the dynamics in the sieve bed will get closer and closer to reality, but at the same time the number of ODEs increases too, and with that the computation time. Therefore a trade-off has to be made. In Figure 4-1 the results of different simulations with varying N can be found. In the left plot, the purity of the 100^{th} cycle is plotted against N, and in the right plot the computation time has been plotted against N.

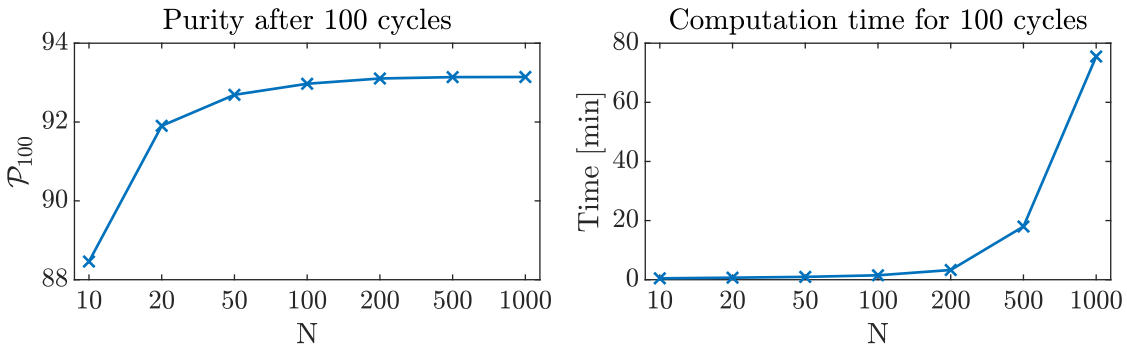

Figure 4-1: Two plots showcasing the effect of increasing N. The purity will go to a steady-state value, whereas the computation time skyrockets for large N.

The simulations have been done using MATLAB R2018a on a MacBook Pro, with a 2.7 GHz Intel Core i5 processor and 8GB of RAM. Looking ahead to Chapter 5, where large datasets have to be created, it is wise to weigh the computation time more than the slight error in purity that a smaller N entails. For that reason a spatial grid size of $N = 100$ has been chosen. Any increase in N will result in a marginally smaller purity error, but the computation time starts to increase exponentially.

4-1-2 Effect of Flux Limiter

The need of a flux limiter can also be demonstrated via simulations. Two cases will be compared: one where the van Leer flux limiter is used with the interpolation scheme (see Section 3-3-3), and one where the value of γ_{N_2} at the boundaries of the volumes is simply the average of the two neighboring volumes.

The results can be found in Figure 4-2. Here the spatial profiles of the O_2 mole fraction during the feed step can be seen, after both models have been simulated for 100 cycles, with $\tau_f = 5s$. Each line in the plot denotes a different time t in the cycle. It is clear that a flux limiter is needed: in the case without, unnatural oscillations due to the sharp wave fronts occur, whereas in the case with flux limiter the concentration waves are smooth.

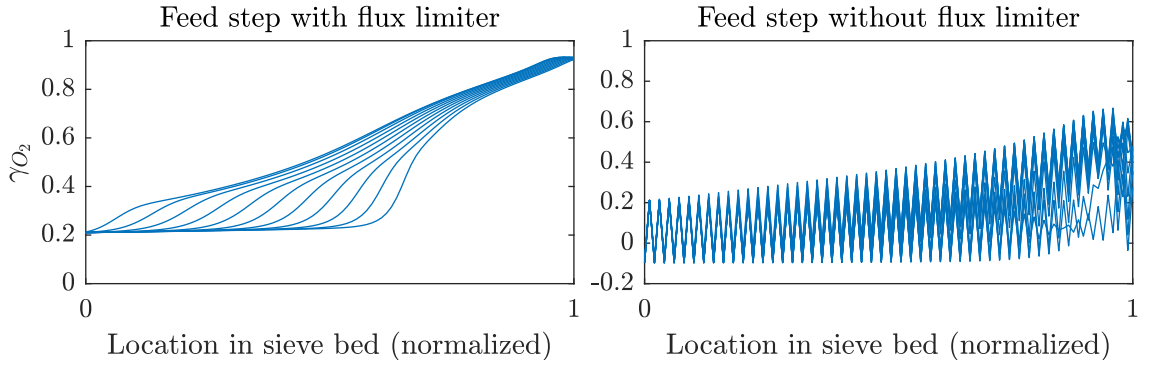

Figure 4-2: Effect of flux limiter. For much larger N, this effect would not happen, but the computational effort would be too high, as discussed in Section 4-1-1.

4-2 Cyclic Steady State Behavior

Now the steady-state behavior of the POC will be discussed. In the context of dynamical systems, a system is said to be in steady state if its states are no longer changing over time. If this definition is used for a PSA system, it would never reach steady state: due to its periodic nature, states like pressure and mole fractions change continuously. Therefore a different definition will be used, that of Cyclic Steady State (CSS).

If the input is held constant, and all other variables are kept the same, a PSA plant is said to reach CSS if conditions at time t are identical to $t+\tau_t$, i.e. exactly one cycle length later [20]:

$$x(z,t) = x(z,t+\tau_t) \quad \text{and} \quad x(t) = x(t+\tau_t). \tag{4-1}$$

CSS is either determined by simulating the plant over a large number of cycles, or by use of a numerical algorithm. In this thesis the former is used, and CSS is said to have been reached if the difference in purity $\mathcal{P}$ between two consecutive cycles is less than a value ε:

$$\text{CSS if} \quad |\mathcal{P}_{k+1} - \mathcal{P}_k| < \varepsilon. \tag{4-2}$$

The purity can also be seen as points on a Poincaré section [40] of the periodic orbit of the PSA system. A Poincaré map then maps the purity of the current cycle, $\mathcal{P}_k$, to the purity of the next cycle, $\mathcal{P}_{k+1}$. If this new purity is at the same location on the Poincaré section ($\pm\varepsilon$), the system is at CSS.

Santos [7] used the same definition, and said $\varepsilon = 0.01$. The same value will be used in this thesis.

4-2-1 Dynamics of a Cycle in CSS

To show the behavior of the POC in CSS, a close look will be taken at a cycle with an input of $\tau_f = 5s$.

Figure 4-3 shows two plots: the pressure in the three main components (sieve beds and oxygen tank), and the O_2 mole fraction in the tank. First the top plot will be discussed. The four PSA steps can be deduced from this image. in the first step, sieve bed 1 is pressurized, and after ca. 1.5 seconds the pressure has risen to the same level as in the oxygen tank, so oxygen rich air will flow to the tank. At the same time, sieve bed 2 is exhausted and its pressure drops to almost ambient levels. The second step shows the pressures in the two sieve beds equalizing, whereas the pressure in the oxygen tank drops due to the lack of product from any of the sieve beds, and the fact that it is still feeding oxygen-enriched air to the user. Steps 3 and 4 are the same as 1 and 2, but with the sieve beds reversed, and that is also clearly visible. The notion of CSS is evident: the three pressures at time $t = 0$ are all identical to the pressures at time $t = \tau_t$.

In the bottom plot of Figure 4-3, the variations in $\gamma^t_{O_2}$ during one cycle can be seen. One might wonder why the mole fraction is not 1, and this is partly due to the argon still present in the gas mixture ($\gamma^t_{Ar} \approx 0.042$), which cannot be removed, and partly due to some nitrogen ($\gamma^t_{N_2} \approx 0.028$) that did not get adsorbed in the sieve beds. The oscillations in mole fractions are due to the internal dynamics in the sieve beds, and these will be explained with Figure 4-4.

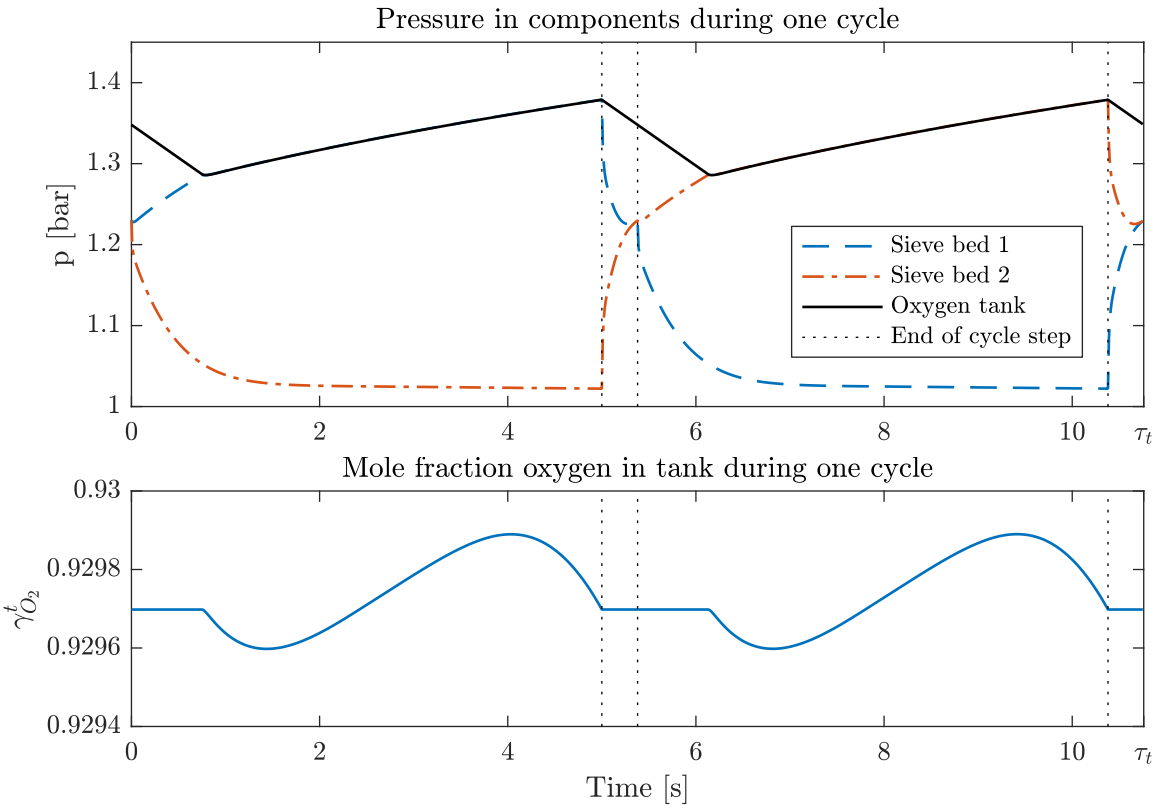

Figure 4-3: Pressure variations in the sieve beds and oxygen tank (top), and variations in the O_2 mole fraction in the tank (bottom).

For the evolution of the other states in the sieve bed, the same type of plot from Section 4-1-2 is used. Two important states will be highlighted: the O_2 mole fraction and the adsorbed quantity of N_2 in a sieve bed. They are both plotted against the normalized spatial location z, each line denoting a different time t. For each subplot, the initial spatial profile is denoted by the red line, and arrows are used to indicate the direction of the spatial profiles and thus the MTZ. The results can be found in Figure 4-4. Since this concerns a PSA process in CSS, the dynamics in the sieve beds are identical in a time-shifted manner: e.g., while sieve bed 1 is at step 1, sieve bed 2 will be at step 3. Therefore the states in only one sieve bed are shown.

For each state four subplots can be seen, each for a step in the PSA cycle. Starting with the first step, feed, the moving MTZ is characterized by the mole fraction staying longer at ambient levels ($\gamma_{O_2,amb} = 0.21$), because the sieve bed is saturated there. The last blue line is at $t = 5s$, so at the end of the feed step a large part of the sieve bed has not been used, i.e. q_{N_2} is not yet everywhere at its equilibrium value. However, waiting until the complete sieve bed has been saturated is not ideal. Nitrogen breakthrough will occur before this happens, and as a result the mole fraction oxygen in the tank will drop. Examples of this phenomenon will be shown in Section 4-2-3.

It was already explained that the purpose of the balance steps, steps 2 and 4, is to equalize the pressures in the sieve beds. In the second step, the air from the sieve bed is directed to the other sieve bed, and the wave propagates to the left. In the fourth step, the roles

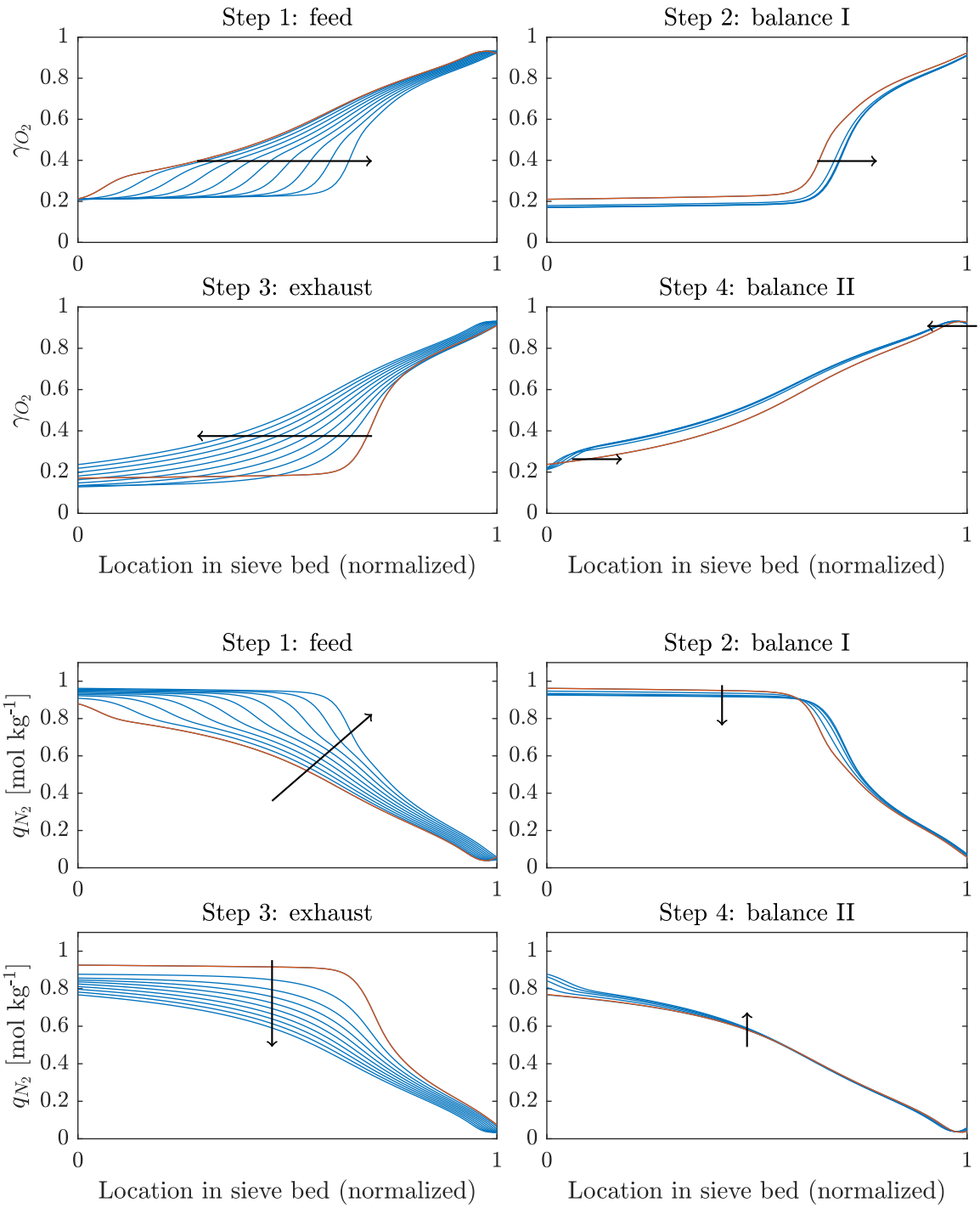

Figure 4-4: Simulated profiles for the O_2 mole fraction (top four plots) and the adsorbed amount of N_2 (bottom four plots), during one cycle at CSS with $\tau_f = 5s$. The red line is the initial profile, and the arrows denote the temporal evolution of the waves. The temporal resolution for steps 1 and 3 is 0.5s, and for steps 2 and 4 ca. 0.08s.

are reversed: this sieve bed receives air from the other sieve bed and from the compressor, denoted by the two arrows pointing in opposite directions in the subplot for γ_{O_2}.

The last step to be discussed is the exhaust step. Here the pressure is lowered in the sieve bed, and as a result the adsorption capacity of the adsorbent is reduced. Nitrogen is released in the adsorber and that can be seen in the profiles of γ_{O_2}: initially, $\gamma_{O_2} < 0.21$ at $z = 0$, meaning that the outgoing air has a larger nitrogen mole fraction than ambient. After some time, the value of $\gamma^1_{O_2}$ will rise above ambient levels, and exhausting this air becomes less and less efficient since you are removing partially oxygen-enriched air.

At the end of the cycle, both the O_2 mole fraction and the adsorbed amount of N_2 are back at their levels from the beginning of the cycle, and the next cycle begins.

4-2-2 CSS Input/Output Dynamics

Recall from Section 3-4, that the two cycle-averaged outputs are the purity, $\mathcal{P}$, and the average compressor power, $\bar{P}^c$. In this section the CSS relationship between these outputs and the input τ_f will be discussed. The reason these so-called cycle-domain outputs are used, and not their time-domain counterparts, is because the time-domain outputs oscillate during a cycle, making it harder to see the Input/Output (I/O) relationship.

In Figure 4-5 the CSS I/O relationships are plotted. For now, the range $\tau_f \in [3, 10]s$ will be evaluated for a better understanding of the PSA dynamics, and in Chapter 5 the operating range for the controller will be explained.

The first I/O relationship, $\mathcal{P}$ and τ_f, is a parabolic one: there is a maximum at $\tau_f \approx 6.5$, because for lower inputs there is not enough time to adsorb the maximum amount of nitrogen. For higher inputs, nitrogen breakthrough starts to occur: the sieve bed gets increasingly saturated, and N_2 leaks through to the oxygen tank, causing the mole fraction O_2 in the tank to drop. Breakthrough will be the subject of the next section.

The CSS I/O dynamics of the average compressor power are almost linear. There is a noticeable rise in gradient at $\tau_f \approx 9s$, and this is due to the increased saturation of the sieve beds. With the sieve beds getting saturated, more and more gas can flow through the sieve bed instead of being adsorbed, resulting in a higher increase in pressure.

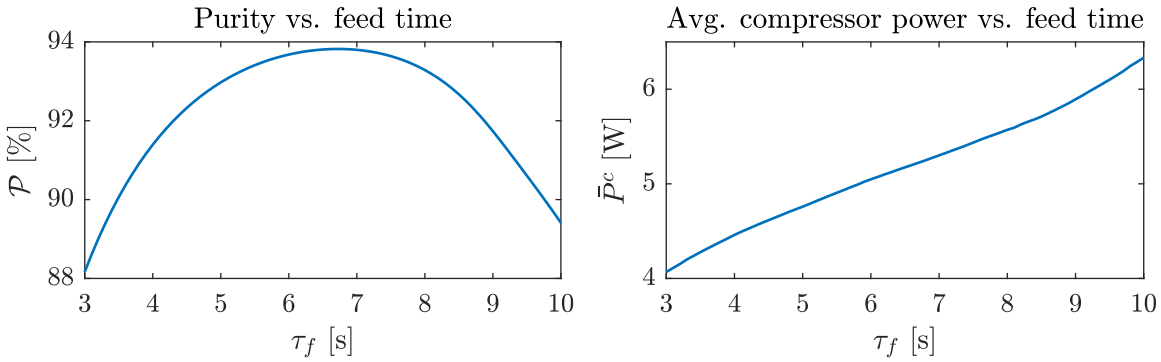

Figure 4-5: Input/Output dynamics of the POC at CSS.

4-2-3 Breakthrough

Nitrogen breakthrough was already briefly mentioned in Chapter 2, but now what a high-fidelity model of the POC has been created, this phenomenon can be explained better.

The dynamics of a nitrogen breakthrough can be seen in Figure 4-6, where the spatial profiles of γ_{O_2} are plotted for different times t in a cycle. The feed time is $\tau_f = 10s$, and normally there is no reason to use such a long cycle time. However, here it is used because the breakthrough effects are clearly visible.

Bearing in mind that the dynamics in both sieve beds are identical during CSS, the dynamics of a sieve bed undergoing nitrogen breakthrough are as follows. There is more interaction between the adsorbers during a cycle with breakthrough, so that is why the sieve bed number will be mentioned too. During the first (feed) step, the concentration wave propagates through adsorber 1. With the longer feed time, this adsorber gets increasingly saturated, resulting in a drop in O_2 mole fraction at the end of the sieve bed. The concentration wave keeps propagating through sieve bed 1 during the second step, and even breaks through into sieve bed 2, which is currently at step 4. The next step for adsorber 1 is the exhaust step, and the gas is exhausted normally. The second sieve bed goes to step 1, but with the concentration wave from the other sieve bed still present. This is the reason for the strange initial profile at $z = 1$ in the plot of step 1 in Figure 4-6. Now the cycle will start again.

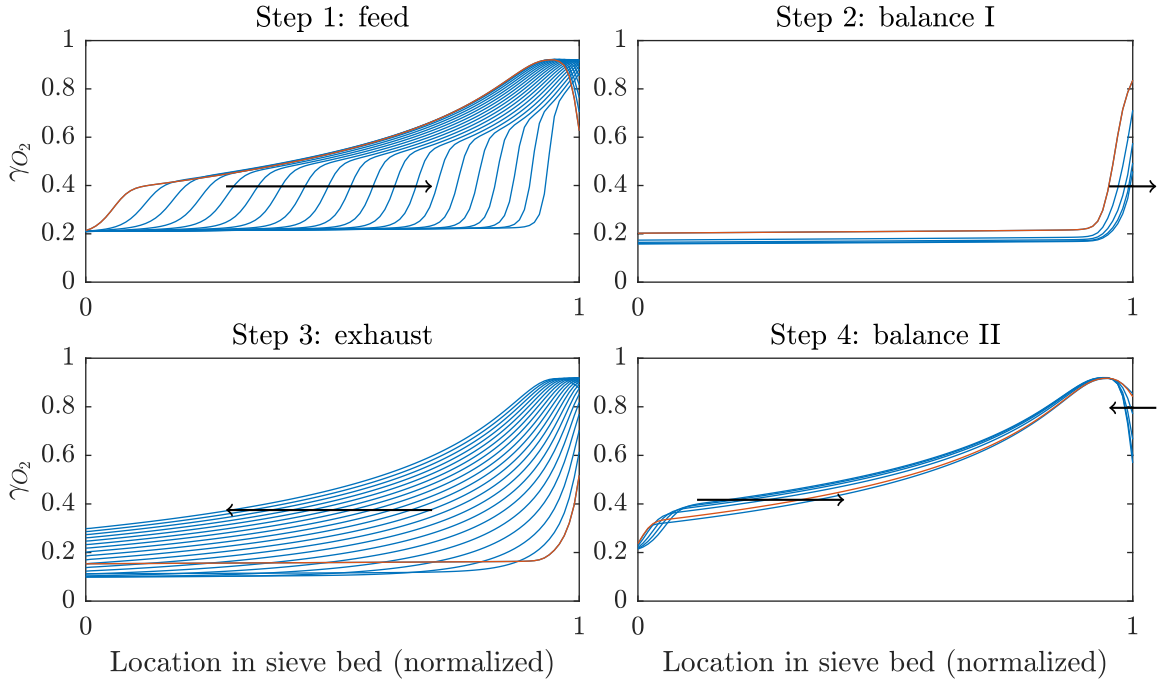

Figure 4-6: Simulated profiles for the O_2 mole fraction during one cycle at CSS with $\tau_f = 10s$, at which nitrogen breakthrough occurs. Initial profiles are denoted with red lines. The temporal resolution is the same as in Figure 2-4.

The result of the breakthrough can be seen in the oxygen tank dynamics, shown in Figure 4-7. Obviously, the average O_2 mole fraction is lower than in a case without breakthrough, e.g. in Figure 4-3. The oscillations are also larger: they will increase in magnitude for larger τ_f.

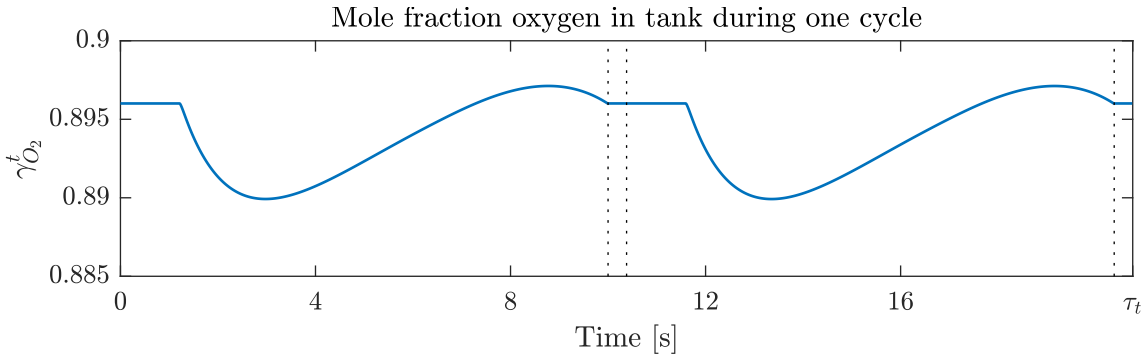

Figure 4-7: Variations in mole fraction O_2 in the oxygen tank during a cycle at CSS with $\tau_f = 10s$.

4-3 Transient Behavior

In this section the transient behavior of the POC will be discussed. To do so, the high-fidelity model has been simulated subject to step changes in the input τ_f. The start-up behavior (i.e. starting with "empty" sieve beds and ambient conditions in the oxygen tank) will not be considered, since for this study the POC is assumed to start at CSS.

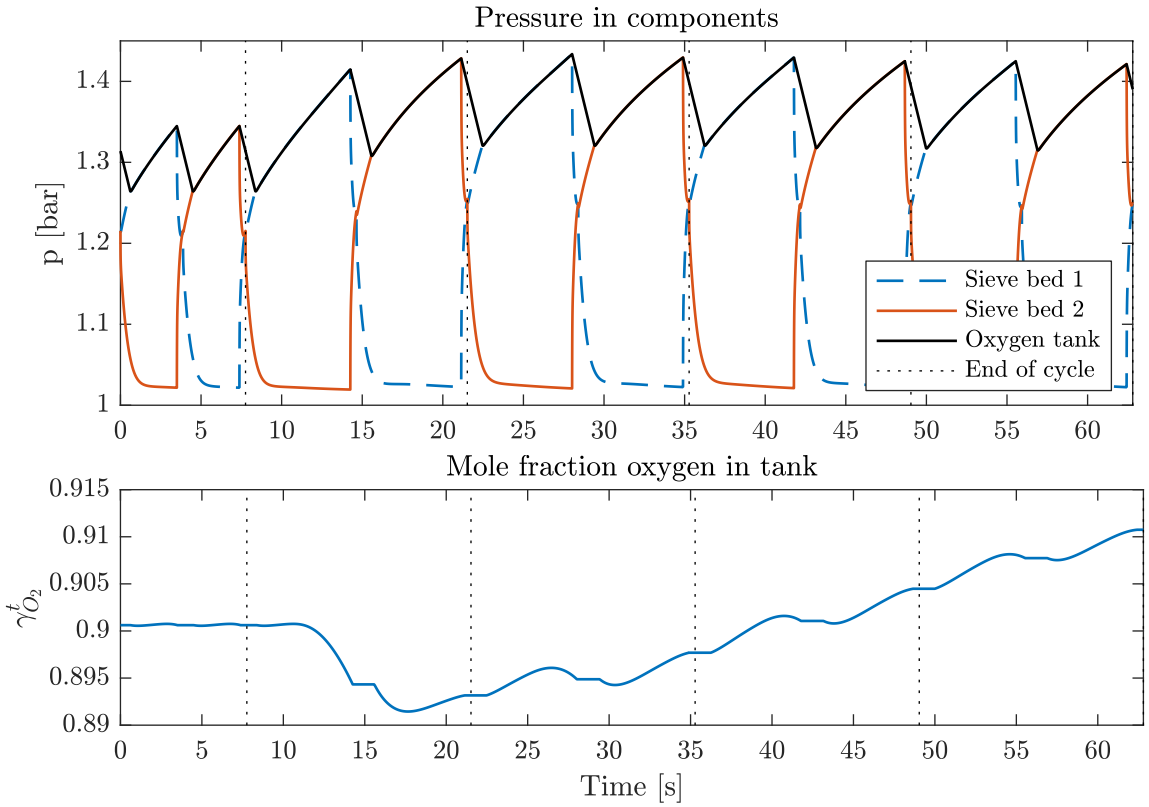

Figure 4-8: Transient behavior of the pressure in the three main components and the mole fraction O_2 in the tank. The system is initiated at CSS with $\tau_f = 3.5s$, and after one cycle a new input is give of $\tau_f = 6.5s$.

First, the response of the time-domain states and outputs will be reviewed: Figure 4-8 shows the result of a step change in feed time. $\tau_f = 3.5s$ initially (with the system at CSS), and after one cycle the feed time is increased to $\tau_f = 6.5s$. The pressure in the components rises immediately, whereas the O_2 mole fraction in the tank has a much slower response. This is due to the oxygen tank, which acts as a filter. However, that does not explain the non-minimum-phase behavior that $\gamma^t_{O_2}$ shows.

Non-minimum-phase systems are systems with right-half-plane zeros in the complex plane (if in continuous time), and this property manifests itself as an "undershoot". During a step response the system output first goes the opposite way before changing direction and going to its steady state value, as can be seen in the figure. The reason this system has non-minimum-phase behavior is as follows. In CSS, the initial profiles of the states in the adsorber are tailored to the corresponding input. After a sudden increase in feed time, the adsorbers need time to adjust, and this causes a nitrogen breakthrough. The non-minimum-phase behavior strongly depends on the input: it is not always this noticeable.

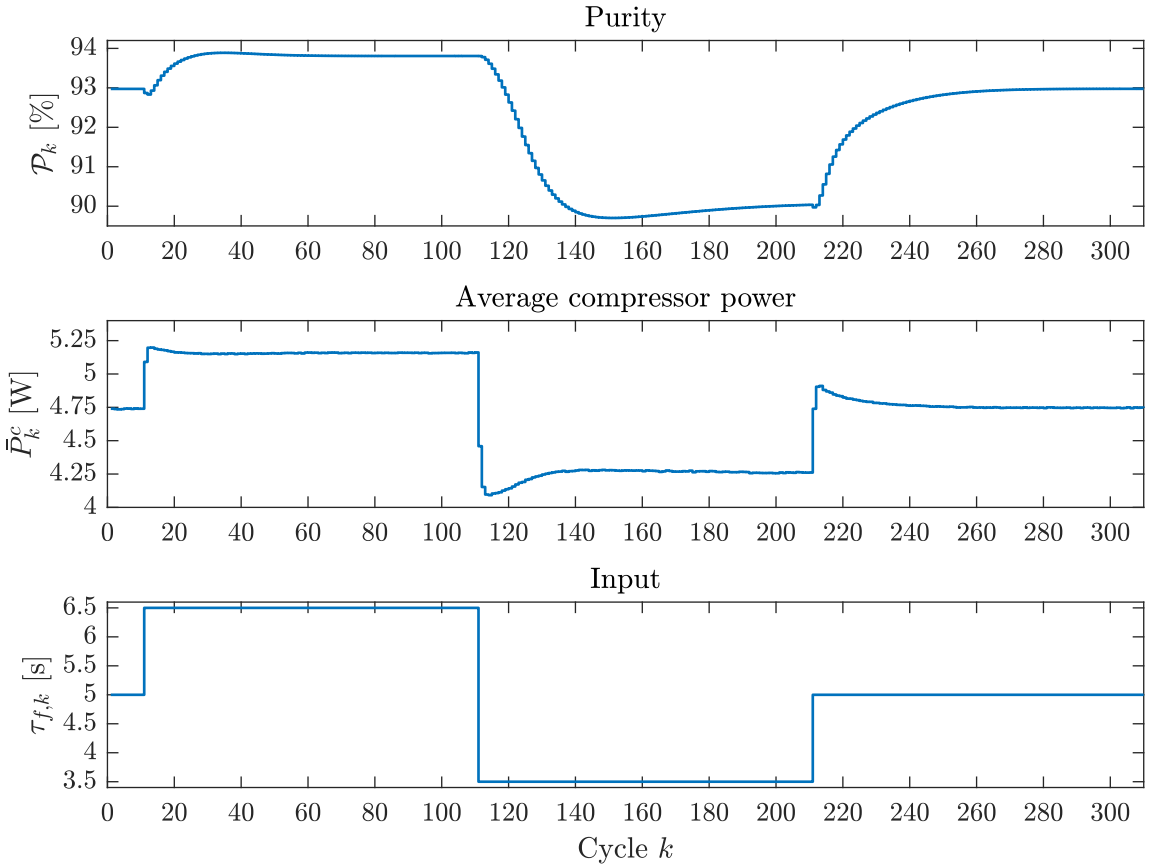

Figure 4-9: Response of the cycle-domain outputs on different step changes in the input τ_f.

To get an idea of the steady-state value of the outputs, it is better to use the cycle-domain outputs $\mathcal{P}_k$ and $\bar{P}^c_k$, given the long settling time. The results of a longer simulation (310 cycles) have been plotted in Figure 4-9, where multiple step changes have been applied to the input. The same dynamics as in Figure 4-8 can be detected: the purity has a very slow response with some non-minimum-phase undershoots, and the average compressor power

reacts immediately, although it does not go directly to its steady state value.

If the signal of the average compressor power is inspected closely, some very small random oscillations can be spotted: it seems to be affected by (white) noise. This is likely because of the variable step ODE solver, and the timer that determines when to switch between steps and end the cycle. Due to the variable steps, the length of a cycle varies within a few milliseconds, and thus the average compressor power will also vary slightly. The purity does not have this problem, because of the filtering effect of the oxygen tank. Increasing the relative error tolerance of the solver did not change this.

4-4 Comparisons with Literature

Comparisons between this model and other dynamical models of POCs or oxygen concentrators in general should be made with an annotation. No study has been found that matches the structure of this POC completely, and therefore the dynamics will differ slightly. The general PSA dynamics, like traveling concentration waves, (nitrogen) breakthrough and the course of the pressure inside a sieve bed will have the same characteristics though. An assessment of the validity of this model will therefore be based on the resemblance between the dynamics of the POC in question and dynamics of POCs in the literature.

The study with the PSA plant that resembles this POC the most is one by Bitzer [20, 3]. In this study a dynamical model is made of a (non-portable) oxygen concentrator, slightly larger than the POC in this thesis. Although it differs in size, the same dynamics in the sieve beds can be seen. The two rightmost plots in Figure 4-10 show some simulation results of this study, and these resemble the concentration waves found in this POC.

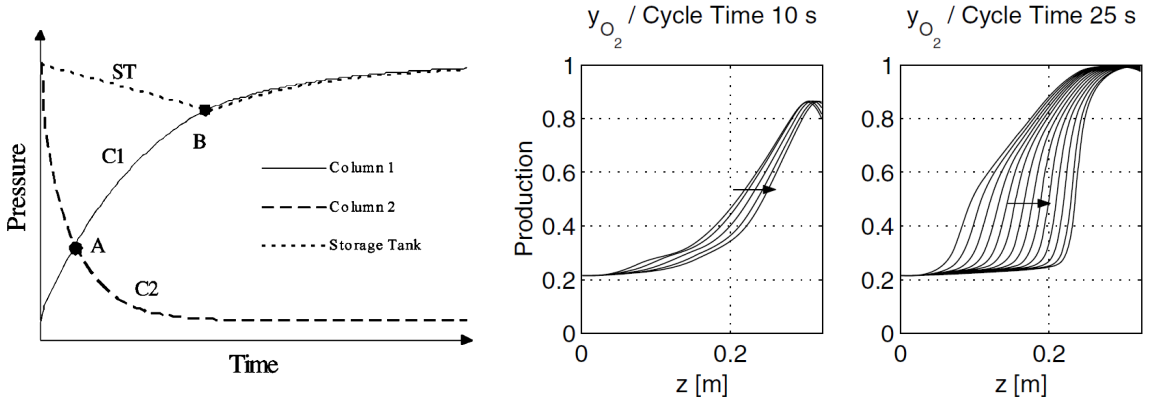

Figure 4-10: Left: development of pressure in a sieve bed. Retrieved from [2]. Right: traveling concentration waves in sieve bed for different cycle times. Retrieved from [3].

Other studies worth mentioning are the ones by Santos et al. [7, 2]. These are about the simulation and optimization of small oxygen PSA units, and closely resemble this POC. The greatest difference is the absence of the pressure equalization steps, and the result of that can be found in the left plot of Figure 4-10. Here the pressure in a sieve bed starts at ambient levels, whereas in Figure 4-3, the pressure starts halfway. Nevertheless, the same concept can

be seen in Figure 4-10: the pressure in the sieve bed increases, until it matches the pressure in the storage tank, and the pressure in the other sieve bed is reduced to ambient levels.

Comparing transient behavior is difficult. For example, Bitzer also shows some transient behavior of the system, but any non-minimum-phase behavior can not be seen. This difference in transient behavior is likely due to the larger size of the oxygen concentrator that is being investigated.

Other studies that also show resemblance are [19, 21, 22], all about PSA for oxygen concentrators.

4-5 Concluding Remarks

In this chapter the steady-state and transient behavior of the POC has been studied. For that reason, the high-fidelity model of Chapter 3 has been used to simulate the system.

The plant dynamics are characterized by traveling waves, which travel back and forth through the adsorbers during a cycle. If the input is held constant, the states of the POC become periodic during each cycle, which is called Cyclic Steady State. Transient behavior is studied with the cycle-domain outputs, and the slow response time of the system has become evident.

The validation of the model has been done by comparing it with other studies about oxygen concentrators. Although there are no studies that have POCs that are identical, the dynamics of these devices adhere to the same principles: the traveling concentration waves can be found, as well as the typical pressure profiles in the sieve beds. This is reason to assume that the high-fidelity model of Chapter 3 shows realistic behavior.

Chapter 5

Batch Modeling

In the previous two chapters, the design and validation of the high-fidelity model has been discussed. Although this model is perfectly suited for simulation, using it for a prediction model for a controller will cause problems, because of the computational load. Therefore, for prediction purposes, a hybrid batch model is derived in this chapter.

The first section is about the background of batch modeling: what exactly constitutes a batch model, why it is used, and how it is used in the literature. The approach taken in this thesis will also be highlighted. Section 5-2 concerns the identification of local batch models, and in Section 5-3 the hybrid part of the model is derived. The chapter will conclude with a summary.

5-1 Background

Before the reasoning behind the use of a batch model is explained, an explanation of what a batch model actually is (in the context of a PSA system) will be given.

A batch model uses cycle-averaged outputs and input(s), to calculate the cycle-averaged outputs of the next cycle. In the case of this POC, the outputs are the purity $\mathcal{P}_k$ and the average compressor power $\bar{P}^c_k$, and the input the feed time $\tau_{f,k}$, all defined in Section 3-4. The aim of this chapter is therefore to find the following function f:

$$y_{k+1} = f(y_k, \tau_{f,k}), \qquad y_k = \begin{bmatrix} \mathcal{P}_k \\ \bar{P}^c_k \end{bmatrix}. \tag{5-1}$$

An illustrative example of this batch model, and the difference between a time-domain output and a cycle-averaged output, can be found in Figure 5-1.

5-1-1 Motivation

There are multiple reasons why a batch model is better suited as a prediction model than a high-fidelity model. The first one is that the high-fidelity model is simply too complex to

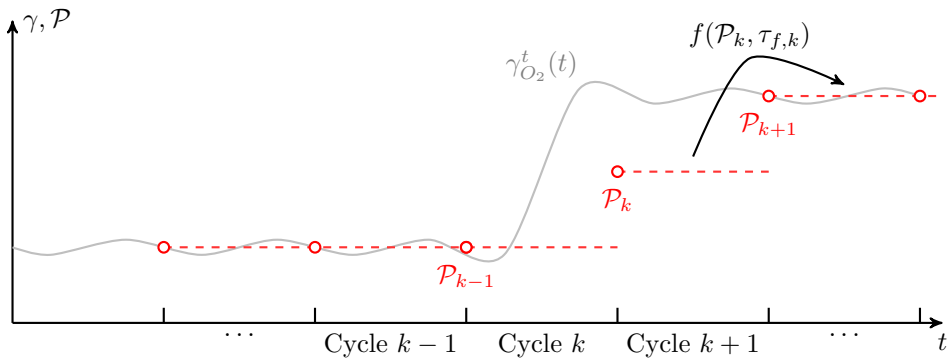

Figure 5-1: Illustrative comparison between the cycle-averaged purity $\mathcal{P}_k$, and the continuous time-domain output of the high-fidelity model, $\gamma^t_{O_2}$. Note that the purity is calculated at the end of the cycle. A batch model is used to find the purity (and also power usage, but that is not shown) of the next cycle given the current input and outputs.

use for real-time control purposes. The sharp wave fronts present in the sieve bed require a relatively low sample time (0.01s), and combined with the large time constant of a PSA system, a controller needs to simulate a very large amount of time steps. Given that this controller has to compute an optimal control move on-line, a simpler model is desirable. This reason is also often cited in the literature [41, 42].

It is possible to counter the first argument by simply using a more powerful processor. However, the available hardware on a POC should also be taken into consideration: ideally the controller would be able to run on the current hardware, which is tailored for the current (computationally cheap) control method (see Chapter 2). So for that reason, again a simple model would be best. One can argue that e.g., order-reduction, like is done in [24], would also reduce the complexity of the high-fidelity model, making it possibly suited for prediction purposes. However, recalling the step responses in Chapter 4, the dynamics of the time-domain outputs in Figure 4-8 are less linear than the cycle-domain outputs shown in Figure 4-9. So it is likely that a batch model will be much simpler than such a reduced-order model.

5-1-2 Batch Models in Literature

There is no absence of batch models in the PSA literature. Two kinds of batch models feature prominently: linear models [41, 42] and Hammerstein models [3, 43]. Linear models are of the standard form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k + Du_k, \end{aligned} \tag{5-2}$$

with x_k the states, y_k the output, u_k the input and A, B, C, D the system matrices. Hammerstein models are an extension on linear models. Instead of directly using the input u_k, a static nonlinear map is used that maps the true input u_k to a virtual input w_k:

$$w_k = f(u_k), \tag{5-3}$$

which is then used in Eq. (5-2) instead of u_k. The advantage of this static input mapping is that it adds some extra flexibility to the model, whilst keeping the dynamics linear.

Like said before, the PSA plants of the studies in which these batch models have been found, differ from the PSA structure found in the POC. Take the study by Khajuria and Pistikopoulos [41]: the plant here is an industrial H_2, CH_4 PSA separation system, with four adsorbers of 0.6m in size and an operating range of 99.975 - 100% H_2 purity. An 8^{th} order linear state-space batch model was identified using Pseudo-Random Binary Sequence (PRBS)-like signals, with a fit of 82%.

Urich et al. [42] also used a linear model for a batch model of a single-sieve bed RPSA POC. Single sieve bed PSA systems are rare, and according to the authors such a system enabled the use of a linear batch model. A fit of 92% was obtained for the oxygen concentration.

The use of Hammerstein models has been described in the studies by Bitzer [3] and Peng et al. [43]. Both studies are on O_2, N_2 separation. Unfortunately both authors do not go into their identification procedures, nor do they discuss validation of their batch models.

5-1-3 Approach

As the results will show, the approaches from Section 5-1-2 are not directly suited for this POC. The input/output dynamics are not as linear as other PSA plants, possibly due to its small size and relatively short PSA cycle. A reason for this could be that larger PSA plants (with larger storage tanks) are better able to filter out the nonlinear wave dynamics than a smaller plant.

In order to solve this problem, the following approach has been taken. Firstly, the input range of the system is reduced. The same tactic has been used by Khajuria [41], who only exited the respective PSA plant for feed times $\tau_f \in [78, 88]s$. Because of the difference in sizes between that PSA plant and the POC, a direct comparison cannot be made. The choice has been made to use the following input range $\mathcal{U}$ in this thesis:

$$\mathcal{U} = \{\tau_f \mid 3.5 \geq \tau_f \geq 6.5\}. \tag{5-4}$$

The reasoning behind this choice is as follows. Recall Figure 4-5 in Chapter 4, where the I/O dynamics in CSS were shown. The lower bound has been chosen because for lower τ_f the CSS purity is below the minimum value, 90%. The upper bound is the maximum attainable purity for this POC: after that nitrogen breakthrough starts to occur, and it is therefore not feasible to use inputs larger than the upper bound.

The second part of the approach is to use a hybrid model. Hybrid systems are systems consisting of both continuous and discrete dynamics, e.g. a system that switches between different linear models given a switching parameter. This hybrid model consists of multiple local models (or submodels), each local model identified for a convex subset of the input space, $U_m \subseteq \mathcal{U}$, with every set disjoint from the others. A controller is then able to switch between these submodels, depending on the desired input. This will be further explained in Section 5-3.

Looking back at Section 5-1-1, it was stressed that this batch model should be as simple as possible. By using linear models or Hammerstein models as local models, the complexity of the hybrid model is kept to a minimum. Moreover, their usage has been proven in the literature. Identification and validation of these local models will be the subject of the next section.

5-2 Local Model Identification

Identification of the submodels is done by exciting the high-fidelity model of Chapter 3 with known input data in open-loop, recording the corresponding output data, and finding the Input/Output relationship with identification methods.

In this section, two modeling decisions will be explained. The first is the type of local model (Hammerstein or linear), and the second is the amount of local models used, i.e. the degree of division of the input space.

The optimal amount of submodels will be determined empirically. Three sets of submodels are identified: one where a single model is used for the complete range, one where three submodels are used, and one with 13 local models. The input range $\mathcal{U}$ remains the same for all three sets.

5-2-1 Identification Methods

The linear models are identified using a subspace identification method called Numerical algorithm for Subspace IDentification (N4SID). Other methods exist for identification of linear models, but because of personal experience with N4SID this method has been chosen. Now a brief explanation of this method will follow, but for an in-depth description the reader is referred to [44], of which the following is also extracted.

Subspace methods use two so-called Hankel matrices, one composed of (the identification) input data and the other of the corresponding output data, to construct subspaces that are related to the system matrices of the state-space model. The system model is then obtained in a non-iterative way via the solution of a number of linear-algebra problems. This means no optimization routine is required for this identification method. The System Identification Toolbox (SIT) of MATLAB [45] contains a function called `n4sid` which is used for identification. Chosen parameters will be discussed in the next section.

There is much literature available on identification of Hammerstein models, e.g. [46, 47]. However, instead of implementing an identification method of one of these studies, the SIT function `nlhw` has been used which shows appropriate results. This is an optimization-based function, i.e. the model parameters are estimated by iteratively solving an optimization problem.

In Eq. (5-3), the nonlinear map of the Hammerstein model was not defined. It could be any nonlinear function, but an often used function is a polynomial expansion [48], which will also be used in this study:

$$f(u_k) = \sum_{i=0}^{N} \alpha_i u_k^i. \tag{5-5}$$

Here, N is the order of the polynomial, and α_i the coefficients which need to be optimized in addition to the state-space matrices to achieve the best fit.

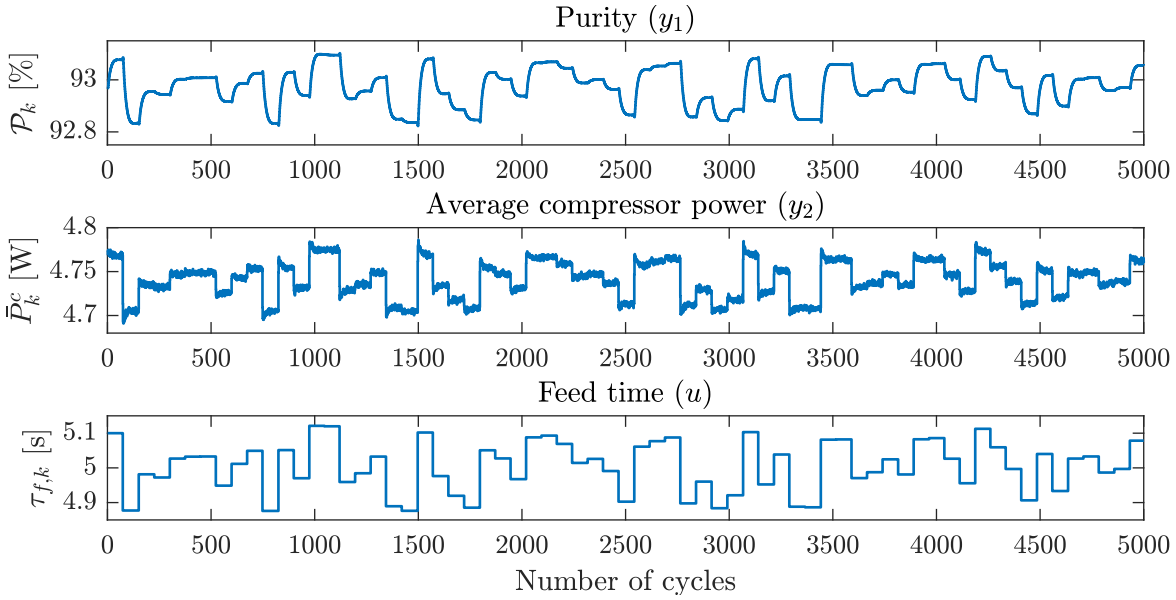

Figure 5-2: Part of the identification data for the local modal with range $\tau_f \in [4.875, 5.125]$.

5-2-2 Generation of Data Sets

In this section the generation of the identification data sets will be explained. The most important aspect of this process is the choice of input sequence: the signal should be persistently exciting, i.e. excite all dynamics of a system. For that reason, random signals like the previously mentioned PRBS are ideal.

In total, 16 identification data sets have been created, one for each (local) model. For clarity, each data set is denoted with $\mathcal{I}_{U_m}^M$, where M stands for the set of submodels it belongs to (i.e. $M = 1$ for the global model, $M = 3$ for the model consisting of 3 submodels, and $M = 13$ for the model consisting of 13 submodels). U_m is the range or input space of that specific data set. So e.g. the first data set for the model with 13 submodels is denoted with $\mathcal{I}_{\tau_f \in [3.375, 3.625]}^{13}$.

With that having said, some care has to be taken to make the input signal persistently exciting for the POC. Three properties are crucial [13]:

- The maximum change in signal value $|\Delta\tau_{f,k}| = |\tau_{f,k} - \tau_{f,k-1}|$: a very large change can perturb the POC beyond the scope of fitting a linear model. Therefore the maximum change has been set to $|\Delta\tau_{f,k}| = 0.25$.
- The switching time for the input τ_f, called t_{switch}. This is the time for which the signal stays constant before switching to another random value. If the switching time is set too low, low frequency dynamics of the PSA system may not be captured. The value of this switching time is $t_{switch} = 800s$, slightly longer than the settling times for different values of τ_f. After this time, the current cycle will finish and then a new input is fed to the system.
- The length of the input sequence. Each input sequence is measured in amount of input changes, with each input held for a minimum of 800s. The input sequences of $\mathcal{I}_{U_m}^{13}$ all have 400 input changes, the input sequences of $\mathcal{I}_{U_m}^{3}$ all have 800 input changes, and the input sequence for the global model, with data set $\mathcal{I}_{\tau_f \in [3.4, 6.6]}^{1}$, has 1000 input changes.

Proving that input sequences are in fact persistently exciting is possible (at least for linear systems [49]), but in this thesis this has not been done. Instead the identification results are used: if the fits of the local models are high enough, the amount of excitation is deemed sufficient. This will be discussed in Section 5-2-5.

In Table 5-1 all the ranges for the identification data sets are shown, along with the fits. The reason some data sets have ranges that exceed the global range $\mathcal{U}$, defined in Eq. (5-4), is to also excite the system properly at the edge of the range. As an example, a part of the identification data set $\mathcal{I}^{13}_{\tau_f \in [4.875,5.125]}$ is shown in Figure 5-2.

To ensure that there is no case of overfitting on the identification data, each data set will be split up in a validation subset and a identification subset. Recall that the input sequences are measured in input changes, each input held for a minimum of 800s. The first 75% of these input changes (and the corresponding outputs) are used for identification, and the remaining 25% are used for validation. So e.g., any data set $\mathcal{I}^{13}_{U_m}$ has a identification subset with 300 input changes and a validation subset with 100 input changes.

5-2-3 Determining Model Properties

There are some properties that need to be determined before identification. These are the order of the linear models (or the order of the linear part of the Hammerstein models), if there is any direct input feedthrough to the output (a D matrix), and the order of the input polynomial for the Hammerstein model.

Model Orders

When identifying a system, the order has to be chosen manually. Sometimes this can be done by analyzing the physical system and counting the states, but since there are no tangible states in a batch model, it has to be estimated in a different way. For that reason, estimating the order of the linear batch models is done with Hankel Singular Values. These singular values provide a measure of the energy stored in each state of the system, and are calculated as follows [50], given a stable state-space system:

$$\sigma_{H,i} = \sqrt{\lambda_i(\mathcal{W}_o \mathcal{W}_c)}, \tag{5-6}$$

here λ_i are the eigenvalues of the product of $\mathcal{W}_o$ and $\mathcal{W}_c$, the observability and controllability Gramians. These Gramians can be computed by solving the (discrete time) Lyapunov equations:

$$A\mathcal{W}_c A^T - \mathcal{W}_c + BB^T = 0, \tag{5-7a}$$

$$A^T\mathcal{W}_o A - \mathcal{W}_o + C^T C = 0. \tag{5-7b}$$

Of course, the system matrices are not yet known. To get an idea of the order, a system with a model order larger than the true model order is identified. The actual order is then estimated by looking at where the biggest drop in "energy" is, i.e. which states are part of the system and which states are noise.

An initial guess was made of order 20, and if no clear distinction was visible between noise and system states, another attempt would be made with a larger order. However, this was not needed, as can be seen in Figure 5-3. Here the Hankel singular values have been plotted for three datasets. Looking at the figure, clear drops can be seen between the 5^{th} and 6^{th} singular values for data sets 1 and 2, and a drop between the 4^{th} and 5^{th} singular value for the third data set. This indicates that a 4^{th} or 5^{th} order state-space model would be a good initial guess. The best results were obtained with 5^{th} order models, so that order has been used for all local models.

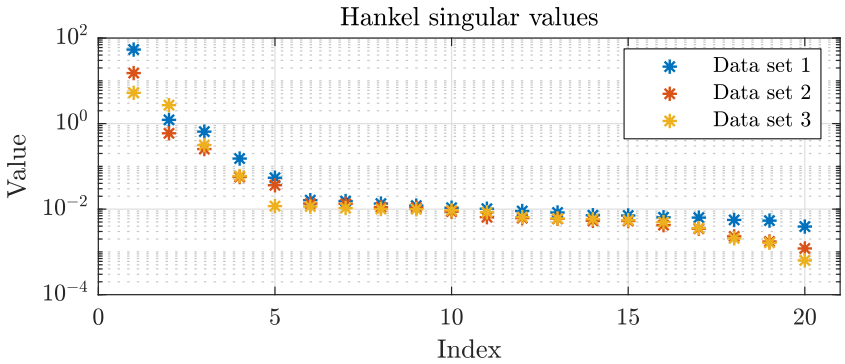

Figure 5-3: Hankel Singular values of three datasets.
Data set 1 $= \mathcal{I}^{13}_{\tau_f \in [3.375, 3.625]}$. Data set 2 $= \mathcal{I}^{13}_{\tau_f \in [4.875, 5.125]}$. Data set 3 $= \mathcal{I}^{13}_{\tau_f \in [6.375, 6.625]}$.

For the identification of the Hammerstein models, more information of the state-space models needs to be specified, namely the number of zeros, number of poles and the input delay. An alternative is to provide the optimization algorithm with an initial linear model. Delays and the number of poles and zeros are then extracted from this model. This is the chosen approach: every Hammerstein model will be "initiated" with the corresponding linear model, so the order of the Hammerstein models is also five.

Direct Feedthrough

Although the two linear batch models in the literature [41, 42] both did not have any direct input feedthrough to the output, in this study another conclusion is drawn.

Not having a D matrix would mean the system has a input delay of one step, i.e. any change in input is not directly visible. Looking at Figure 5-2, and then especially at $\bar{P}^c_k$, the effect of an input change is directly noticeable in the output. For that reason a D matrix is added to the state-space system. This can also be explained with reasoning: if the input is changed during a cycle, e.g. τ_f is lowered from 6 to 3 seconds, the average compressor power will also drop immediately instead of waiting one cycle.

Static Input Nonlinearity

The order of the polynomial from Eq. (5-5) should also be chosen before identification. It is set to 2, since any higher orders did not improve the fit of the Hammerstein models:

$$f(u_k) = \alpha_0 + \alpha_1 u_k + \alpha_2 u_k^2. \tag{5-8}$$

5-2-4 Data Preprocessing

Before the datasets can be used for identification, some preprocessing is needed. Namely, a linear model is not able to capture any static offset the in the identification data, and therefore the data needs to be detrended. The two offsets are the means of the input and output-data: $\bar{U}_m$ and $\bar{Y}_m$, respectively. After identification, the two offsets are added back to all local state-space models. This is done by augmenting the models with an extra state, $x_{o,k} = 1 \; \forall k$:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ x_{o,k+1} \end{bmatrix} &= \begin{bmatrix} A_m & -B_m\bar{U}_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ x_{o,k} \end{bmatrix} + \begin{bmatrix} B_m \\ 0 \end{bmatrix} u_k, \\ y_k &= \begin{bmatrix} C_m & (\bar{Y}_m - D_m\bar{U}_m) \end{bmatrix} \begin{bmatrix} x_k \\ x_{o,k} \end{bmatrix} + D_m u_k, \end{aligned} \tag{5-9}$$

and similarly for the dynamic part of the Hammerstein models. Note that in Eq. (5-8) there is also a constant term, α_0. In theory this term could be used to account for the input offset, but better fits were obtained if the offsets were removed from the data before identification.

5-2-5 Results

The identification results have been obtained as follows. The identification subsets of every data set $\mathcal{I}^M_{U_m}$ have been used to identify both linear and Hammerstein models, the former with the MATLAB function `n4sid` and the latter with the MATLAB function `nlhw` (see Section 5-2-1). As a measure of fit for both the identification and validation subsets, the Normalized Root Mean Square Error (NRMSE) is used:

$$\text{fit} = 100\left(1 - \frac{||y - \hat{y}||}{||y - \bar{y}||}\right). \tag{5-10}$$

Here y is the true output of the high-fidelity model, $\hat{y}$ the output of the identified model and $\bar{y}$ is the mean of y.

The fits can be found in Table 5-1. Different observations can be made looking at these results. One might have suspected that the Hammerstein models would outperform the linear models, and they do, but barely. Another interesting point is the fact that the average compressor power can be captured adequately by a global model, but the purity can not. This is most likely due to the (nonlinear) breakthrough and saturation effects that the oxygen mole fraction experiences.

In Section 5-1-3 it was already hinted that a global model would not work for this POC. To substantiate this, the global model (and as a comparison also some local models) have been validated using validation subsets of $\mathcal{I}^{13}_{U_m}$. The goal is to show that the fit of the global model is even worse when looking at a very narrow input range, and this can be seen in Table 5-2, where the validation subset of $\mathcal{I}^{13}_{\tau_f \in [3.625, 3.875]}$ is used. It is now clear a global model is not an option for this POC: fits of 41% are not acceptable.

Now that the option for a global model has been eliminated, the choice for the type of model and amount of submodels can be explained. This choice has been made with the hybrid

Amount of local models	Range U_m (τ_f [s])	Linear model fit [%]				Hammerstein model fit [%]			
		Identification		Validation		Identification		Validation	
		$\mathcal{P}_k$	$\bar{P}_k^c$	$\mathcal{P}_k$	$\bar{P}_k^c$	$\mathcal{P}_k$	$\bar{P}_k^c$	$\mathcal{P}_k$	$\bar{P}_k^c$
13	$[3.375, 3.625]$	97.19	93.0	97.08	92.74	98.68	93.25	98.79	92.9
	$[3.625, 3.875]$	96.9	92.72	96.97	93.05	98.8	92.83	98.8	93.12
	$[3.875, 4.125]$	97.71	91.89	97.61	92.01	98.86	92.33	98.84	92.55
	$[4.125, 4.375]$	97.36	91.25	97.22	90.9	98.91	91.0	99.01	90.68
	$[4.375, 4.625]$	97.32	90.8	97.14	90.8	98.98	90.81	98.87	90.77
	$[4.625, 4.875]$	97.35	90.46	97.15	90.85	99.02	90.44	99.0	90.82
	$[4.875, 5.125]$	97.46	90.91	97.55	91.36	99.05	90.45	98.92	90.91
	$[5.125, 5.375]$	96.87	90.95	96.59	90.85	98.93	90.75	98.86	90.59
	$[5.375, 5.625]$	97.14	92.15	96.96	91.73	98.71	91.84	98.68	91.43
	$[5.625, 5.875]$	96.38	91.32	96.1	91.05	98.35	90.53	98.31	90.25
	$[5.875, 6.125]$	94.2	89.43	93.87	89.77	97.65	89.83	97.64	90.23
	$[6.125, 6.375]$	94.14	89.86	94.12	90.36	96.21	89.3	96.15	89.74
	$[6.375, 6.625]$	91.26	91.08	92.27	91.49	92.76	90.48	93.33	90.85
3	$[3.4, 5.0]$	85.04	93.68	84.78	93.74	88.44	97.79	89.79	97.94
	$[5.0, 5.9]$	87.41	97.74	87.21	97.67	90.6	96.45	90.38	96.44
	$[5.9, 6.6]$	82.16	95.64	82.72	95.69	86.7	95.34	86.97	95.62
1 (global)	$[3.4, 6.6]$	72.32	96.37	69.42	94.23	73.74	96.83	66.43	94.76

Table 5-1: Identification and validation fits of linear and Hammerstein batch models. The three different sets of local models are separated.

framework in mind: although it is theoretically possible to combine some hybrid approaches with Hammerstein models, it adds complexity which can be avoided. Linear models offer almost the same fit, if enough local models are used. That is why, for this study, a batch model consisting of 13 linear submodels is chosen. In Appendix A-1, the state-space parameters of these local models can be found.

Model corresponding to data set	Validation fits [%]			
	Linear model		Hammerstein model	
	$\mathcal{P}_k$	$\bar{P}_k^c$	$\mathcal{P}_k$	$\bar{P}_k^c$
$\mathcal{I}^{13}_{\tau_f \in [3.625,3.875]}$	96.97	93.05	98.79	93.1
$\mathcal{I}^{3}_{\tau_f \in [3.4,5.0]}$	74.43	85.61	72.87	90.91
$\mathcal{I}^{1}_{\tau_f \in [3.4,6.6]}$	41.14	73.01	-235.1	14.91

Table 5-2: Fits of all viable models on the validation subset of $\mathcal{I}^{13}_{\tau_f \in [3.625,3.875]}$.

5-3 Hybrid Models

Now that the choice of submodels has been explained, the hybrid aspect of the batch model will be discussed. This section will cover some options on how to "stitch" the local models together, and explain the reasoning behind the chosen approach.

The three hybrid models that are considered will be introduced in Section 5-3-1, and in Section 5-3-2 the final choice is explained.

5-3-1 Hybrid Models in Literature

The following hybrid models have been researched: hybrid automata, Linear Parameter Varying (LPV) systems, and Piecewise Affine (PWA) systems.

Hybrid Automaton

The hybrid automaton is a very general description of a hybrid system, and often considered to be the formal model for hybrid systems. The following is extracted from [51] and modified slightly to depict the case of this POC batch model.

The structure of a hybrid automaton can be seen in Figure 5-4. Let $q_m \in \mathcal{Q}$ be the local linear models with input ranges $\mathcal{U}_m$, $G(q_m, q_n)$ the guard conditions, and $R(q_m, q_n) : \mathcal{X} \to M(\mathcal{X})$ the reset maps. Here $\mathcal{X}$ is the set of possible states. An example of a guard condition is the following:

$$\begin{aligned} G(q_1, q_2) &= \{u_k \mid u_k > \beta\}, \\ G(q_2, q_1) &= \{u_k \mid u_k \leq \beta\}. \end{aligned} \tag{5-11}$$

That is, the transition between models takes place whenever the input passes a certain value β. To prevent excessive switching between models, the guard condition is sometimes extend with a time-based condition, called dwell time (t_{dwell}). Switching can then only take place after waiting for t_{dwell} time units.

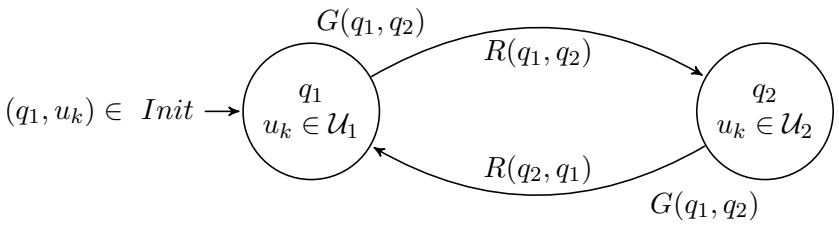

Figure 5-4: Example of a hybrid automaton with two local (not necessarily) linear models q_1 and q_2. The system is initialized at q_1, and the guard condition G controls the switching between models. If switching takes place, states are passed through a reset map R.

LPV Systems

LPV models are, as the name suggests, linear models where the system matrices are parameter dependent. This parameter p can be a function of a state, input, exogenous variable, or combinations hereof, but in this thesis it is only based on the input:

$$\begin{aligned} x_{k+1} &= A(p(u_k))x_k + B(p(u_k))u_k, \\ y_k &= C(p(u_k))x_k + D(p(u_k))u_k, \end{aligned} \tag{5-12}$$

Identification of these kinds of models can be done by identifying multiple linear models for known parameters p, and then fitting functions for the parameter-dependent values in the state-space matrices. However, this requires all identified models to be in a coherent state-space realization.

Different state-space realizations have been used in the literature, e.g. the Controllable Canonical Form (CCF) [52], or balanced realizations using the observability and controllability Gramians [53]. Another proposal by Paijmans et al. [54] is a series interconnection of coherent low-order state-space submodels. Some of these methods require the manual sorting of eigenvalues, which can be tedious for high-order systems.

PWA and PWL Systems

A PWA system is a special case of hybrid automaton, where the dynamic equations and switching rules are linear functions of the state (and sometimes input) [55, 56]. Piecewise affine systems are described by the following state-space equations:

$$\begin{aligned} x_{k+1} &= A_m x_k + B_m u_k + f_m, \\ y_k &= C_m x_k + D_m u_k + g_m, \end{aligned} \qquad \text{for } \begin{bmatrix} u_k \\ x_k \end{bmatrix} \in \mathcal{X}_m \ , \tag{5-13}$$

where $\{\mathcal{X}_m\}_{m=1}^{s}$ is a partition of the state + input set, and f_m, g_m are constant vectors. If these are null, the system is Piecewise Linear (PWL). Each subsystem is defined by the 6-tuple $\{A_m, B_m, C_m, D_m, f_m, g_m\}, m \in \{1, \ldots, m\}$.

An additional requirement, not always explicitly stated, is that the states are continuous on the boundaries between regions m [57, 58]. This complicates matters, since this is not the case with the local models described in Section 5-2.

5-3-2 Choice of Hybrid Model

Ideally, a "simple" hybrid model should be used for the batch model. Literature on control of PWL and LPV systems is extensive, so those are preferred over the more general hybrid automaton.

However, with they way the local models have been identified, an LPV model is not feasible. This is due to the input (and output) offsets that have to be added back to the state-space models. Recall Eq. (5-9), where the state-space matrices were extended with the input offset. After applying the parameterization, the difference equation for the states becomes:

$$\begin{bmatrix} x_{k+1} \\ x_{o,k+1} \end{bmatrix} = \begin{bmatrix} A(p(u_k)) & -B(p(u_k))\bar{U}(p(u_k)) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ x_{o,k} \end{bmatrix} + \begin{bmatrix} B(p(u_k)) \\ 0 \end{bmatrix} u_k. \tag{5-14}$$

Since the input offset is the mean of the used identification input set, the parameterized offset becomes an approximation of the input itself: $\bar{U}(p(u_k)) \approx u_k$. This means the B matrix is canceled out (recall that $x_{o,k} = 1 \ \forall k$), and the input now only appears indirectly in the A matrix. Therefore the steady-state response may still be acceptable, but the transient response will not. So another type of hybrid model must be used.

Concerning the other preferred option, a PWL system, a small look ahead must be made to Chapter 6. It was already mentioned in Section 5-3-1 that the requirement of continuous states at the boundaries was not met for this batch model. Since no literature was found that proposed a control strategy for PWL or PWA systems with such discontinuities, the choice was made to not implement this type of model.

With two of the three models eliminated, the remaining option is to use a hybrid automaton as a hybrid model. The downside of such a model are the nonlinear reset maps, hindering the use of linear control methods, but the advantage of a hybrid automaton is its great flexibility. Since no other feasible model structures were found in the literature, the choice has been made to use a hybrid automaton.

The local models q_m will be the 13 linear models identified in Section 5-2, and now the guard conditions G and the reset maps R will be defined.

Guard Conditions

The guard conditions are simple: switch to whichever subsystem is in the range of the input. For each $m \in [1, 13]$, the guard conditions are:

$$G(q_m, q_n) = \{u_k \mid u_k \in U_n\}, \qquad n = [1, \dots, 13] \setminus \{m\}. \tag{5-15}$$

Reset Maps

The reset maps are as follows. Only the first output (purity) and the corresponding parts of the state space matrices C_n and D_n are used for the reset map. Given the output of the current local model $y_{k,m}$, and the output of the next local model $y_{k,n}$, a map R should ensure the states are reset in such a manner that the two outputs are identical:

$$y_{n,k} = y_{m,k}, \tag{5-16a}$$

$$C_n x_{n,k} + D_n u_k = y_{m,k}. \tag{5-16b}$$

In other words, find states $x_{n,k}$ such that $y_{n,k} = y_{m,k}$. Every reset map $R(q_m, q_n)$ calculates these states as follows. By using the output of five time steps $[y_{k-5,m}, \dots, y_{k,m}]$, a system of equations can be set up:

$$\begin{bmatrix} C_n x_{n,k-4} \\ \vdots \\ C_n x_{n,k} \end{bmatrix} + \begin{bmatrix} D_n u_{k-4} \\ \vdots \\ D_n u_k \end{bmatrix} = \begin{bmatrix} y_{m,k-4} \\ \vdots \\ y_{m,k} \end{bmatrix}, \tag{5-17a}$$

$$\begin{bmatrix} C_n x_{n,k-4} \\ C_n \left(A_n x_{n,k-4} + B_n u_{k-4}\right) \\ \vdots \\ C_n \left(A_n^4 x_{n,k-4} + \sum_{l=k-4}^{k-1} A_n^{k-l-1} B_n u_l\right) \end{bmatrix} + \begin{bmatrix} D_n u_{k-4} \\ D_n u_{k-3} \\ \vdots \\ D_n u_k \end{bmatrix} = \begin{bmatrix} y_{m,k-4} \\ y_{m,k-3} \\ \vdots \\ y_{m,k} \end{bmatrix}. \tag{5-17b}$$

Moving the unknown terms to the left-hand side and the known terms to the right-hand side, this becomes:

$$\underbrace{\begin{bmatrix} C_n \\ C_n A_n \\ \vdots \\ C_n A_n^4 \end{bmatrix}}_{\mathcal{O}} x_{n,k-4} = \underbrace{\begin{bmatrix} y_{m,k-4} \\ y_{m,k-3} \\ \vdots \\ y_{m,k} \end{bmatrix} - \begin{bmatrix} D_n u_{k-4} \\ D_n u_{k-3} \\ \vdots \\ D_n u_k \end{bmatrix} - \begin{bmatrix} 0 \\ C_n B_n u_{k-4} \\ \vdots \\ C_n \sum_{l=k-4}^{k-1} A_n^{k-l-1} B_n u_l \end{bmatrix}}_{v}, \tag{5-18}$$

where v denotes a vector composed of the known terms. Let the first five columns of $\mathcal{O}$ be denoted by $\mathcal{O}_{*,[1,\dots,5]}$, and the last column by $\mathcal{O}_{*,6}$. The states $x_{n,k-4}$ are then calculated as:

$$x_{n,k-4} = \begin{bmatrix} \mathcal{O}_{*,[1,\dots,5]}^{-1} (v - \mathcal{O}_{*,6}) \\ 1 \end{bmatrix}. \tag{5-19}$$

The reasoning behind this is as follows. Recall from Section 5-2-4 that the (5^{th} order) models all have been augmented with an extra state to account for the input and output offsets. Since this state is already known, it is not necessary to calculate this state. Note that $\mathcal{O}_{*,[1,\dots,5]}$ is the observability matrix of the unaugmented system (using only the first output), and since these are all observable and thus have full rank, the inverse exists. The inputs $[u_{k-4}, \dots, u_k]$ are subsequently used (along with $x_{n,k-4}$) to calculate the values of the states $x_{n,k}$.

Problems may arise if the used measurements correspond to inputs that are not in (or close to) the range of the new model q_n, since that model is not identified for that range. This has been countered by constraining the rate of change of the input and will be discussed in Chapter 6.

In Figure 5-5 a comparison can be seen between the high-fidelity model and the hybrid batch model, using the mentioned local linear models, guard conditions and reset maps.

5-4 Concluding Remarks

In this chapter the development of a hybrid batch model has been discussed. Because the cycle-to-cycle dynamics of the POC are nonlinear, a single linear or Hammerstein model is not sufficient.

Therefore, a hybrid automaton has been designed. Each local model is a linear model, valid for a subset of the input space $\mathcal{U}$. The local models have been identified using input-output data of the high fidelity model, and validation results show sufficiently high fits.

The local models are connected through each other with guard conditions and reset maps; the reset maps ensure the states of the batch model are reset when switching between two local linear models.

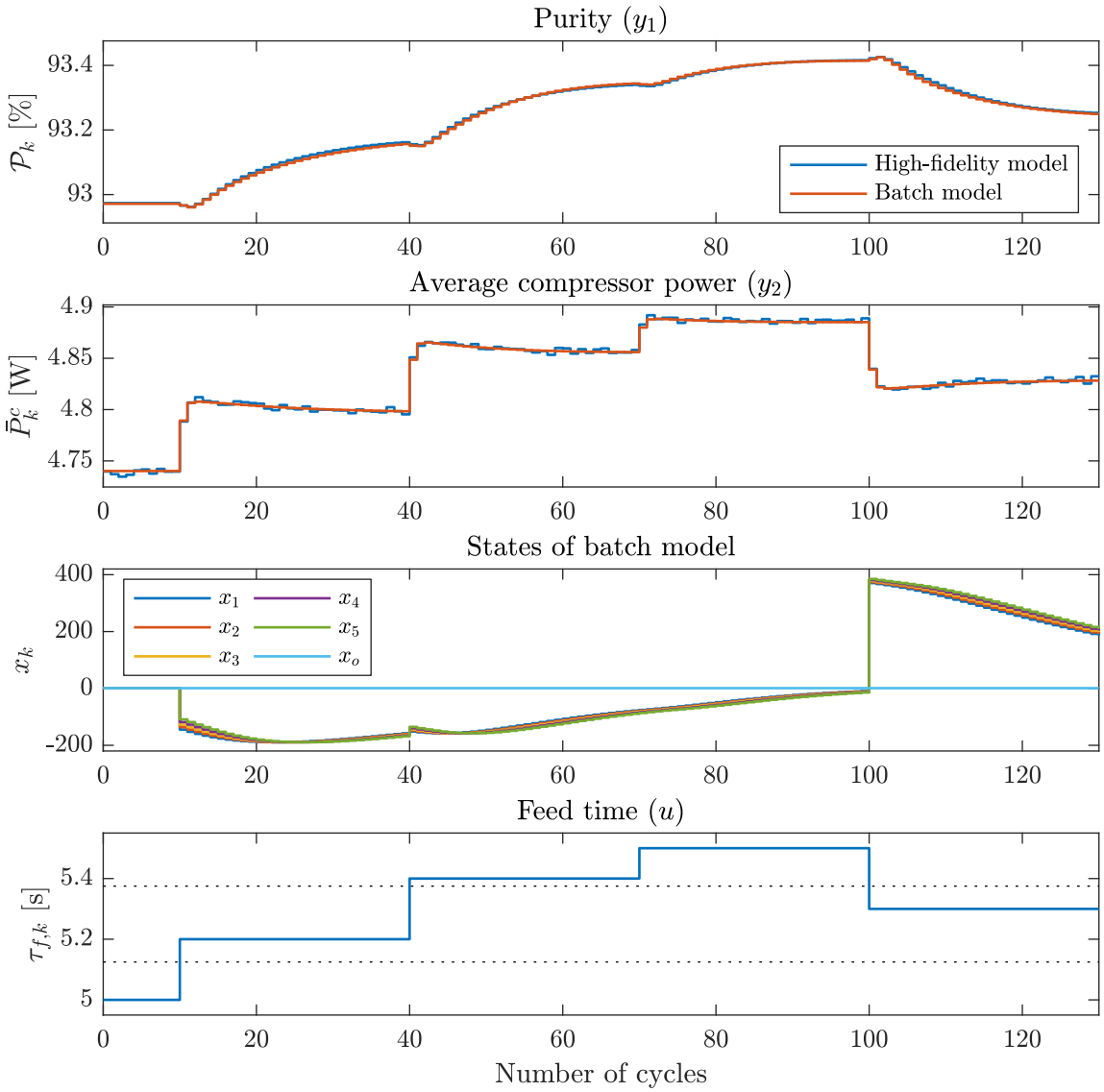

Figure 5-5: Comparison between the cycle-averaged outputs of the high-fidelity model and the batch model. Both models are initiated from CSS with $\tau_f = 5s$. The states of the batch model are also plotted, and the effect of the reset map can be seen when the input enters the range of another local model. The boundaries of the input spaces U_m of the local models are denoted with dotted lines.

Chapter 6

Model Predictive Control

This chapter will cover the design and derivation of the Model Predictive Control (MPC) strategy used to control the POC. MPC is a control method that exploits a (simplified) plant model to predict the future states and outputs, and calculates a optimal control input by minimizing a suitable cost function, whilst adhering to a set of constraints.

In this case, the objective is to minimize the compressor power usage, while keeping the purity above a minimum value. The hybrid batch model from Chapter 5 is used as a prediction model.

The lay-out of this chapter is as follows. Section 6-1 provides some background information, and discusses the motivation behind the choice for MPC. In the next section, the control approach will be explained. Section 6-3 concerns the derivation of the cost functions, and in Section 6-4 the tuning of the parameters is discussed. The chapter will end with a conclusion.

6-1 Background

MPC is based on iterative, finite horizon optimization of a plant model: at each time step k, a cost function is minimized over the prediction horizon H_p, yielding a sequence of optimal control inputs $\hat{u}_k$ for the next H_c steps, the control horizon. The following is based on [11, 59]. The cost function used in this thesis is:

$$\mathcal{J} = \tilde{y}_2^T Q_y \tilde{y}_2 + \tilde{u}^T Q_u \tilde{u}. \tag{6-1}$$

Here $\tilde{y}_2$ is a vector of predicted outputs $y_2 = \bar{P}_k^c$ over the prediction horizon, and $\tilde{u}$ a vector consisting of the differences of consecutive inputs, i.e. $\Delta\hat{u}_k = \hat{u}_k - u_{k-1}$:

$$\tilde{y}_2 = \begin{bmatrix} y_{2,k} \\ \vdots \\ y_{2,k+H_p-1} \end{bmatrix}, \qquad \tilde{u} = \begin{bmatrix} \Delta\hat{u}_k \\ \vdots \\ \Delta\hat{u}_{k+H_c-1} \end{bmatrix}. \tag{6-2}$$

The first term penalizes the second output $\bar{P}_k^c$, and the second term is used to penalize any excessive change in control input, which is needed to prevent the input from oscillating around a steady-state value. The matrices Q_y and Q_u are weighing matrices, used to change the relative importance of one of the terms, or e.g., one specific predicted output over the other outputs. The control horizon is smaller than the prediction horizon: $H_c < H_p$. After the control horizon, the control input is held constant: $\Delta\hat{u}_{k+i} = 0$ for $i \geq H_c$.

The optimization problem that is solved (each time step k) is the following:

$$\begin{aligned} \min_{\tilde{u}} \quad & \mathcal{J} = \tilde{y}_2^T Q_y \tilde{y}_2 + \tilde{u}^T Q_u \tilde{u}, \\ \text{s.t.:} \quad & \tilde{y}_1 \geq \mathcal{P}_{min}, \\ & \tilde{u} \in \tilde{\mathcal{U}}. \end{aligned} \tag{6-3}$$

Where $\tilde{y}_1$ is a vector of predicted purity outputs, also of length H_p. $\mathcal{P}_{min}$ is the minimum acceptable purity, and $\tilde{u} \in \tilde{\mathcal{U}}$ are constraints to ensure the input optimized input $\hat{u}_k$ stays inside the operating range $\mathcal{U}$, and that limit the maximum allowable input change Δu_{max}. Technically, the Input/Output (I/O) relationships described by the batch model are also constraints. However, these will be incorporated into the cost function (see Section 6-3).

Although H_c control inputs are optimized, only the first control input is used each time step. This way the most recent measurements are utilized immediately.

Like said before, to compute these predicted outputs, the hybrid batch model from Chapter 5 is used. At each time step, this batch model is initiated with the estimated states and the known input of the previous cycle u_{k-1}. To estimate the batch states, an observer is needed, which will be discussed in Section 6-2-4.

6-1-1 Motivation

In Chapter 1, the reason for developing an MPC-based controller was already given. It will be repeated here.

The main goal of this study is to improve the efficiency of the POC, by reducing the power consumption. Because of sieve bed degradation, as discussed in Section 2-2-4, the dynamics of the POC will change over time. The current control method (see Section 2-1-1) uses lookup-tables to adjust the valve timings in open-loop, and has no direct knowledge of the system dynamics. Therefore the tolerances might not be as tight as is desired.

Because MPC uses a model of the plant, it does have information about the system dynamics, and therefore is able to predict and act on changes of the system. MPC lends itself well to this particular case, since the objective is to minimize the power consumption. Constraints like a minimum purity value can be added, and the controller could be extended to a Multi-Input Multi-Output (MIMO) case (e.g., the addition of the feed flow rate as an input).

Another advantage of MPC is adaptation. Currently, tuning a POC is done at the factory. So if a device needs a sieve bed replacement, it has to be sent back to the factory. Using MPC, it might be possible to use a controller which automatically adapts itself, based on the location of the user (e.g., high or low altitude) and the capacity of the sieve beds. This way the POC does not need to be returned.

The aim of this thesis is to show the feasibility of MPC. A self-tuning controller or other improvements will be part of future work.

6-1-2 MPC in the PSA Literature

Initially developed in the 1970s, MPC has been a popular subject for academic and industrial research. The popularity of this control strategy is due to its ability to predict future plant outputs, providing early warnings of potential problems, and the fact that the controller can handle constraints. This makes MPC an excellent choice for the process industry, where input and output constraints play an important role [60]. The same holds for PSA processes: different studies have been published that use MPC. For example, both studies mentioned in Chapter 5 that use a single linear batch model, the first by Khajuria [13] and the other by Urich et al. [42], use MPC to control their respective PSA plant.

Unfortunately, both studies have PSA plants that differ much from the POC in this thesis, and the objectives of these studies (disturbance rejection) also also different. Therefore comparing these studies with the approach in this chapter is not helpful.

6-2 Switched Linear MPC

The approach that has been taken in this thesis can described as switched linear MPC. Instead of using the complete hybrid batch model to compute the predicted states, only the local linear model q_m corresponding to the last known input, $u_{k-1} \in U_m$, is used. If the controller steers the system to a different local model (i.e. the first optimized input $\hat{u}_k$ lies in a different range U_n), in the next cycle the cost function will be updated with the new state-space matrices, states and inputs, and this new cost function is optimized.

This approach, switched MPC, can also be found in the literature. Two kinds are found: MPC where switching is not incorporated in the optimization (as is the case in this study), e.g., [61], and MPC where switching is incorporated in the optimization, e.g., [62]. However, switched MPC where the input is the only switching variable has not been found.

First, the reasoning behind this approach is explained. After that, in Section 6-2-2, switching between local models is explained, and the stability of this approach is analyzed. In Section 6-2-3 Quadratic Programming (QP) is explained, and lastly in Section 6-2-4, the design of the observer, needed to estimate the states of the batch model, is discussed.

6-2-1 Convexity

To understand why this approach has been taken, the notion of convexity needs to be explained. A function $f(x)$ is convex on an interval $[a, b]$ if for any two points x_1, x_2 in $[a, b]$ and any $\lambda \in (0, 1)$ [63],

$$f\left(\lambda x_1 + (1-\lambda)x_2\right) \leq \lambda f(x_1) + (1-\lambda) f(x_2). \tag{6-4}$$

As a consequence, a convex function only has a global minimum, and no local ones. Likewise, a convex cost function only has one minimum, and this poses an advantage to a solver: there is no chance of getting stuck in local minima, so the calculated input will always be the optimal one. Global solvers can handle non-convex optimization problems, but are more complex and often it cannot be proven that they will find the global minimum.

The hybrid batch model as described in the previous chapter is, due to the reset maps R, a nonlinear model. If this nonlinear model is used to compute $\tilde{y}_1$ and $\tilde{y}_2$ in Eq. (6-3), there is no guarantee that the resulting optimization problem will be convex. In fact, simulations with the hybrid batch model as a predictive model for an MPC have shown the controller getting stuck in local minima. This behavior is unacceptable for the POC: e.g., not reaching the required purity levels can be harmful to the user.

6-2-2 Switching

Like has been said, switching between local models is based on the inputs. At the end of step k, after optimization, the range U_n of the newfound input u_k is compared with the range of the previous input $u_{k-1} \in U_m$:

$$\text{switch if} \quad U_m \neq U_n. \tag{6-5}$$

If so, the local model q_n corresponding to the range U_n is used during optimization in the next cycle. To prevent oscillations, i.e. unnecessary switching, a dwell time has been added (see Section 5-3-1). During this dwell time, the controller can not switch back to the local model(s) used in the previous five cycles or steps k.

This is not a hard proof that this switched system is stable. This has not been done for the controller: proving that a "hybrid" MPC-based controller is stable, is difficult. Instead, the input weight Q_u and input constraints are chosen to limit any oscillation (see Section 6-4), and linear MPC is used to ensure the inputs are optimal over the prediction horizon.

A consequence of this approach is that, during optimization, some of the optimized inputs will lie outside the current operating range U_m, as illustrated in Figure 6-1. Since it is not possible to switch local models during optimization, the wrong state-space matrices will have been used for these inputs.

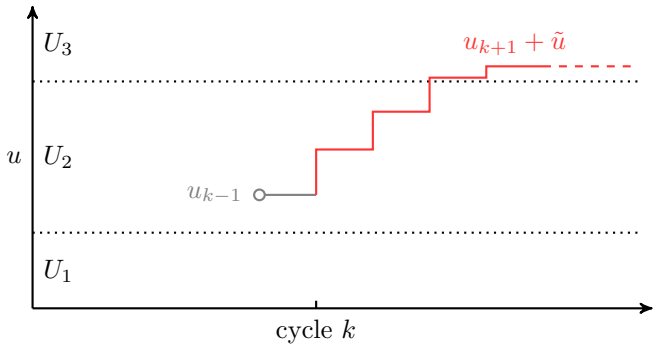

Figure 6-1: Example of optimization during cycle k. Starting from the previous input u_{k-1}, inputs $\tilde{u}$ are optimized over the control horizon (here $H_c = 4$). Since $u_{k-1} \in U_2$, local model q_2 has been used for optimization. Note that the last two optimized inputs are in U_3.

To show that for neighboring models (i.e. with neighboring ranges U_m), using the wrong model is not detrimental, additional validation on the local models has been done. Every local model $q_m, m = [1, \ldots, 13]$ has been validated with the validation data sets from its (up to four) neighboring models. The results can be found in Table 6-1, and the same measure

of fit as in Chapter 5 has been used: the NRMSE defined in Eq. 5-10. For most of the local models, the fits of the closest neighboring validation sets are still above 75%; only for higher input ranges the fit on the purity starts getting worse, and there are some outliers with the second input, average compressor power.

Model	Fit of y_1 [%] on validation set of model					Fit of y_2 [%] on validation set of model				
(q_m)	q_{m-2}	q_{m-1}	q_m	q_{m+1}	q_{m+2}	q_{m-2}	q_{m-1}	q_m	q_{m+1}	q_{m+2}
1	-	-	97.04	84.52	71.58	-	-	92.74	88.28	66.22
2	-	81.12	96.85	84.12	71.15	-	87.05	93.11	85.03	54.83
3	59.19	81.26	97.21	84.77	71.28	84.03	85.03	91.93	79.39	78.6
4	59.25	80.7	97.23	84.46	71.3	74.99	47.96	90.9	89.04	83.06
5	60.54	81.34	97.14	84.31	70.58	-9.48	89.35	90.8	89.33	86.26
6	59.18	81.18	97.13	83.69	69.19	82.6	89.66	90.85	89.17	59.58
7	60.11	81.03	97.66	82.6	67.54	86.25	88.83	91.37	84.45	83.66
8	57.16	79.55	96.55	81.85	64.71	87.29	82.75	90.85	90.13	88.78
9	55.01	79.1	96.93	78.24	58.97	61.63	90.94	91.73	91.08	51.63
10	49.18	75.44	96.07	74.54	50.93	87.7	90.29	91.05	69.59	57.69
11	36.28	68.38	93.73	69.69	39.72	84.02	85.69	89.87	87.17	85.83
12	18.84	59.0	94.14	60.03	-	71.21	87.73	90.36	89.7	-
13	-2.96	48.22	92.27	-	-	85.89	89.53	91.49	-	-

Table 6-1: Fits of every local model m on the validation sets of its four neighboring models (and its own), to show that the models are also able to predict dynamics which lie slightly out of their operating range. On the left, y_1 (purity) is shown, and on the right, y_2 (average compressor power) is shown. Model 1 is the model with range $U_1 \in [3.375, 3.625]$, model 13 the model with range $U_{13} \in [6.375.6.625]$, et cetera.

In other words, the local models are still able to predict the dynamics corresponding to inputs which lie slightly out of their operating range. Measures are taken to ensure the predicted input stay close to "their" input range U_m, namely by choosing a small control horizon H_c and setting a constraint on the maximal allowable change in input Δu_{max}. This approach will be validated in the next chapter, and the tuning of the two mentioned parameters is discussed in Section 6-4.

6-2-3 Quadratic Programming

Since the prediction model is now a single local linear model, the optimization problem can be recast as a Quadratic Programming problem [59]. Such an optimization problem has the following form:

$$\begin{aligned} \min_{\tilde{u}} \quad & \mathcal{J} = \frac{1}{2}\tilde{u}^T H \tilde{u} + f^T \tilde{u}, \\ \text{s.t.:} \quad & A_{ineq}\tilde{u} \leq b_{ineq}, \end{aligned} \tag{6-6}$$

where A_{ineq} and b_{ineq} are a matrix and vector making up the (in)equality constraints. If H, called the Hessian, is a positive definite matrix and the (in)equality constraints are linear, the optimization problem is convex. Such a problem can be easily solved by any suitable solver,

and in this study the MATLAB function `quadprog` has been used, which is designed to solve these QP problems.

The optimization algorithm that has been used is called 'interior-point-convex', the default algorithm for `quadprog`. Interior-point algorithms can solve linear and nonlinear convex optimization problems, and are based on so-called barrier functions [64].

The matrices and vectors H, f, A_{ineq} and b_{ineq} are built up from linear combinations of the state-space matrices and states of the current local model, and the previous input u_{k-1}. Their derivation will be explained in Section 6-3.

6-2-4 Observer Design

Like said in the previous section, an observer is needed to estimate the states of the batch model. The switching aspect of the controller is incorporated in the observer: since the states pass a reset map when a switch happens, the observer should also have knowledge of this reset map R.

The linear part of the observer is a Luenberger observer [65]. Other types of observers have not been evaluated, since this was not the focus of this thesis, and this observer showed accurate estimations (i.e. low observer error).

Let the estimated states be denoted by $\hat{x}_k$, and equivalently the estimated output with $\hat{y}_k$. Given the state-space matrices of the current local model A_m, B_m, C_m, D_m, the observer dynamics are:

$$\hat{x}_{k+1} = A_m\hat{x}_k + B_m u_k + L_m\left(y_k - \hat{y}_k\right), \tag{6-7a}$$

$$\hat{y}_k = C_m\hat{x}_k + D_m u_k, \tag{6-7b}$$

where L_m is called the observer gain. An observer is called asymptotically stable if the observer error $e_k = \hat{x}_k - x_k$ converges to zero for $k \to \infty$. The error dynamics are:

$$\begin{aligned} e_{k+1} &= \hat{x}_{k+1} - x_{k+1}, \\ &= A_m\hat{x}_k + B_m u_k + L_m\left(y_k - \hat{y}_k\right) - A_m x_k + B_m u_k, \\ &= (A_m - L_m C_m)e_k. \end{aligned} \tag{6-8}$$

Therefore, by choosing the observer gain L_m such that the eigenvalues of the matrix $A_m - L_m C_m$ are within the unit disk, the observer is made asymptotically stable. A requirement for this is that all pairs A_m, C_m are observable. Like said in Section 5-3-2, this is the case, so all local observers are asymptotically stable.

A high observer gain L_m ensures that the observer states converge quickly to the real states. The downside of a high observer gain is that the initial error can get dangerously large because of this high L_m, and any noise is also amplified. Tuning of $L_m, m = [1, \ldots, 13]$ will be discussed in Section 6-4.

If the optimization algorithm switches between subsystems, the state-space matrices of the observer will also switch to match the current local model. The observer states $\hat{x}_k$ are subsequently reset with the reset map $R(q_m, q_n)$ from Section 5-3-2. Since all the error dynamics are asymptotically stable, the observer will remain asymptotically stable if a state reset happens.

During simulations, it became clear that using only the purity output, y_1, yielded better results than using both outputs for the observer. This is due to the noise present in the signal of the average compressor power, as explained in Section 4-3, whereas the purity is a smooth signal. A Kalman filter specializes in filtering noise, which might have solved this problem, but only using y_1 was simpler and also effective.

6-3 Derivation of Optimization Problem

In this section the derivation of the cost function and the constraints will be explained. That is, how to change Eq. (6-1) into the form of a QP problem as shown in Eq. (6-6). The following is based on [59].

First, recall that $\tilde{y}_i, i \in \{1, 2\}$ is a vector of length H_p of predicted outputs of either the purity (y_1) or the average compressor power (y_2), and that $\tilde{u}$ is a vector of the (to be optimized) differences between consecutive inputs of length H_c. Let $\tilde{x}$ be a vector of corresponding predicted states, its length being $6H_p$. In this section the subscript m, denoting the local model, will be dropped, since switching does not take place during optimization. The following holds for all q_m.

Since the differences in inputs $\Delta\hat{u}_k$ are optimized, it is convenient to also write the state-space difference equations in terms of $\Delta\hat{u}_k$:

$$x_{k+1} = Ax_k + B\Delta\hat{u}_k + Bu_{k-1}, \tag{6-9a}$$
$$y_k = Cx_k + D\Delta\hat{u}_k + Du_{k-1}. \tag{6-9b}$$

Note that u_{k-1} is known and $\hat{u}_k$ is the to-be-optimized input. The goal is to write any output $y_{k+i}, i \in [0, H_p - 1]$ in terms that are known (state-space matrices, current states x_k, previous input u_{k-1}) and in terms of $\tilde{u}$.

The resulting difference equation is shown here:

$$x_{k+i} = A^i x_k + \sum_{j=0}^{i-1} A^j B u_{k-1} + \begin{bmatrix} \sum_{j=0}^{i-1} A^j B & \sum_{j=0}^{i-2} A^j B & \dots & B \end{bmatrix} \begin{bmatrix} \Delta\hat{u}_k \\ \Delta\hat{u}_{k+1} \\ \vdots \\ \Delta\hat{u}_{k+i} \end{bmatrix}, \tag{6-10a}$$

$$y_{k+i} = Cx_{k+i} + Du_{k-1} + \begin{bmatrix} D & D & \dots & D \end{bmatrix} \begin{bmatrix} \Delta\hat{u}_k \\ \Delta\hat{u}_{k+1} \\ \vdots \\ \Delta\hat{u}_{k+i} \end{bmatrix}. \tag{6-10b}$$

An added advantage of this notation is that for $i \geq H_c, \Delta u_{k+i} = 0$, because after the control horizon the input is held constant. Using this notation, the predicted state and output vectors

$\tilde{x}$ and $\tilde{y}$ at time step k can be written as:

$$\underbrace{\begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+H_c} \\ x_{k+H_c+1} \\ \vdots \\ x_{k+H_p-1} \end{bmatrix}}_{\tilde{x}} = \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^{H_c} \\ A^{H_c+1} \\ \vdots \\ A^{H_p-1} \end{bmatrix}}_{\tilde{A}} x_k + \underbrace{\begin{bmatrix} 0 \\ B \\ \vdots \\ \sum_{i=0}^{H_c-1} A^i B \\ \sum_{i=0}^{H_c} A^i B \\ \vdots \\ \sum_{i=0}^{H_p-2} A^i B \end{bmatrix}}_{\tilde{B}_0} u_{k-1} +$$

$$\underbrace{\begin{bmatrix} 0 & \dots & 0 \\ B & \dots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_c-1} A^i B & \dots & B \\ \sum_{i=0}^{H_c} A^i B & \dots & AB+B \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-2} A^i B & \dots & \sum_{i=0}^{H_p-H_c-1} A^i B \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} \Delta\hat{u}_k \\ \vdots \\ \Delta\hat{u}_{k+H_c-1} \end{bmatrix}}_{\tilde{u}} \tag{6-11a}$$

$$\underbrace{\begin{bmatrix} y_k \\ \vdots \\ y_{k+H_c-1} \\ \vdots \\ y_{k+H_p-1} \end{bmatrix}}_{\tilde{y}} = \underbrace{\begin{bmatrix} C & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C \end{bmatrix}}_{\tilde{C}} \underbrace{\begin{bmatrix} x_k \\ \vdots \\ x_{k+H_p-1} \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} D \\ \vdots \\ D \end{bmatrix}}_{\tilde{D}_0} u_{k-1} + \underbrace{\begin{bmatrix} D & \dots & 0 \\ \vdots & \ddots & \vdots \\ D & \dots & D \\ \vdots & \vdots & \vdots \\ D & \dots & D \end{bmatrix}}_{\tilde{D}} \underbrace{\begin{bmatrix} \Delta\hat{u}_k \\ \vdots \\ \Delta\hat{u}_{k+H_c-1} \end{bmatrix}}_{\tilde{u}} \tag{6-11b}$$

If the first output is needed, $\tilde{y}_1$, the extended matrices $\tilde{C}, \tilde{D}_0$ and $\tilde{D}$ are made up of the respective row (e.g., the first for $\tilde{y}_1$) of the matrices C and D. The extended matrices are then denoted as $\tilde{C}_1, \tilde{D}_{0,1}$ and $\tilde{D}_1$. The same holds for $\tilde{y}_2$.

Using these extended matrices, the predicted output $\tilde{y}_1$ can be written as:

$$\begin{aligned} \tilde{y}_1 &= \tilde{C}_1\tilde{x} + \tilde{D}_{0,1}u_{k-1} + \tilde{D}_1\tilde{u}, \\ &= \tilde{C}_1\left(\tilde{A}x_k + \tilde{B}_0 u_{k-1} + \tilde{B}\tilde{u}\right) + \tilde{D}_{0,1}u_{k-1} + \tilde{D}_1\tilde{u}, \\ &= \underbrace{\tilde{C}_1\tilde{A}x_k + \left(\tilde{C}_1\tilde{B}_0 + \tilde{D}_{0,1}\right)u_{k-1}}_{\tilde{F}_1} + \underbrace{\left(\tilde{C}_1\tilde{B} + \tilde{D}_1\right)}_{\tilde{G}_1}\tilde{u}. \end{aligned} \tag{6-12}$$

For clarity, the extended matrices have been grouped together into a vector of known elements $\tilde{F}_1$, and a matrix which depends on $\tilde{u}$, denoted by $\tilde{G}_1$. The prediction vector of the second output is also written like this:

$$\tilde{y}_2 = \tilde{F}_2 + \tilde{G}_2\tilde{u}. \tag{6-13}$$

Cost Function

Using the newfound vectors and matrices $\tilde{F}_i, \tilde{G}_i, i \in \{1, 2\}$, the original cost function from Eq. (6-1) can be rewritten:

$$\begin{aligned} \mathcal{J} &= (\tilde{F}_2 + \tilde{G}_2\tilde{u})^T Q_y (\tilde{F}_2 + \tilde{G}\tilde{u}) + \tilde{u}^T Q_u \tilde{u}, \\ &= \tilde{F}_2^T Q_y \tilde{F}_2 + 2\tilde{F}_2^T Q_y \tilde{G}_1 \tilde{u} + \tilde{u}^T \left(\tilde{G}_1^T Q_y \tilde{G}_1 + Q_u\right) \tilde{u}. \end{aligned} \tag{6-14}$$

The first term in this equation does not depend on $\tilde{u}$, and thus it will not be affected by the optimization. The actual value of the cost function is not needed, only for which $\tilde{u}$ it is minimal. Therefore it can be dropped from the cost function, so:

$$\mathcal{J} = \underbrace{2\tilde{F}_2^T Q_y \tilde{G}_1}_{f^T} \tilde{u} + \tilde{u}^T \underbrace{\left(\tilde{G}_1^T Q_y \tilde{G}_1 + Q_u\right)}_{2H} \tilde{u}. \tag{6-15}$$

The cost function is now in the form of the QP problem of Section 6-2-3. Q_y and Q_u are positive definite by design, and the product of the matrices $\tilde{G}_1^T Q_y \tilde{G}_1$ is positive-definite. Therefore the matrix $2H$ is positive-definite, and the optimization problem is convex.

Constraints

Now that the cost function has been rewritten into the QP form, only the constraints remain. These all have to be rewritten into the form $A_{ineq}\tilde{u} \leq b_{ineq}$ (since there are no equality constraints). Starting with the constraint on the purity, which is rewritten, using Eq. (6-13):

$$\tilde{y}_1 \geq \mathcal{P}_{min}, \tag{6-16a}$$

$$-\tilde{G}_1\tilde{u} \leq \tilde{F}_1 - \mathcal{P}_{min}. \tag{6-16b}$$

Eq. (6-16b) is a so-called hard constraint: the purity cannot fall under the limit. This is a potential problem. If this limit is raised during simulation because a higher purity level is desired, the controller may not able to provide an input for which this constraint holds. With that in mind, slack variables $\tilde{s}$ have been added to this optimization problem such that this purity constraint becomes a soft constraint. Eq. (6-16b) becomes:

$$-\tilde{G}_1\tilde{u} - \tilde{s} \leq \tilde{F}_1 - \mathcal{P}_{min}, \tag{6-17}$$

$$-\tilde{s} \leq 0, \tag{6-18}$$

and these slack variables will also be penalized:

$$\min_{\tilde{u},\tilde{s}} \mathcal{J} = \frac{1}{2}\left(\tilde{u}^T H \tilde{u} + \tilde{s}^T Q_{\tilde{s}} \tilde{s}\right) + f^T \tilde{u}. \tag{6-19}$$

If there is a $\tilde{u}$ that satisfies Eq. (6-16b), $\tilde{s} = 0$. If not, then $\tilde{s}$ will be the smallest value that satisfies Eq. (6-17). The extra constraint in Eq. (6-18) ensures the slack variables are positive. $Q_{\tilde{s}}$ is a positive definite weighing matrix for these slack variables, and tunes the "softness" of this constraint. This will also be discussed in Section 6-4.

The constraints on the input are as follows. Firstly, the maximum change is constrained by the vector Δu_{max}, which is tuneable, and the second constraint ensures the optimized inputs $\tilde{u}$ stay inside the operating range $\mathcal{U}$.

$$|\tilde{u}| \leq \Delta u_{max}, \tag{6-20}$$

$$\begin{bmatrix} 3.5 \\ \vdots \\ 3.5 \end{bmatrix} \leq u_{k-1} + \begin{bmatrix} \Delta \hat{u}_k \\ \vdots \\ \sum_{i=k}^{k+H_c-1} \Delta \hat{u}_i \end{bmatrix} \leq \begin{bmatrix} 6.5 \\ \vdots \\ 6.5 \end{bmatrix}. \tag{6-21}$$

Rewriting the above constraints into the form $A_{ineq}\tilde{u} \leq b_{ineq}$ is considered trivial and will not be shown. Instead, these two constraints (Eqs. (6-20) and (6-21)) will be denoted by $\tilde{u} \in \tilde{\mathcal{U}}$.

Final Form

The final form of the first optimization problem, minimizing the power consumption, is obtained by first defining the following matrices:

$$H_{aug} = \begin{bmatrix} \tilde{G}_2^T Q_y \tilde{G}_2 + Q_u & 0 \\ 0 & Q_{\tilde{s}} \end{bmatrix}, f_{aug} = \begin{bmatrix} 2\tilde{F}_2^T Q_y \tilde{G}_2 \\ 0 \end{bmatrix}, \tilde{u}_{aug} = \begin{bmatrix} \tilde{u} \\ \tilde{s} \end{bmatrix}, \tag{6-22}$$

and

$$A_{aug} = \begin{bmatrix} -\tilde{G}_1 & -I \\ 0 & -I \end{bmatrix}, b_{aug} = \begin{bmatrix} \tilde{F}_1 - \mathcal{P}_{min} \\ 0 \end{bmatrix}. \tag{6-23}$$

Yielding the final form:

$$\begin{aligned} \min_{\tilde{u}_{aug}} \quad & \mathcal{J} = \frac{1}{2}\tilde{u}_{aug}^T H_{aug} \tilde{u}_{aug} + f_{aug}^T \tilde{u}_{aug}, \\ \text{s.t.:} \quad & A_{aug}\tilde{u}_{aug} \leq b_{aug}, \\ & \tilde{u} \in \tilde{\mathcal{U}}. \end{aligned} \tag{6-24}$$

6-4 Tuning

In the previous sections the approach, and the derivation of the cost functions and constraints have been explained. This section will discuss the tuning of the MPC parameters.

Firstly, the observer gains L_m. Like said in Section 6-2-4, the poles of the matrices $A_m - L_m C_m, m = [1, \ldots, 13]$ should be placed inside the unit circle. If they are chosen too fast (i.e. too close to zero), oscillations in the error may appear. For that reason the poles p of every matrix $A_m - L_m C_m$ have been placed at:

$$p = \begin{bmatrix} -0.4 & 0.4 & -0.45 & 0.45 & 0.5 \end{bmatrix}. \tag{6-25}$$

Different pole locations have been evaluated, and these values provided the best results. The values of the observer gains L_m can be found in Appendix A-2.

MPC Parameters

The parameters for the optimization have been chosen as follows. They are the same for every local model, since tuning the parameters for every model individually is tedious, and good results were obtained this way.

First, the prediction horizon H_p will be discussed. It is good practice to pick the prediction horizon as long as possible [59], i.e. increase H_p until further increases have no effect on the performance. For that reason, the prediction horizon has been set as $H_p = 20$.

The control horizon H_c and the maximum change in input Δu_{max} are used to constrain the input, as discussed in Section 6-2-2. Table 6-1 showed that fits were still acceptable when the input was close to the original range of the used local model. For that reason, a control horizon of $H_c = 2$ and a maximum input change of $\Delta u_{max} = 0.1$ has been chosen. With these two parameters it can be guaranteed that the optimized inputs $[\hat{u}_k, \hat{u}_{k+1}]$ will lie inside the range of the current model q_m and the two closest neighboring models q_{m-1}, q_{m+1}, i.e.:

$$u_{k-1} + \begin{bmatrix} \Delta\hat{u}_k \\ \Delta\hat{u}_k + \Delta\hat{u}_{k+1} \end{bmatrix} \in U_m + U_{m-1} + U_{m+1}. \tag{6-26}$$

The downside is that the controller will be slow because of this small Δu_{max}.

All that remains are the weighing matrices Q_y, Q_u and $Q_{\tilde{s}}$, for the output, input and slack variables, respectively. $Q_{\tilde{s}}$ should be high, to penalize any constrain violations. In this case, $Q_{\tilde{s}} = 500 I_{H_p}$ was chosen (where I_w is a $(w \times w)$-identity matrix). The choice of the other weighing matrices, Q_y and Q_u, is more important, since they help shape the dynamics of the controller. If $Q_u = 0$, excessive input changes are not penalized and the input will oscillate unnecessary. Obviously, if $Q_y \to 0$, the cost function has a minimum for $\tilde{u} = 0$, so $u_k = u_{k-1}$. Therefore a trade-off must be made.

That trade-off is the following: pick Q_u and Q_y such that unnecessary input oscillations are kept to a minimum, and that the transient response is as fast as possible. The following weighting matrices met this criteria: $Q_u = 10 I_{H_c}$ and $Q_y = (1/20) I_{H_p}$.

In Chapter 7 the simulation results of the controller, using these parameters, will be discussed.

6-5 Concluding Remarks

In this chapter the design of the controller has been discussed. MPC has been chosen as the control method, since its optimization-based approach is very suited for the goal of this thesis. Local linear MPC is implemented, such that the optimization problem can be cast as a convex QP problem.

Two cost functions have been derived: one that minimizes the power consumption of the compressor with a constraint on the purity, and the other that minimizes the tracking error between a desired purity value and the true purity.

The batch states are estimated with an Luenberger observer which makes use of the hybrid batch model from Chapter 5.

Chapter 7

Results

This chapter will show the simulation results obtained with the MPC-based controller that was derived in Chapter 6. The high-fidelity model from Chapter 3 is used as a plant model for the POC, and the predictive model for the controller is the hybrid batch model, derived in Chapter 5.

Three topics will be discussed in this chapter. First, the used observer will be analyzed, in Section 7-1. Subsequently, in Section 7-2, the performance of the controller under normal operation (i.e. no sieve bed degradation in the POC) will be reviewed. After that, a case study on sieve bed degradation will be discussed: the POC will be simulated with degraded sieve beds, such that the performance of the derived controller can be compared with the case without degradation. The chapter will conclude with a summary.

7-1 Observer Performance

In this section, two decisions regarding the design of the observer will be discussed. The first is why only the purity $\mathcal{P}$ is used to estimate the states $\hat{x}$ (instead of both outputs), and the second is about the impact of the reset maps R on the observer performance.

7-1-1 Choice of Outputs for Observer

To show why only the purity is used for the observer, two simulations have been done in which the high-fidelity model is controlled with the MPC-strategy derived in Chapter 6. In the first simulation, only $\mathcal{P}$ is used by the observer to estimate the states, and in the second, both outputs are used by the observer. For both simulations, $\mathcal{P}_{min}$ undergoes a single step change. The results can be found in Figure 7-1. Because the "real" batch states are not known, a comparison between observer states and real states cannot be shown. Instead the observer error, $e_k = \hat{x}_k - x_k$, will be approximated with the differences between estimated outputs and true outputs. Let $e_{1,k}$ be the error between $\hat{y}_1$ and y_1 (purity), and likewise $e_{2,k} = \hat{y}_2 - y_2$

(for the average compressor power). Also, the observer using only $\mathcal{P}$ will be called observer 1, and the observer that uses $\mathcal{P}$ and $\bar{P}^c$ will be called observer 2.

The advantage of observer 1 is visible in the plot showing $e_{1,k}$. For observer 1, this error goes to zero quickly, whereas for observer 2, this error keeps oscillating. Note also that there is almost no difference in magnitudes for $e_{2,k}$. The result is a much smoother input trajectory (with less switching) for the case with observer 1 compared to the case with observer 2.

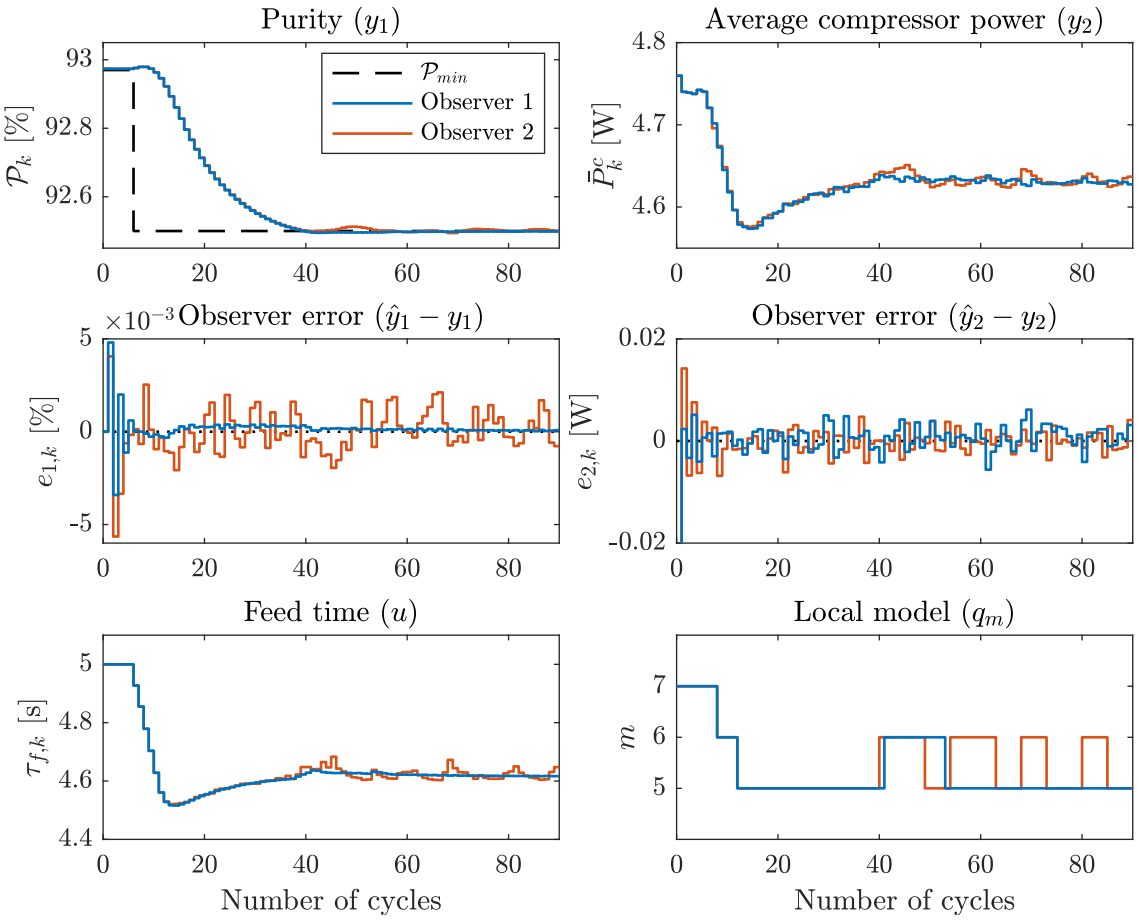

Figure 7-1: Comparison between the observer that only uses the purity to estimate the states (observer 1), and the observer that uses both outputs to estimate the states (observer 2). Since the actual batch states are not known, the observer error is approximated with the outputs of the observer and the plant.

These results are somewhat surprising, since this means that the observer receiving less information performs better. This is due to the noise present in the second output ($\bar{P}^c$), as explained in Section 4-3. If the observer tries to steer $e_{2,k}$ to zero, it does not take the noise into account, resulting in unnecessary oscillations in the states and the outputs, also visible in Figure 7-2, where the estimated states of both observers are shown for these simulations.

In hindsight, a Kalman filter could have been useful, since it is designed to filter out noise present in a signal, such that both outputs could be used by the observer. However, since the performance of observer 1 (so using only one output) was deemed sufficient, further research into state estimators has not been done.

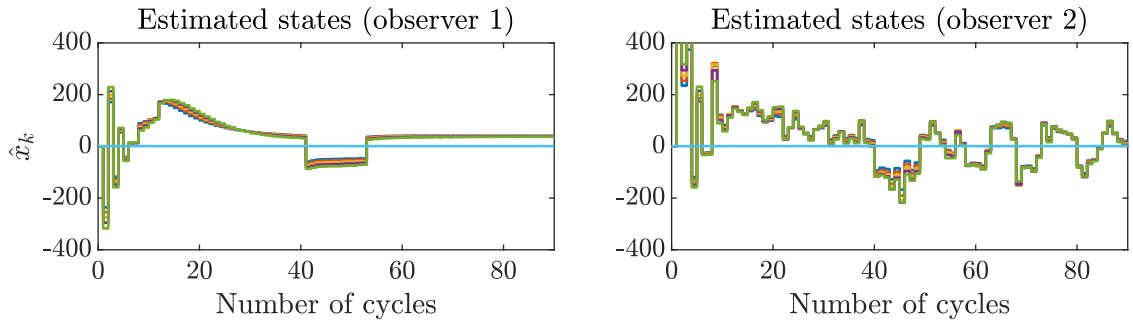

Figure 7-2: Estimated states of both observers.

7-1-2 Effect of Reset Maps

In section 5-3-2, the reset maps R were derived. These reset maps ensure that the states are reset when a switch happens, and in this section the necessity of these reset maps will be shown. In Figure 7-3 the results of a simulation can be seen, where an observer has been used that does not have the reset maps incorporated. The same step change has been put on $\mathcal{P}_{min}$ as in the previous section, such that this observer can be compared with the observers discussed there.

It is clear that not using the reset maps results in instability. Because of the switching between local models, an observer error appears, which steers the input to another local model, such that a new observer error appears, and the process repeats itself. Were it not for the dwell time of five cycles, the oscillations would even be more severe. In other words, the reset maps are necessary in providing stability for the observer and thus the controller.

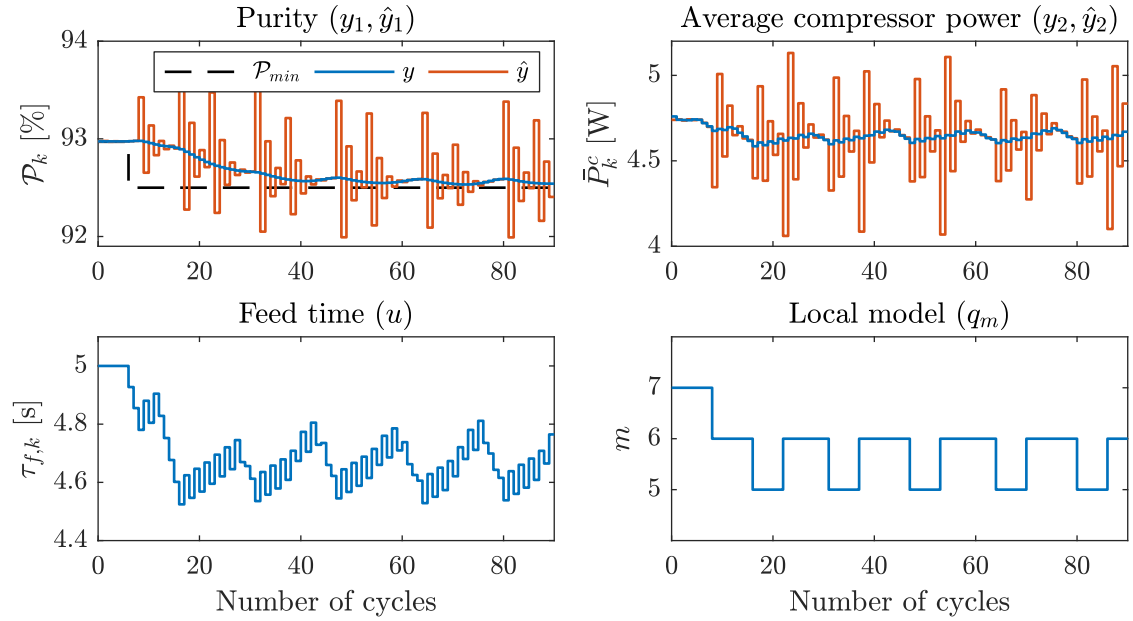

Figure 7-3: Simulation of the observer (using only the purity) without reset maps. Observer errors are not shown like in Figure 7-1, because the differences between y and $\hat{y}$ can be seen from the top two plots.

7-2 Controller Performance

Now the performance of the controller under normal operation will be discussed (so without sieve bed degradation). Recall that the objective of the controller is to minimize the average compressor power $\bar{P}^c$, subject to a (soft) constraint on the purity $\mathcal{P}$, i.e. $\mathcal{P} \geq \mathcal{P}_{min}$. In practice, this means that the controller should steer the purity to this minimal value. To show that the controller can handle different settings (i.e. minimal purity levels), and can switch between them, $\mathcal{P}_{min}$ will be subjected to step changes, which the controller should track.

The reason the controller should steer the purity to $\mathcal{P}_{min}$ is because of the I/O relationships. Recall Figure 4-5 in Chapter 4, where the I/O relationships of both outputs were shown. The CSS relationship between $\bar{P}^c$ and τ_f is linear, so to minimize the power consumption one has to minimize the feed time, only constrained by $\mathcal{P}_{min}$. Since the other I/O relationship (of the purity) is also monotonically increasing for the current operating range (defined in Section 5-1-3), steering the purity to $\mathcal{P}_{min}$ will result in the lowest possible power consumption.

Tracking of $\mathcal{P}_{min}$, i.e. the minimization of a certain tracking error, is not the explicit objective of the controller such that this MPC framework is suitable for future improvements. The analogy between the two objectives (power minimization and tracking) only holds because of the current simplifications and assumptions; if e.g., the input range $\mathcal{U}$ is extended, the I/O relationship between $\mathcal{P}$ and τ_f is no longer monotonically increasing, and thus tracking $\mathcal{P}_{min}$ does not guarantee power minimization.

In Figure 7-4, the results of a simulation are shown. The POC has been simulated for approx. 400 cycles, and $\mathcal{P}_{min}$ is subjected to step changes, as previously explained in this section. The top two plots show the two outputs of the POC: the purity (y_1) and the average compressor power (y_2). In the top plot, $\mathcal{P}_{min}$ has also been plotted in addition to the purity. In the third plot, the "tracking error" $e_{\mathcal{P}}$ is shown, i.e. how close the purity is to $\mathcal{P}_{min}$. The fourth plot shows the input, and the bottom plot shows the switching, i.e. the local model for the respective cycle.

Looking at the top plot, it can be seen that the controller is successful in steering the system to the lowest possible purity value, since the tracking error goes to zero after every step change. The speed of the response is limited by the constraints on Δu_k (see Section 6-4), which ensures the prediction model uses only viable local models for the optimization. Due to the weights on the input, Q_u, input oscillations are kept to a minimum. This is also visible in the bottom plot: there is no excessive switching between local models.

In the third plot, around cycles $k = 170$ and $k = 270$, overshoots can be seen. This is a consequence of the soft constraint on $\mathcal{P}$: here $\mathcal{P}_{min}$ was increased, and therefore the constraint is suddenly violated, resulting in nonzero slack variables. Since these are heavily penalized in the cost function, the controller will then aggressively steer the purity above this new minimum value to get the slack variables to zero again. The result is a peak in the input and thus a slight (positive) overshoot. This does not happen when $\mathcal{P}_{min}$ decreases, hence the response of the system is much more gradual when this happens (e.g., around $k = 50$).

The fact that there is also an overshoot (again in the third plot) around $k = 370$ is strange, since $\mathcal{P}_{min}$ has been lowered during that step. This could be due to the observer errors, which are plotted in Figure 7-5.

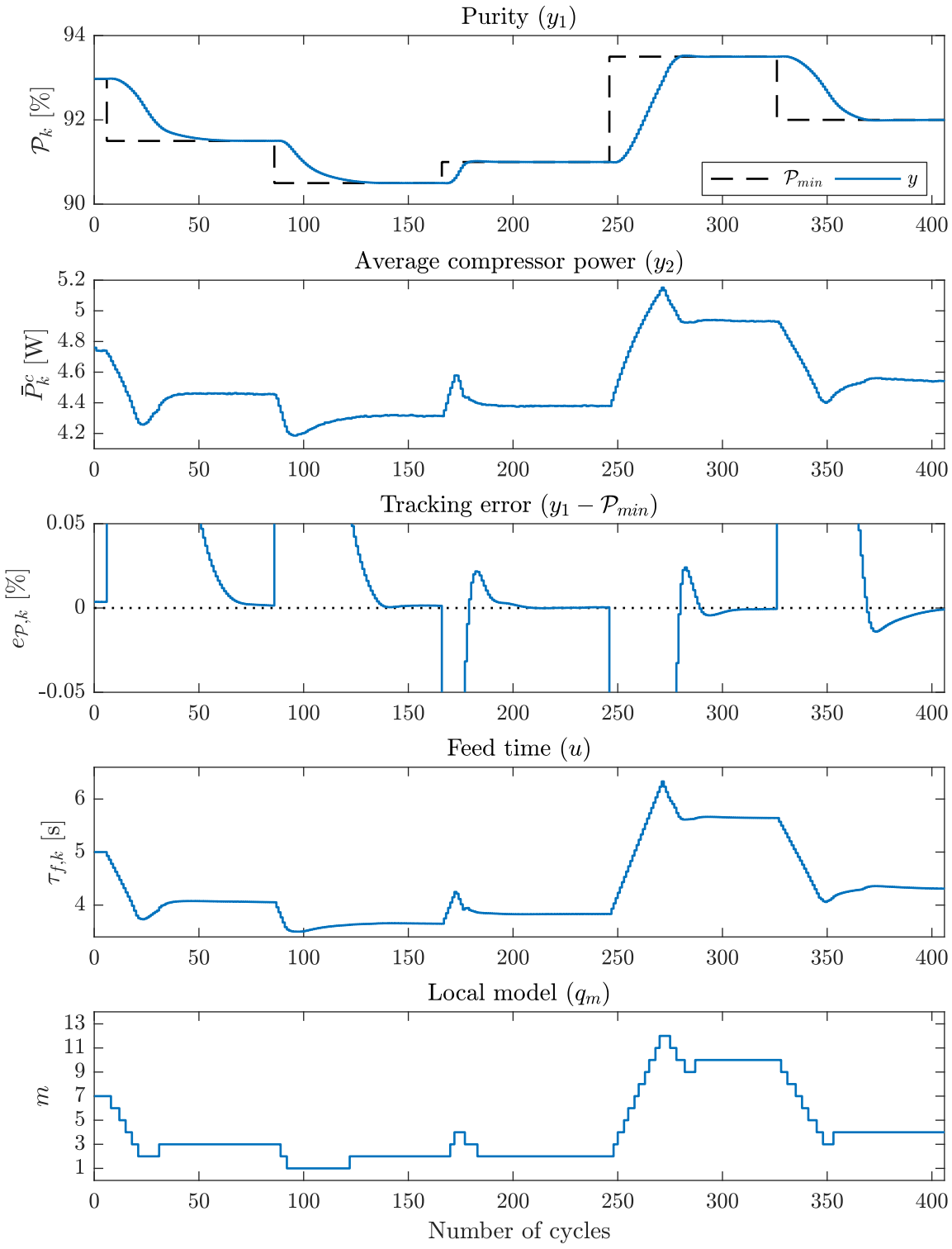

Figure 7-4: Simulation results of the POC under normal operation. $\mathcal{P}_{min}$ is the minimum allowable purity.

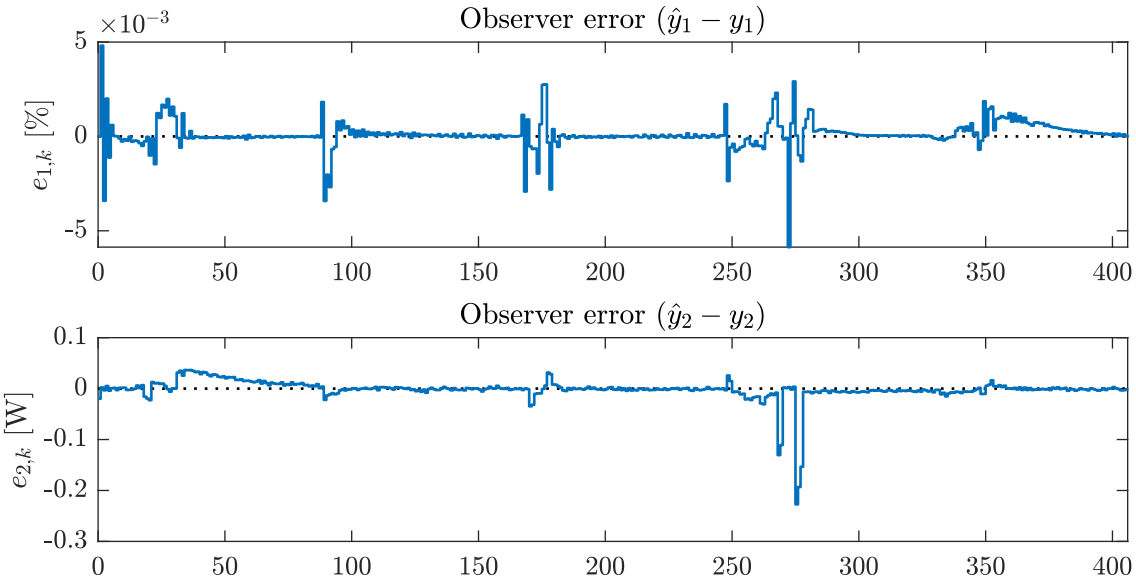

Figure 7-5: Errors between outputs of observer and outputs of the plant.

In Figure 7-5, the two observer errors, as explained in Section 7-1, are shown. There are some oscillations during transient behavior, but the system remains stable. However, there is a noticeable observer error in the top plot around cycle $k = 370$, which could explain the overshoot around the same cycle in Figure 7-4. There are also some observer errors visible between $\hat{y}_2$ and y_2, namely around cycles $k = 50$ and $k = 270$. All these observer errors are likely due modeling inaccuracies, but the exact cause is unknown.

All in all, the MPC strategy is able to control the POC adequately. There are some unexplained observer errors, but the system does not become unstable because of this.

7-2-1 Computation Time of Controller

Recall from Section 1-2, that one of the sub-objectives is that the controller should be able to run on the existing hardware of the POC. Since implementation of this MPC strategy on a real device is not in the scope of this research, this cannot be tested. As an alternative, the computation time of the controller will be assessed.

This is done by comparing the simulation times t_{sim} of the case discussed in this section, with a simulation without the controller (and observer), i.e. in open-loop. The optimized inputs from the first case will be used as inputs for the plant. The difference in simulation time, Δt_{sim}, can be seen as the time needed by the controller to calculate the inputs. If this is divided by the amount of cycles, 427, an estimate of the computation time of the controller per cycle, called t_{MPC}, is obtained:

$$t_{MPC} = \frac{1}{n_{cycles}} \Delta t_{sim}. \tag{7-1}$$

For the case with controller, $t_{sim} = 553.1s$, and for the case without controller, $t_{sim} = 373.7s$. Given the amount of cycles, 427, this means that $t_{MPC} \approx 0.42s$. In other words, on average

the controller needs $0.42s$ to compute a new input every cycle. The simulations have been done with MATLAB R2018a on a MacBook Pro, with a 2.7 GHz Intel Core i5 processor and 8GB of RAM.

At first sight, it seems that the built-in hardware on the POC will not be able to use this controller as-is. Given that there is a substantial difference in computing power between the built-in hardware and the laptop used for these simulations, t_{MPC} will likely be much higher when using the existing POC hardware. On the other hand, this code has not been optimized for usage on a real controller (e.g., compiled), and that might reduce the computation time. Further research is needed to conclude applicability on a real POC.

7-3 Case study: Sieve Bed Degradation

In this final section, a case study will be discussed: the performance of the MPC strategy will be evaluated when the POC is simulated with sieve bed degradation. The reason for sieve bed degradation was already given in Section 2-2-4: some water vapor present in the air gets irreversibly adsorbed by the adsorbent, with the result that some of the zeolite crystals are unable to adsorb nitrogen (or oxygen). This means the total adsorption capacity of the sieve bed decreases, changing the dynamics of the POC. Like said in Chapter 1, sieve bed degradation is not incorporated in the predictive model of the controller.

Degradation of the sieve beds has been modeled by setting some of the values of the adsorbed phase concentration q to zero (see Section 3-3-2), such that for some axial locations in the sieve beds, adsorption is no longer possible:

$$\frac{dq_{i,n}^{j}}{dt} = 0, \quad q_{i,n}^{j}(t_0) = 0, \quad \begin{matrix} j \in \{1,2\}, \\ i \in \{N_2, O_2\}, \end{matrix} \quad n = [1, \ldots, N_{degr}]. \tag{7-2}$$

Here $q_{i,n}^{j}(t_0)$ denotes the initial condition, and N_{degr} is the amount of volumes that are degraded, so for e.g., 10% sieve bed degradation, $N_{degr} = 10$. Different levels of degradation will be evaluated and compared to the non-degraded case: 5%, 10% and 20% sieve bed degradation.

The effect of sieve bed degradation on the I/O relationships can be found in Figure 7-6. Multiple observations can be made. First of all, since the effective sieve bed length gets shorter for increasing degradation, nitrogen breakthrough will happen earlier and the maximum achievable purity also decreases. Also notice that for low τ_f, the purity rises for increasing sieve bed degradation. This is caused by the increase of pressure in the sieve beds, leading to a higher adsorption capacity. This increase in pressure is due to the accumulation of nitrogen molecules that cannot be adsorbed by the degraded section of the sieve beds; more molecules present leads to a faster rise in pressure. This higher average pressure can also be seen in the right plot, showing the I/O relationship of $\bar{P}^c$ and τ_f. This all means that for increasing sieve bed degradation, the peak of the purity will move down and to the left (to lower τ_f), whereas the average compressor power will only rise (given that the mass flow out of the compressor stays constant). A control strategy that adjusts the timings based on the degradation would therefore need to decrease the feed time τ_f for increasing sieve bed degradation, given that adhering to $\mathcal{P}_{min}$ is still feasible.

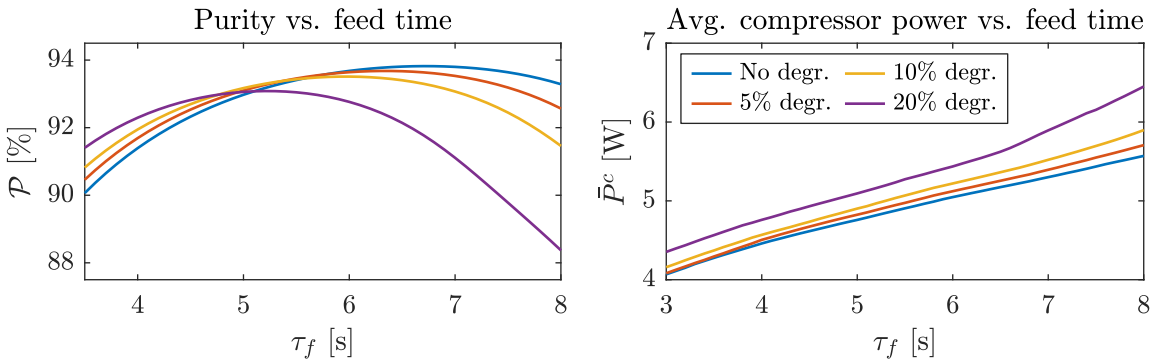

Figure 7-6: I/O relationships at CSS of the POC with various levels of sieve bed degradation compared to those of a POC without degradation.

Ideally, the developed MPC strategy would be compared to the current control method of the POC. Recall from Section 2-1-1, that the current control method is to tune the valve timings based on factory calibration: as the adsorption capacity of the sieve beds decreases over time, the timings are adjusted such that the constraints on the flow of oxygen-enriched air are not violated. However, these timings and adjustments are not available for comparison, and thus the performance of this open-loop strategy is unknown. Therefore, the controller will be compared with two open-loop strategies: one that has no knowledge of the degradation, and one that has full knowledge (i.e. perfect adjustments to the timings). This way the performance of the controller can be compared with a worst-case scenario and an optimal one. In Section 7-2 it was explained that for minimal power consumption, $\mathcal{P} = \mathcal{P}_{min}$. With that in mind, Figure 7-6 will be used to provide inputs for the two open-loop cases: the strategy without knowledge uses the "no degradation" I/O relationships to steer the purity to $\mathcal{P}_{min}$ (regardless of any degradation), and the strategy with full knowledge uses the I/O relationships of the respective level of degradation to calculate the inputs.

Three control strategies are evaluated (MPC, best-case open-loop and worst-case open-loop), as well as four degrees of sieve bed degradation: 5%, 10%, 20% and no degradation. Each of these cases will be simulated three times, with different values for $\mathcal{P}_{min}$ (91.5%, 92.5% and 93%). So in total, 36 scenarios are simulated. The performances of the control strategies are measured by the CSS outputs: $\mathcal{P}_{CSS}$ and $\bar{P}^c_{CSS}$. The results can be found in Table 7-1, and in Figure 7-7 the results of simulations with $\mathcal{P}_{min} = 92.5\%$ have been plotted as an example.

As can be seen from Table 7-1, the performance of the MPC strategy is identical to the (best-case) open-loop strategy when no degradation is considered; the fact that MPC can control such a system was already discussed in the previous section. For 5% sieve bed degradation, the difference in performance between the best-case open-loop strategy and the MPC strategy is still small; a difference in $\bar{P}^c_{CSS}$ of 0.01 to 0.03 W. However, for increasing degradation of the sieve beds, the performance of the MPC strategy worsens. For e.g., $\mathcal{P}_{min} = 93\%$ and a sieve bed degradation of 20%, the MPC performance is closer to the performance of the worst-case open-loop strategy than that of the best-case open-loop strategy.

It is clear that not incorporating the sieve bed degradation in the predictive model leads to performance limitations when the POC is simulated with degraded sieve beds. The MPC strategy outperforms a worst-case open-loop strategy, but that is to be expected since MPC uses feedback and has knowledge of the system. On the other hand, comparing MPC with

a "best-case" scenario is not really fair; it is unlikely that the current control method is able to perfectly adjust the valve timings to counter the degradation. Nevertheless, to truly counter sieve bed degradation, the MPC strategy needs to be improved. Possible solutions are to implement an adaptive control approach, or to incorporate the degradation into the predictive models of the controller.

Amount of degradation	MPC strategy $\mathcal{P}_{CSS}$ [%]	MPC strategy $\bar{P}^c_{CSS}$ [W]	Open-loop (best-case) $\mathcal{P}_{CSS}$ [%]	Open-loop (best-case) $\bar{P}^c_{CSS}$ [W]	Open-loop (worst-case) $\mathcal{P}_{CSS}$ [%]	Open-loop (worst-case) $\bar{P}^c_{CSS}$ [W]
	$\mathcal{P}_{min} = 91.5\%$					
none	91.50	4.46	91.50	4.46	-	-
5%	91.55	4.47	91.50	4.46	91.79	4.51
10%	91.63	4.52	91.50	4.47	92.04	4.58
20%	91.90	4.66	91.50	4.58	92.36	4.77
	$\mathcal{P}_{min} = 92.5\%$					
none	92.50	4.63	92.50	4.63	-	-
5%	92.64	4.69	92.50	4.66	92.69	4.71
10%	92.75	4.74	92.50	4.68	92.82	4.76
20%	92.84	4.94	92.50	4.81	92.90	4.96
	$\mathcal{P}_{min} = 93\%$					
none	93.00	4.75	93.00	4.75	-	-
5%	93.06	4.80	93.00	4.79	93.12	4.82
10%	93.10	4.86	93.00	4.82	93.19	4.90
20%	93.05	5.08	93.00	5.02	93.06	5.09

Table 7-1: Results of simulations with three different control methods, four levels of sieve bed degradation, and three different values for the minimal purity value $\mathcal{P}_{min}$. Note that the worst-case and best-case open-loop strategies are the same for "no degradation", therefore the results of the former have been left out.

7-4 Concluding Remarks

In this chapter, simulations of the POC and the controller have been discussed. It has been shown that the controller is able to steer the plant to the desired purity, and that the system is stable. However, any change in dynamics, e.g., due to sieve bed degradation, results in sub-optimal performance. Further development of the controller is needed to also fully cover these cases.

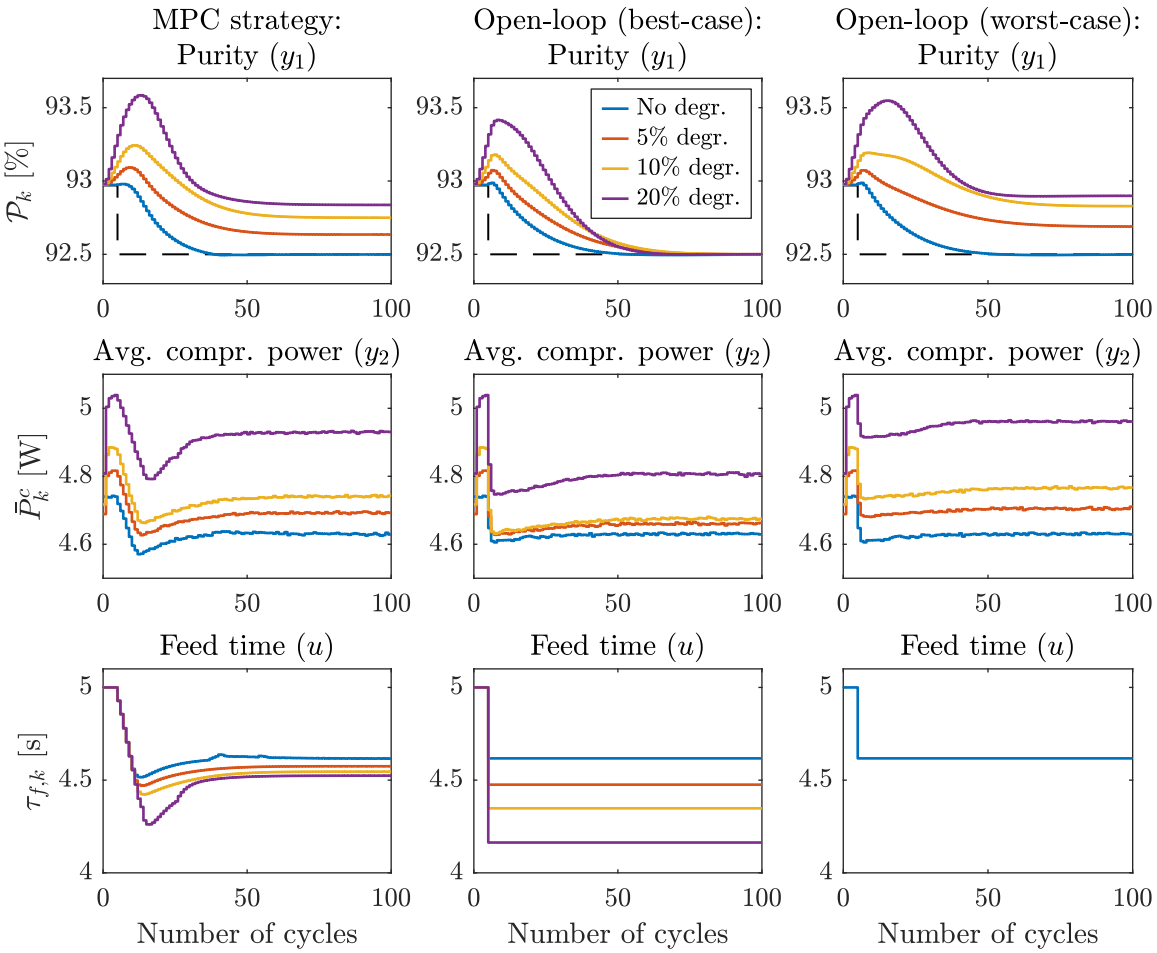

Figure 7-7: Performance comparisons between the three discussed control methods: MPC, best-case open-loop, and worst-case open-loop, on a POC with various degrees of sieve bed degradation. The minimal purity value is $\mathcal{P}_{min} = 92.5\%$, shown by the black dashed line in the top three plots.

Chapter 8

Conclusions

In this final chapter, first a summary of the research will be given. After that, the research objective, introduced in Section 1-2, will be revisited see whether the goal of this thesis has been achieved. Lastly, some recommendations and suggestions for future work are provided.

8-1 Summary

In this thesis, a switched linear Model Predictive Control (MPC) method has been developed, that minimizes the power consumption of a Portable Oxygen Concentrator (POC), while respecting the constraints on the purity. POCs are mobile devices that provide oxygen-enriched air, by separating nitrogen and oxygen with a cyclic process called Pressure Swing Adsorption (PSA). The aim of the controller is to optimize this process, such that no oxygen is wasted inside the device and the desired purity is held. This in turn ensures that the compressor can operate at its lowest possible power output.

The POC has two sieve beds, in which nitrogen is adsorbed and oxygen can flow through to an oxygen tank. These sieve beds alternate between adsorbing nitrogen (i.e. producing oxygen) at high pressure, and exhausting this nitrogen (i.e. regenerating the sieve bed) at low pressure. This is done in an alternating manner, such that one sieve bed is producing and the other regenerating. The dynamics inside these sieve beds manifest themselves as traveling concentration waves: this Mass Transfer Zone (MTZ), where the adsorption of nitrogen happens inside a sieve bed, propagates through a sieve bed. If this zone reaches the end of the sieve bed, nitrogen leaks through and the purity (i.e. the concentration of oxygen in the tank) drops. The input of the POC is the time that a sieve bed is producing oxygen every cycle, called the feed time τ_f.

A high-fidelity model of the POC has been developed in order to simulate the device and design a controller. The coupled PDEs that describe the dynamics of the sieve beds are rewritten as coupled ODEs by spatially discretizing the equations, such that any ODE solver is able to simulate the process. This high-fidelity model is subsequently validated by simulating it

in open-loop, and the described dynamics are shown. Because of its complexity, this high-fidelity model is not suited as a predictive model for the MPC strategy. Therefore, a hybrid batch model has also been developed. This model encompasses the cycle-to-cycle dynamics of the POC; two outputs, the purity $\mathcal{P}$ and the average compressor power $\bar{P}^c$ are sampled once per cycle. The model has a hybrid linear structure: 13 linear models have been identified for disjoint convex sets of the input space, which are connected to each other with a hybrid automaton.

MPC is an optimization-based control method. At every step, a cost function is minimized over a finite horizon that produces an optimal input u. The MPC strategy that has been used can be described as switched linear MPC: instead of optimizing the complete hybrid batch model, only the local model corresponding to the previous input is optimized. This way a convex optimization problem is obtained. The states of the batch model are unknown, and are used in the cost function. Therefore an observer also has been developed, which uses the reset maps of the hybrid batch model to switch between local models.

Finally, the results of the simulations with the newly designed controller are discussed. First, some design choices regarding the observer are justified. After that, the performance of the MPC strategy is evaluated. In these simulations, the minimum purity value $\mathcal{P}_{min}$ is subjected to step changes, to show that the controller is able to handle different purity settings, and can switch between them. The results show that the MPC strategy is able to minimize the power consumption, steering the purity $\mathcal{P}$ to $\mathcal{P}_{min}$. Finally, a case study has been done, in which the POC is subjected to various degrees of sieve bed degradation. The performance of the MPC strategy is compared to two open-loop cases: a worst-case, which does not take the degradation into account, and a best-case, which perfectly counters the degradation (i.e. minimizes the power consumption). Although the performance of the controller is acceptable for low levels of sieve bed degradation (i.e. 5%), for increasing levels (up to 20%), the performance of the MPC strategy worsens compared to the best-case scenario.

8-2 Evaluation of Research Objective

Now the research objective will be evaluated. The main research objective was to:

> *Develop a model predictive controller that minimizes the power consumption of the POC, while keeping the oxygen concentration in the tank above a certain limit.*

And the two sub-objectives were to:

1. *Develop a model that is capable of accurately simulating the dynamics of a POC.*
2. *Develop a simplified predictive model for the controller, that is simple enough such that it could be implemented on a POC.*

As the results in this thesis show, a power-minimizing MPC strategy has been successfully developed for a model of a POC with fully functional sieve beds (i.e. no sieve bed degradation). A high-fidelity model of the POC has been designed and developed, and open-loop simulations

show that the dynamics of the POC resemble the dynamics of similar studies on POCs. Therefore the first sub-objective is deemed to be met. The design of a predictive model for the controller has also been successful: validation of the local models has shown that the cycle-to-cycle dynamics are captured adequately, which is substantiated by the performance of the controller. However, it cannot be guaranteed that this predictive model is simple enough to use on a POC. Further research is needed to answer that question.

The performance of the MPC strategy has also been evaluated on a POC that is subjected to sieve bed degradation. Simulation results have shown that the MPC strategy is close to optimal for small levels of degradation, but for increasing sieve bed degradation, it fails to minimize the power consumption. Given all the results in this thesis, the main conclusion is that the developed MPC strategy needs further improvement if it is to replace the current control method of the POC. Some suggestions for improvements are given in Section 8-3.

8-3 Recommendations and Future Work

This section will discuss some possible improvements and ideas that could be developed in the future.

- Incorporate sieve bed degradation.
  Like discussed in the previous section, sieve bed degradation needs to be incorporated in the MPC strategy if MPC is to replace the current control method. One possibility is to incorporate this degradation into the model, but due to the large time scale this becomes difficult. Another option is to use adaptive or hybrid MPC techniques that can handle the change in dynamics.

- Add compressor RPM as input.
  The addition of a second input, the compressor RPM, would regulate the mass flow into the compressor. Although this mass flow does not change much during a normal cycle, also having this input will provide a slight increase in efficiency of the PSA process, and different flow settings can be simulated.

- Identification of local models.
  The current identification method for the 13 batch models is tedious. Because the data sets use one sample per cycle, creating a sufficiently long identification data set takes a considerable amount of time. If the MPC strategy is to be applied in the real world, research towards a different, faster identification method is strongly recommended.

- Hybrid MPC
  The chosen MPC approach is not hybrid: only one local model is optimized every time step, and the switching is not incorporated in the optimization. The MPC strategy could therefore be improved by incorporating this switching, bearing in mind that convexity should ideally be preserved.

- (Re)consider other control strategies
  The performance of MPC on a POC with sieve bed degradation is not optimal. Other control strategies, e.g., conventional feedback control, might be better suited to steer

the system to $\mathcal{P}_{min}$, such that power consumption is minimized. The identified local linear models, and of course the high-fidelity model of the POC, can be used to develop and benchmark such control strategies.

Appendix A

Appendix

A-1 State-Space Matrices for Local Models

The state-space matrices for the local models can be found in Eqs. (A-1), (A-2) and (A-3).

$$A_m = \begin{bmatrix} & & a_m & & & \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad m = [1, \ldots, 13],$$

$$\begin{aligned}
a_1 &= \begin{bmatrix} 3.43 & -4.54 & 2.87 & -0.847 & 0.0902 & -3.5 \end{bmatrix}, \\
a_2 &= \begin{bmatrix} 3.34 & -4.26 & 2.56 & -0.691 & 0.0606 & -3.75 \end{bmatrix}, \\
a_3 &= \begin{bmatrix} 3.41 & -4.45 & 2.72 & -0.753 & 0.0691 & -4.0 \end{bmatrix}, \\
a_4 &= \begin{bmatrix} 3.19 & -3.92 & 2.27 & -0.604 & 0.0568 & -4.25 \end{bmatrix}, \\
a_5 &= \begin{bmatrix} 3.13 & -3.76 & 2.11 & -0.525 & 0.0421 & -4.51 \end{bmatrix}, \\
a_6 &= \begin{bmatrix} 3.28 & -4.11 & 2.41 & -0.636 & 0.0582 & -4.75 \end{bmatrix}, \\
a_7 &= \begin{bmatrix} 3.14 & -3.76 & 2.12 & -0.542 & 0.0416 & -5.0 \end{bmatrix}, \\
a_8 &= \begin{bmatrix} 3.03 & -3.49 & 1.88 & -0.465 & 0.0367 & -5.26 \end{bmatrix}, \\
a_9 &= \begin{bmatrix} 3.0 & -3.4 & 1.79 & -0.426 & 0.0312 & -5.5 \end{bmatrix}, \\
a_{10} &= \begin{bmatrix} 2.63 & -2.55 & 1.16 & -0.258 & 0.0188 & -5.75 \end{bmatrix}, \\
a_{11} &= \begin{bmatrix} 3.22 & -3.91 & 2.18 & -0.543 & 0.0479 & -6.0 \end{bmatrix}, \\
a_{12} &= \begin{bmatrix} 3.12 & -3.59 & 1.8 & -0.34 & 0.00829 & -6.25 \end{bmatrix}, \\
a_{13} &= \begin{bmatrix} 2.86 & -2.95 & 1.29 & -0.199 & 0.00508 & -6.5 \end{bmatrix}.
\end{aligned} \tag{A-1}$$

$$B_m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad m = [1, \ldots, 13], \tag{A-2}$$

$$\begin{aligned}
C_1 &= \begin{bmatrix} 0.00448 & 0.0857 & -0.176 & 0.0989 & -0.0114 & 90.1 \\ 0.159 & -0.529 & 0.653 & -0.355 & 0.0717 & 2.87 \end{bmatrix}, & D_1 &= \begin{bmatrix} -0.0249 \\ 0.396 \end{bmatrix}, \\
C_2 &= \begin{bmatrix} -0.011 & 0.123 & -0.202 & 0.0987 & -0.00836 & 90.9 \\ 0.152 & -0.497 & 0.601 & -0.32 & 0.063 & 3.03 \end{bmatrix}, & D_2 &= \begin{bmatrix} -0.019 \\ 0.351 \end{bmatrix}, \\
C_3 &= \begin{bmatrix} -0.00907 & 0.106 & -0.182 & 0.094 & -0.009 & 91.5 \\ 0.122 & -0.405 & 0.498 & -0.269 & 0.0537 & 3.06 \end{bmatrix}, & D_3 &= \begin{bmatrix} -0.0234 \\ 0.344 \end{bmatrix}, \\
C_4 &= \begin{bmatrix} -0.00422 & 0.08 & -0.128 & 0.0588 & -0.00336 & 92.0 \\ 0.112 & -0.347 & 0.4 & -0.203 & 0.0376 & 3.33 \end{bmatrix}, & D_4 &= \begin{bmatrix} -0.0315 \\ 0.28 \end{bmatrix}, \\
C_5 &= \begin{bmatrix} -0.00524 & 0.0735 & -0.111 & 0.048 & -0.00133 & 92.5 \\ 0.109 & -0.333 & 0.378 & -0.189 & 0.0343 & 3.42 \end{bmatrix}, & D_5 &= \begin{bmatrix} -0.0334 \\ 0.261 \end{bmatrix}, \\
C_6 &= \begin{bmatrix} -0.0117 & 0.0883 & -0.132 & 0.0611 & -0.00186 & 92.8 \\ 0.101 & -0.322 & 0.383 & -0.199 & 0.0373 & 3.48 \end{bmatrix}, & D_6 &= \begin{bmatrix} -0.0294 \\ 0.25 \end{bmatrix}, \\
C_7 &= \begin{bmatrix} -0.0146 & 0.0898 & -0.123 & 0.0549 & -0.00602 & 93.2 \\ 0.0903 & -0.275 & 0.309 & -0.154 & 0.0292 & 3.54 \end{bmatrix}, & D_7 &= \begin{bmatrix} -0.0363 \\ 0.24 \end{bmatrix}, \\
C_8 &= \begin{bmatrix} -0.0297 & 0.135 & -0.167 & 0.0695 & -0.00677 & 93.4 \\ 0.0862 & -0.255 & 0.279 & -0.136 & 0.0259 & 3.53 \end{bmatrix}, & D_8 &= \begin{bmatrix} -0.025 \\ 0.244 \end{bmatrix}, \\
C_9 &= \begin{bmatrix} -0.0165 & 0.0872 & -0.108 & 0.0423 & -0.00309 & 93.6 \\ 0.0797 & -0.233 & 0.251 & -0.118 & 0.0212 & 3.57 \end{bmatrix}, & D_9 &= \begin{bmatrix} -0.0337 \\ 0.239 \end{bmatrix}, \\
C_{10} &= \begin{bmatrix} -0.0159 & 0.076 & -0.0745 & 0.0208 & -0.0038 & 93.8 \\ 0.0771 & -0.197 & 0.179 & -0.0706 & 0.0117 & 3.64 \end{bmatrix}, & D_{10} &= \begin{bmatrix} -0.0333 \\ 0.23 \end{bmatrix}, \\
C_{11} &= \begin{bmatrix} -0.0244 & 0.111 & -0.144 & 0.0629 & -0.00458 & 93.8 \\ 0.0683 & -0.215 & 0.247 & -0.121 & 0.021 & 3.78 \end{bmatrix}, & D_{11} &= \begin{bmatrix} -0.0253 \\ 0.209 \end{bmatrix}, \\
C_{12} &= \begin{bmatrix} -0.0191 & 0.0909 & -0.115 & 0.0436 & 1.77 \cdot 10^{-4} & 94.0 \\ 0.0664 & -0.204 & 0.228 & -0.107 & 0.0172 & 3.94 \end{bmatrix}, & D_{12} &= \begin{bmatrix} -0.0326 \\ 0.184 \end{bmatrix}, \\
C_{13} &= \begin{bmatrix} -0.0209 & 0.0896 & -0.0964 & 0.0255 & 0.00256 & 94.0 \\ 0.0639 & -0.179 & 0.177 & -0.0696 & 0.00794 & 4.01 \end{bmatrix}. & D_{13} &= \begin{bmatrix} -0.0368 \\ 0.176 \end{bmatrix}.
\end{aligned} \tag{A-3}$$

A-2 Observer Gains

$$
\begin{aligned}
L_1 &= \begin{bmatrix} 2.49 & 1.70 & 0.659 & -0.694 & -2.41 & 0 \end{bmatrix}^T \cdot 10^4, \\
L_2 &= \begin{bmatrix} 0.888 & 1.00 & 1.13 & 1.28 & 1.45 & 0 \end{bmatrix}^T \cdot 10^6, \\
L_3 &= \begin{bmatrix} -1.30 & -1.34 & -1.39 & -1.43 & -1.48 & 0 \end{bmatrix}^T \cdot 10^7, \\
L_4 &= \begin{bmatrix} 1.01 & 1.24 & 1.49 & 1.74 & 1.95 & 0 \end{bmatrix}^T \cdot 10^4, \\
L_5 &= \begin{bmatrix} 0.552 & 0.792 & 1.07 & 1.36 & 1.64 & 0 \end{bmatrix}^T \cdot 10^4, \\
L_6 &= \begin{bmatrix} 2.24 & 2.94 & 3.57 & 4.05 & 4.32 & 0 \end{bmatrix}^T \cdot 10^4, \\
L_7 &= \begin{bmatrix} -0.811 & -0.874 & -0.942 & -1.01 & -1.09 & 0 \end{bmatrix}^T \cdot 10^5, \\
L_8 &= \begin{bmatrix} -1.09 & -1.13 & -1.18 & -1.23 & -1.30 & 0 \end{bmatrix}^T \cdot 10^5, \\
L_9 &= \begin{bmatrix} -2.17 & -2.39 & -2.62 & -2.86 & -3.13 & 0 \end{bmatrix}^T \cdot 10^4, \\
L_{10} &= \begin{bmatrix} -3.23 & -3.62 & -4.03 & -4.43 & -4.73 & 0 \end{bmatrix}^T \cdot 10^3, \\
L_{11} &= \begin{bmatrix} -0.676 & -0.793 & -0.931 & -1.09 & -1.28 & 0 \end{bmatrix}^T \cdot 10^7, \\
L_{12} &= \begin{bmatrix} -1.21 & -1.27 & -1.33 & -1.40 & -1.48 & 0 \end{bmatrix}^T \cdot 10^5, \\
L_{13} &= \begin{bmatrix} -1.13 & -1.17 & -1.21 & -1.24 & -1.27 & 0 \end{bmatrix}^T \cdot 10^4,
\end{aligned}
\tag{A-4}
$$

Bibliography

[1] S. Sircar and A. L. Myers, *Gas separation by zeolites*, vol. 1063. Marcel Dekker Inc.: New York, 2003.

[2] J. Santos, A. Portugal, F. Magalhaes, and A. Mendes, "Simulation and optimization of small oxygen pressure swing adsorption units," *Industrial & engineering chemistry research*, vol. 43, no. 26, pp. 8328–8338, 2004.

[3] M. Bitzer, "Model-based nonlinear tracking control of pressure swing adsorption plants," in *Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems*, pp. 403–418, Springer, 2005.

[4] COPD Working Group, "Long-term oxygen therapy for patients with chronic obstructive pulmonary disease (COPD): an evidence-based analysis," *Ontario health technology assessment series*, vol. 12, no. 7, p. 1, 2012.

[5] V. Kim, J. O. Benditt, R. A. Wise, and A. Sharafkhaneh, "Oxygen therapy in chronic obstructive pulmonary disease," *Proceedings of the American Thoracic Society*, vol. 5, no. 4, pp. 513–518, 2008.

[6] C. A. Grande, "Advances in pressure swing adsorption for gas separation," *ISRN Chemical Engineering*, 2012.

[7] J. Santos, A. Portugal, F. Magalhaes, and A. Mendes, "Optimization of medical PSA units for oxygen production," *Industrial & engineering chemistry research*, vol. 45, no. 3, pp. 1085–1096, 2006.

[8] G. Carta, *Perry's Chemical Engineers' Handbook. Section 16: Adsorption and Ion Exchange*. McGraw-Hill Pub., 2008.

[9] "Acceptance Criteria for Portable Oxygen Concentrators." https://www.faa.gov/about/initiatives/cabin_safety/portable_oxygen/. Online, accessed: 05-mar-2020.

[10] D. W. Hunter and F. R. Frola, "Energy efficient oxygen concentrator," Nov. 4 2008. US Patent 7,445,663.

[11] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[12] N. D. Hutson, S. C. Zajic, and R. T. Yang, "Influence of residual water on the adsorption of atmospheric gases in Li- X zeolite: Experiment and simulation," *Industrial & Engineering Chemistry Research*, vol. 39, no. 6, pp. 1775–1780, 2000.

[13] H. Khajuria, *Model-based Design, Operation and Control of Pressure Swing Adsorption Systems*. PhD thesis, Imperial College London, 2011.

[14] A. Agarwal, L. T. Biegler, and S. E. Zitney, "A superstructure-based optimal synthesis of PSA cycles for post-combustion CO_2 capture," *AIChE journal*, vol. 56, no. 7, pp. 1813–1828, 2010.

[15] R. T. Yang, *Adsorbents: fundamentals and applications*. John Wiley & Sons, 2003.

[16] D. Ruthven, S. Farooq, and K. Knaebel, *Pressure Swing Adsorption*. VCH Publishers, Inc., 1994.

[17] C. W. Skarstrom, "Method and apparatus for fractionating gaseous mixtures by adsorption," July 12 1960. US Patent 2,944,627.

[18] N. H. Berlin, "Method for providing an oxygen-enriched environment," Oct. 25 1966. US Patent 3,280,536.

[19] J.-G. Jee, J.-S. Lee, and C.-H. Lee, "Air separation by a small-scale two-bed medical O_2 pressure swing adsorption," *Industrial & engineering chemistry research*, vol. 40, no. 16, pp. 3647–3658, 2001.

[20] M. Bitzer, *Control of Periodically Operated Adsorption Processes*. VDI-Verlag, 2004.

[21] S. Farooq, D. Ruthven, and H. Boniface, "Numerical simulation of a pressure swing adsorption oxygen unit," *Chemical Engineering Science*, vol. 44, no. 12, pp. 2809–2816, 1989.

[22] K. G. Teague and T. F. Edgar, "Predictive dynamic model of a small pressure swing adsorption air separation unit," *Industrial & engineering chemistry research*, vol. 38, no. 10, pp. 3761–3775, 1999.

[23] D. G. Pelletier, M. S. McClain, and M. P. Chekal, "Replaceable sieve bed for portable oxygen concentrator," Mar. 5 2013. US Patent 8,388,745.

[24] A. Agarwal, *Advanced strategies for optimal design and operation of pressure swing adsorption processes*. PhD thesis, Carnegie Mellon University, 2010.

[25] P. A. Webley and J. He, "Fast solution-adaptive finite volume method for PSA/VSA cycle simulation; 1 single step simulation," *Computers & Chemical Engineering*, vol. 23, no. 11-12, pp. 1701–1712, 2000.

[26] P. Cruz, J. Santos, F. Magalhaes, and A. Mendes, "Cyclic adsorption separation processes: analysis strategy and optimization procedure," *Chemical Engineering Science*, vol. 58, no. 14, pp. 3143–3158, 2003.

[27] A. Beeyani, K. Singh, R. Vyas, S. Kumar, and S. Kumar, "Parametric studies and simulation of PSA process for oxygen production from air," *Polish journal of chemical technology*, vol. 12, no. 2, pp. 18–28, 2010.

[28] "Flow Coefficients k_v for Liquid, Steam or Gas." https://www.engineeringtoolbox.com/kv-coefficients-d_1931.html. Online, accessed: 09-sept-2019.

[29] "STP - Standard Temperature and Pressure & NTP - Normal Temperature and Pressure." https://www.engineeringtoolbox.com/stp-standard-ntp-normal-air-d_772.html. Online, accessed: 08-mar-2020.

[30] B. Nesbitt, *Handbook of valves and actuators: valves manual international*, ch. 2 - Properties of fluids. Elsevier, 2011.

[31] L. Jiang, L. T. Biegler, and V. G. Fox, "Simulation and optimization of pressure-swing adsorption systems for air separation," *AIChE Journal*, vol. 49, no. 5, pp. 1140–1157, 2003.

[32] S. Sircar and B. Hanley, "Production of oxygen enriched air by rapid pressure swing adsorption," *Adsorption*, vol. 1, no. 4, pp. 313–320, 1995.

[33] T. Nitta, T. Shigetomi, M. Kuro-oka, and T. Katayama, "An adsorption isotherm of multi-site occupancy model for homogeneous surface," *Journal of Chemical engineering of Japan*, vol. 17, no. 1, pp. 39–45, 1984.

[34] A. L. Myers and J. M. Prausnitz, "Thermodynamics of mixed-gas adsorption," *AIChE Journal*, vol. 11, no. 1, pp. 121–127, 1965.

[35] "Partial Differential Equations - MATLAB & Simulink - Mathworks Benelux." https://nl.mathworks.com/help/matlab/math/partial-differential-equations.html. Online, accessed: 05-sept-2019.

[36] R. Eymard, T. Gallouët, and R. Herbin, "Finite volume methods," *Handbook of numerical analysis*, vol. 7, pp. 713–1018, 2000.

[37] B. P. Leonard, "A stable and accurate convective modelling procedure based on quadratic upstream interpolation," *Computer methods in applied mechanics and engineering*, vol. 19, no. 1, pp. 59–98, 1979.

[38] G. W. Griffiths and W. E. Schiesser, "2 - linear advection equation," in *Traveling Wave Analysis of Partial Differential Equations* (G. W. Griffiths and W. E. Schiesser, eds.), pp. 7 – 45, Boston: Academic Press, 2012.

[39] A. W. Dowling, S. R. Vetukuri, and L. T. Biegler, "Large-scale optimization strategies for pressure swing adsorption cycle synthesis," *AIChE Journal*, vol. 58, no. 12, pp. 3777–3791, 2012.

[40] J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42. Springer Science & Business Media, 2013.

[41] H. Khajuria and E. N. Pistikopoulos, "Dynamic modeling and explicit/multi-parametric MPC control of pressure swing adsorption systems," *Journal of Process Control*, vol. 21, no. 1, pp. 151–163, 2011.

[42] M. D. Urich, R. R. Vemula, and M. V. Kothare, "Multivariable model predictive control of a novel rapid pressure swing adsorption system," *AIChE Journal*, vol. 64, no. 4, pp. 1234–1245, 2018.

[43] H. Peng, F. Couenne, and Y. Le Gorrec, "Robust control of a pressure swing adsorption process," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 7310–7315, 2011.

[44] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[45] "System Identification Toolbox - MATLAB." https://www.mathworks.com/products/sysid.html. Online, accessed: 04-feb-2020.

[46] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of Hammerstein-Wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[47] J.-W. van Wingerden and M. Verhaegen, "Closed-loop subspace identification of Hammerstein-Wiener models," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3637–3642, IEEE, 2009.

[48] L. Ljung, *System Identification: Theory for the User*. Prentice Hall information and system sciences series, Prentice Hall PTR, 1999.

[49] M. Green and J. B. Moore, "Persistence of excitation in linear systems," *Systems & Control Letters*, vol. 7, no. 5, pp. 351–360, 1986.

[50] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15. SIAM studies in applied mathematics, 1994.

[51] J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on automatic control*, vol. 48, no. 1, pp. 2–17, 2003.

[52] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "LPV control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, pp. 231–245, 2005.

[53] M. Lovera and G. Mercere, "Identification for gain-scheduling: a balanced subspace approach," in *Proceedings of the 2007 American Control Conference (ACC)*, pp. 858–863, IEEE, 2007.

[54] B. Paijmans, W. Symens, H. Van Brussel, and J. Swevers, "Identification of interpolating affine LPV models for mechatronic systems with one varying parameter," *European Journal of Control*, vol. 14, no. 1, pp. 16–29, 2008.

[55] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE transactions on automatic control*, vol. 45, no. 10, pp. 1864–1876, 2000.

[56] Z. Sun, *Switched linear systems: control and design*. Springer Science & Business Media, 2006.

[57] M. Johansson and A. Rantzer, "Computation of piecewise quadratic lyapunov functions for hybrid systems," in *Proceedings of the 1997 European Control Conference (ECC)*, pp. 2005–2010, IEEE, 1997.

[58] B. De Schutter and T. J. van den Boom, "MPC for continuous piecewise-affine systems," *Systems & Control Letters*, vol. 52, no. 3-4, pp. 179–192, 2004.

[59] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[60] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process dynamics and control*. John Wiley & Sons, 2010.

[61] L. Magni, R. Scattolini, and M. Tanelli, "Switched model predictive control for performance enhancement," *International Journal of Control*, vol. 81, no. 12, pp. 1859–1869, 2008.

[62] L. J. Bridgeman, C. Danielson, and S. Di Cairano, "Stability and feasibility of MPC for switched linear systems with dwell-time constraints," in *Proceedings of the 2016 American Control Conference (ACC)*, pp. 2681–2686, IEEE, 2016.

[63] E. W. Weisstein, "Convex Function – from Wolfram Mathworld." http://mathworld.wolfram.com/ConvexFunction.html. Online, accessed: 27-feb-2020.

[64] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, vol. 13. SIAM studies in applied mathematics, 1994.

[65] D. Luenberger, "An introduction to observers," *IEEE Transactions on automatic control*, vol. 16, no. 6, pp. 596–602, 1971.

Glossary

List of Acronyms

BC	Boundary Condition
CCF	Controllable Canonical Form
COPD	Chronic Obstructive Pulmonary Disease
CSS	Cyclic Steady State
FVM	Finite Volume Method
IAST	Ideal Adsorbed Solution Theory
I/O	Input/Output
LDF	Linear Driving Force
LPV	Linear Parameter Varying
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
MTZ	Mass Transfer Zone
NRMSE	Normalized Root Mean Square Error
N4SID	Numerical algorithm for Subspace IDentification
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
POC	Portable Oxygen Concentrator
PRBS	Pseudo-Random Binary Sequence
PSA	Pressure Swing Adsorption

PWA	Piecewise Affine
PWL	Piecewise Linear
RPSA	Rapid Pressure Swing Adsorption
SIT	System Identification Toolbox
STP	Standard conditions for Temperature and Pressure
QP	Quadratic Programming

List of Symbols

Greek Symbols

γ	Mole fraction	
Δu	Change in control input	
Δz	Length of control volume	[m]
ϵ	Bed voidage	
ε	CSS purity error	[%]
η	Adiabatic efficiency	
κ	Heat capacity ratio of air	
ρ	Density	[kg m^{-3}]
ρ_b	Bulk density	[kg m^{-3}]
ρ_p	Packing density	[kg m^{-3}]
τ_b	Feed time: duration of PSA steps 1 and 3	[s]
τ_f	Balance time: duration of PSA steps 2 and 4	[s]
τ_t	Total PSA cycle time	[s]

Latin Symbols

A_b	Cross sectional area of sieve bed	[m^2]
c	Gaseous phase concentration	[mol m^{-3}]
D_L	Axial dispersion coefficient	[m^2 s^{-1}]
e	Observer error	
G	Guard condition for hybrid automaton	
H^c	Compressor head	[J mol^{-1}]
H_c	Control horizon	
H_p	Prediction horizon	
K_{LDF}	Mass transfer constant for LDF model	[s^{-1}]
k	Langmuir equilibrium constant	[Pa^{-1}]
k_v	Valve flow coefficient	[sl min^{-1}]
$\mathcal{P}$	Purity	[%]
$\mathcal{P}_{min}$	Minimum allowable purity	[%]

P^c	Compressor power	[W]
p	Pressure	[Pa]
Q	Molar flow rate	[mol s^{-1}]
$Q_{\tilde{s}}$	Slack variable weighing matrix	
Q_u	Input weighing matrix	
Q_y	Output weighing matrix	
q	Adsorbed phase concentration	[mol kg^{-1}]
q^*	Equilibrium value for adsorbed phase concentration	[mol kg^{-1}]
q^s	Saturation value for adsorbed phase concentration	[mol kg^{-1}]
q	Local model	
R	Gas constant	[J K^{-1} mol^{-1}]
R	Reset map for hybrid automaton	
T	Temperature	[K]
t	Time	[s]
t_k	Time at which cycle k starts	[s]
$\mathcal{U}$	Input range	
U_m	Input range of local model m	
u	Input	
u_v	Binary valve input	
V	Volume	[m^3]
v	Interstitial velocity	[m s^{-1}]
x	State variable	
$\bar{x}$	Averaged variable; spatial or time	
$\hat{x}$	Observer state	
y	Output variable	
$\hat{y}$	Observer output	
z	Axial coordinate	[m]

Subscripts

amb	Ambient conditions
Ar	Argon
CSS	Denotes variable at Cyclic Steady State
d	Downstream
i	Index for component in gas mixture
k	Index for cycle
m, n	Index for local model
N_2	Nitrogen
n	Index for control volume
O_2	Oxygen
stp	Standard Conditions (STP)
u	Upstream

v	Valve

Superscripts

c	Compressor
j	Index for sieve bed
t	Oxygen tank