

University of Groningen

Toward controlled ultra-high vacuum chemical vapor deposition processes

Dresscher, Martijn

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2019

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Dresscher, M. (2019). *Toward controlled ultra-high vacuum chemical vapor deposition processes*. Rijksuniversiteit Groningen.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Chapter 2

Modeling of fluxes and partial pressures for controller design

We will begin the first part of the thesis by presenting our contributions to the modeling for control of the Ultra-high vacuum chemical vapor deposition (UHVCVD) process dynamics. The modeling framework that we present in this chapter is based on the evolution of flux magnitudes on surfaces inside a vacuum. These flows evolve in accordance with the free molecular flow (FMF) transport regime. An important part of this model is derived from fundamental physical laws and is validated with results from the literature. We show that our model is suitable for control design by presenting and numerically evaluating a state-of-the-art controller for a simple academic example. We will explain the FMF transport regime in further detail in Section 2.1. We subsequently present our control framework and its various components in Section 2.2. Section 2.3 is used to elaborate on methods to obtain one of the central modeling components and provides a verification based on results from the literature. We use Section 2.4 to present a state-of-the-art controller design and evaluate it numerically. This controller design is not used in the other chapters of the thesis. Lastly, we conclude this chapter with some remarks in Section 2.5.

2.1 Free molecular flow preliminaries

The FMF is a particle transport regime in environments that have a large Knudsen number ($\gg 1$). The Knudsen number is defined as the mean free path of a particle divided by a relevant length scale of travel (Knudsen 1909). In an ultra-high vacuum (UHV) environment, we find mean free path lengths mostly longer than one kilometer. The Knudsen number therefore quickly becomes very large. In free molecular flow, gas phase collisions accordingly occur so rarely that gas dynamics are dictated by the solid boundaries. Particularly, the angular distribution of departing particles is of interest, since such distribution in combination with the boundary geometry can be used to determine the flux evolution. These dynamics are essentially less complicated than flow dynamics in the laminar or the transitional regime, due to the lack of dependence on neighboring states. Furthermore, this absence of spatial dependence makes the use of partial differential equations unattractive, as they will be reduced to the influence of boundary conditions and vectors containing information

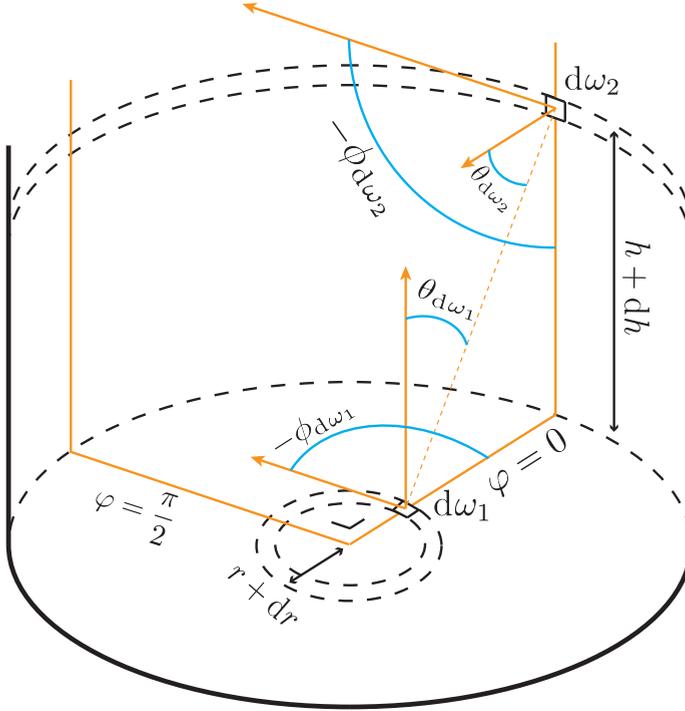


Figure 2.1: Knudsen cosine law coordinate illustration

A graphical interpretation of coordinates used in the Knudsen cosine law is shown. The considered infinitesimal areas are labeled as $d\omega_1$ and $d\omega_2$. Their relative locations are expressed in angles θ and ϕ belonging to each of the infinitesimal areas. The variable φ represents the relative rotation around the central axis of the cylinder.

on velocities. Moreover, the information on velocity vectors of species traversing the vacuum is not directly useful since the probability of gas-phase collisions is negligible. Consequently, what remains of interest, is the evolution of the fluxes interacting with the boundaries. The evolution of fluxes can be determined through a likeliness function, describing the fractions of particles contained in the flux, that will travel in a corresponding direction once they leave the boundary. The basis for this likeliness function is provided by the Knudsen cosine law (Knudsen 1915). Let us consider two surfaces $\omega_1, \omega_2 \in \mathbb{M}$, with \mathbb{M} the 2-dimensional inner surface (facing the vacuum) of the considered geometry. The Knudsen cosine law then provides the transfer between these surfaces, expressed as a function of their infinitesimals $d\omega_1$ and $d\omega_2$. It is, with reference to Fig. 2.1, given by

$$p_{d\omega_1}(d\omega_2) = \frac{\cos(\theta_{d\omega_1}) \cos(\theta_{d\omega_2})}{d_E(d\omega_1, d\omega_2)^2 \pi} d\omega_2, \quad (2.1)$$

where $p_{d\omega_1}(d\omega_2)$ is the fraction of particles that leaves an infinitesimal area $d\omega_1$ in the direction of $d\omega_2$, $\theta_{d\omega_1}$ is the angle between the normal of $d\omega_1$ and the line connecting $d\omega_1$ to $d\omega_2$, $\theta_{d\omega_2}$ is the angle between the normal of $d\omega_2$ and the line connecting $d\omega_1$ and $d\omega_2$, and $d_E(d\omega_1, d\omega_2)$ is the Euclidean distance between $d\omega_1$ and $d\omega_2$. In Fig. 2.1, we furthermore have $\phi_{d\omega_1}$ and $\phi_{d\omega_2}$ as the rotations around the normal of $d\omega_1$ and $d\omega_2$, respectively, and φ as the relative rotation of two infinitesimal surfaces $d\omega_1$ and $d\omega_2$ w.r.t. each other. Lastly, $p_{d\omega_2}(d\omega_1)$ can be obtained by replacing $d\omega_2$ by $d\omega_1$ and vice versa in (2.1).

The Knudsen cosine law is generally considered to hold well for weakly absorbing and rough surfaces, as the particles lose their direction of travel when adsorbed or are reflected in a direction following some likeliness influenced by surface microgeometry. In this thesis, the Knudsen cosine law will serve as the fundamental function describing these transfer fractions. Assuming that it is a static distribution, we can use it to compute constant transfer fractions between discrete space surfaces.

2.2 Modeling framework for FMF

We will use this section to present our free molecular flow modeling framework for the partial pressure controllable UHVCVD. Such a model should, in accordance with our motivation in Section 1.4.1, provide a connection between actuation signals and measured signals, e.g. reactor inputs and outputs. We accordingly require the model to produce partial pressure as an output.

Under vacuum condition, the pressure in a given volume satisfies the relation described by the ideal gas law

$$PV = NRT_a, \quad (2.2)$$

where P is the pressure, V a volume of interest, N the number of moles inside the atom cloud, R the ideal gas constant and T_a the average temperature of the atom cloud. In order to obtain P , we are accordingly interested in determining N and T_a for the precursors of interest.

When we consider N , the simplest way of modeling it is by considering the deposition chamber as a single volume. In this case, the variable N can be modeled by

$$\dot{N}(t) = \dot{N}_{in}(t) - \dot{N}_s(t) - \dot{N}_{out}(t), \quad (2.3)$$

with $t \in \mathbb{R}_{\geq 0}$ the time and where \dot{N}_{in} and \dot{N}_{out} are respectively the number of moles per second entering and exiting the volume and \dot{N}_s is number of moles per second that gets sorbed (or negatively; desorbed) to surfaces in \mathbb{M} . The model (2.3), which we will refer to as the moles model, is a lumped model that does not contain information on the flows occurring inside the volume of interest. As a result, it restricts us from gaining insight in the temperature-sensitive and spatially-dependent sorption characteristics.

We will proceed with presenting a modeling framework that allows for a more realistic description of the flows inside the deposition chamber. Such a modeling framework was previously introduced in (Dresscher et al. 2017) and in this work we consider an adaptation of the previously presented result. For this model, which we will refer to as the flux model, we are interested in the evolution of fluxes over time through a spatial discretization of the inner surface of a geometry containing a FMF regime. Let us discretize the geometry in n -elements. We further consider an incoming flux that can be manipulated by a controller and we define the partial pressure as the output variable. For this flux evolution (without leakage) we consider the relation

$$\bar{x}^+(t) = p^A(\bar{x}(t) - \dot{s}(t) + G(u(t), t)), \quad (2.4)$$

where $\bar{x}^+(t) \in \mathbb{R}^n$ is the flux magnitude at each discretized element after a (small) time step, $p^A \in \mathbb{R}^{n \times n}$ is a constant symmetric matrix describing scattering of the fluxes between surfaces in accordance with (2.1), $\bar{x}(t) \in \mathbb{R}^n$ is the flux magnitude, $\dot{s}(t)$ is the change in moles sorbed to the considered surface and $G(u(t), t)$ is the (discrete time) flux magnitude change caused by new precursor evaporation to each element in the inner surface, with $u(t)$ the input. Consequently, we consider the spatially discretized model as follows

$$\begin{aligned} \dot{\bar{x}}(t) &= (A - \delta I - L)\bar{x}(t) - (A - L)(f(\bar{x}(t), s(t), T(t)) - g(u(t), t)), \\ \dot{z}(t) &= f(\bar{x}(t), s(t), T(t)), \\ \dot{s}(t) &= z(t), \end{aligned} \quad (2.5)$$

$$\bar{y}(t) = h(\bar{x}(t), T_a(t)) = \frac{R}{V} \sqrt{\frac{T_a(t)\pi M}{8R}} \mathbf{1}^\top (p^A \cdot \ell) \bar{x}(t),$$

where \cdot indicates element-wise multiplication. In (2.5), the state $\bar{x}(t) \in \mathbb{R}^n$ represents atomic fluxes (in moles per second) on discrete surface areas $\{\omega_1, \omega_2, \dots, \omega_n\} \in \mathbb{M}$, $z(t)$ is the temporal derivative of $s(t)$, where the state $s(t) \in \mathbb{R}^n$ represents particles that are sorbed to the same collection of surfaces as considered for $\bar{x}(t)$. The symmetric matrix $A \in \mathbb{R}^{n \times n}$ describes the particle transfers that are calculated based on the Knudsen cosine law, between all of the n discrete surfaces, satisfying $\sum_i A_{i,j} = \delta$, for $j = 1, \dots, n$. Here, $\delta \in \mathbb{R}_{\geq 0}$ is a constant that corrects for the time scale of the particle transfer phenomena and $A = \delta p^A$. The matrix I is the identity matrix. The matrix $L \in \mathbb{R}_{\geq 0}^{n \times n}$ incorporates the leakage from the deposition chamber. The sorption function $f(\bar{x}(t), s(t), T(t))$ produces a n -dimensional vector that describes the non-linear sorption and desorption phenomena for each of the n discrete surfaces, with $T \in \mathbb{R}^n$ the associated collection of surface temperatures. These effects are subject to changes in the temperature and the chemical compositions of the surfaces. The input function $g(u(t), t)$ produces a n -dimensional vector that (non-linearly) relates the controlled input(s) $u(t) \in \mathbb{R}^m$ to fluxes directed at the n discrete surfaces. The time dependence represents possible changes in flux source characteristics. For the partial pressure output $\bar{y}(t)$, we have the function $h(\bar{x}(t), T_a(t))$ that relates the fluxes to the partial

pressure. The variable T_a is (as before) the average temperature of the atom cloud, M is the molar mass of the element of interest, R is the gas constant, $\mathbf{1}$ is a n -length column vector of ones and $\ell \in \mathbb{R}^{n \times n}$ a symmetric matrix containing average path lengths of the fluxes through the volume V , where the (i, j) -th entry corresponds to the path length from surface i to surface j .

The model (2.5) considers single input single output (e.g. one evaporation source is related to one partial pressure) as this is what we will use throughout this thesis, but this formulation can easily be expanded to include multiple evaporation sources and partial pressures. The dynamics of multiple elements (which can accordingly form molecules) are then coupled through the chemical sorption phenomena, which can be described by f . The formulation can furthermore easily be adapted to include a sink state, this state will then describe the leakage from the system as described by $\mathbf{1}^\top L \bar{x}(t)$. Having such a state can be beneficial as it (or rather its primitive) can be used to reconstruct the amount of moles that have been evaporated from an evaporation source.

Notice that when we take $n = 1$, (2.3) and (2.5) are very similar. More precisely, for (2.5) we obtain the simplified case with $A = \delta$ and we have

$$N = \bar{x} v^{-1} \ell, \quad (2.6)$$

with v the average speed of the atoms. For a constant v we then obtain

$$\dot{N} = \dot{\bar{x}} v^{-1} \ell. \quad (2.7)$$

Accordingly, with respect to (2.3), we have $\dot{N}_{out} = v^{-1} \ell L \bar{x}(t)$, $\dot{N}_{in} = v^{-1} \ell g(u(t), t)$ and $\dot{N}_s = v^{-1} \ell f(\cdot)$.

2.2.1 The transfer and leakage matrices

The transfer and leakage matrices, A and L , are directly dependent on design choices for the geometry holding the atom cloud, the geometry discretization and time scale of the phenomena.

The transfer matrix A has a fundamental base that is derived from the Knudsen cosine law. This part describes the fraction of incoming flux that is redirected to each discrete surface and will be denoted with p^A . This is in turn dependent on the size and orientation of these surfaces, hence causing the dependence on the design of the geometry and the choice of discretization. The effect of the time scale of the phenomena can subsequently be approximated by scaling the matrix p^A entirely by a constant δ , in order to obtain A . This is an approximation since such a time-scale changes with temperature and it may not be uniform. However, the effect of this non-uniformity is expected to be small and we choose not to incorporate it in the modeling. We will discuss methods to obtain p^A in Section 2.3.

The leakage matrix is dependent on the gaps in the geometry holding the atom cloud. It is typically desirable to have designed leakage in vacuum processes.

2.2.2 The sorption function

The sorption function vector $f(\bar{x}(t), s(t), T(t))$ has the primary purpose of incorporating the changes in fluxes that occur through sorption phenomena. For adsorptions and desorptions, there are generally three ways to obtain such a function: (i) by considering the physical laws that describe adsorption and desorption, (ii) by considering the semi-empirical vapor pressure functions (when modeling vapor pressure dynamics) and (iii) through empirical observations and system identification. The function $f(\cdot)$ should ideally satisfy the dynamics described by all three options. Notice that the function $f(\cdot)$ in fact complements the sorption already described by A . Indeed, the Knudsen cosine law is described to hold for weakly absorbing (or adsorbing) and rough surfaces, as we have discussed in Section 2.1. Some of the sorption dynamics are hence inherited through application of the Knudsen cosine law.

2.2.3 The input function

The input function $g(u(t), t)$ relates the input(s) u to the fluxes \bar{x} . Here, $u(t)$ is typically electrical current which in turn heats up an evaporation source. When the evaporation source is active, it loses mass and its behavior will therefore change over time. The evaporation sources are typically subject to variations. These variations occur between evaporation sources (e.g. run-to-run) and per evaporation source, where performance changes over time as precursor material evaporates. These variations can be dealt with by explicit characterization, modeling and control for such variation (through methods as presented in Part II of this thesis) or by implementation of traditional feedback control for the evaporated material partial pressures (as we will present in Chapter 4).

2.2.4 The output function

The output function $h(\bar{x}(t), T_a(t))$ serves to relate the flux states \bar{x} (and optionally s) to an estimated partial pressure \bar{y} , so that it can be compared to the measured pressure. Using the same reasoning as in the introduction of this section, we can use (2.2) to relate an estimate on the number of moles in a volume V , denoted by N to the estimated pressure \bar{y} . Equation (2.2) accordingly becomes

$$\bar{y} = \frac{NRT_a}{V}. \quad (2.8)$$

For obtaining N , we can use our knowledge on the flux magnitudes on the surfaces.

This relation is then given by

$$N = \frac{1}{v} \mathbf{1}^\top (q \cdot \ell) \mathbf{1}, \quad (2.9)$$

with \cdot an element-wise multiplication, N in moles, v in meters per second, $q \in \mathbb{R}^{n \times n}$ the directed fluxes (from surface i to j) in moles per second and $\ell \in \mathbb{R}^{n \times n}$ a symmetric matrix containing the corresponding path lengths of the fluxes through V in meters.

We can approximate q by using the information on how strongly each surface contributes to the total flux on any other surface, which is stored in A , or more precisely, in p^A . We approximate the fluxes between surfaces accordingly through

$$q = p^A \cdot (\mathbf{1}\bar{x}^\top), \quad (2.10)$$

where $\mathbf{1}$ is an n -length column vector of ones.

For ℓ , we take the average path length through the volume of interest V , between all pairs of surfaces. We accordingly have that the (i, j) -th entry of ℓ corresponds to the path lengths between the surfaces ω_i and ω_j . The matrix is furthermore symmetric since this length is not dependent on direction.

The mean speed of the atoms v can be obtained from the Maxwell-Boltzmann distribution and is given by

$$v = \sqrt{\frac{8RT_a}{\pi M}}, \quad (2.11)$$

with R the gas constant in Joules per mole Kelvin, T_a the average temperature of the atom cloud in Kelvin and M moles mass in kilograms per mole.

The output function in (2.5) is accordingly obtained by combining the equations (2.8)-(2.11).

2.3 Methods for obtaining the transfer matrix

We will now describe how to obtain the matrix A , or more specifically, p^A and δ so that $A = \delta p^A$. We can determine the particles transfer matrix p^A analytically by calculating the area integration of (2.1) explicitly, or through an approximation based on numerical integration. The latter can also be realized through sampling/randomization methods. In general, the (i, j) -th element of p^A (which corresponds to the particles transfer between discrete surfaces ω_i and ω_j) is obtained by solving

$$p_{i,j}^A = \int_{\omega_j} \int_{\omega_i} \frac{\cos(\theta_{d\omega_i}) \cos(\theta_{d\omega_j})}{d_E(d\omega_i, d\omega_j)^2 \pi} d\omega_i d\omega_j. \quad (2.12)$$

It follows that p^A is symmetric and we have $\sum_i p_{i,j}^A = 1$, for all j . We remark that for the simple case where a cylinder is considered and is discretized only along the radius or the height, we have an analytic expression of p^A which is presented in (Cale and Raupp 1990).

Instead of analytically solving the Knudsen cosine law between different discrete spaces in a given geometry, we can approximate p^A through a sampling method, e.g.,

Monte-Carlo (MC) simulation, as is commonly used in the literature on such processes. Solving the Knudsen cosine law explicitly yields the most accurate solution. However, obtaining this solution quickly becomes non-trivial for finer discretizations or more complex geometries. On the other hand, approximating through sampling remains fairly straightforward for fine discretizations and this approach allows us to deal with the more complex geometries that we encounter in real applications of, for example, UHVCVD reactor design.

Let us now explain how to set up the MC simulation that can approximate the fraction of particles transferred between two surfaces. Information on the angles between the infinitesimal surfaces $\theta_{d\omega_1}$ and $\theta_{d\omega_2}$ (Cf., Eq. (2.1)) is required for this simulation. Thus, if we consider the illustration in Fig. 2.1, then we can express the fraction of particle transferred (i.e., the solution to the double integral in (2.12)) by the generalized coordinates θ and ϕ . In this case, for setting up the sampling method, we can define the desired density function $\vartheta : [0, \frac{\pi}{2}] \times [0, 2\pi] \rightarrow \mathbb{R}_{\geq 0}$ by

$$\vartheta(\theta, \phi) = \frac{1}{2\pi} \sin(2\theta) \quad (2.13)$$

for all $(\theta, \phi) \in [0, \frac{\pi}{2}] \times [0, 2\pi]$. The steps for obtaining ϑ are included in Appendix 2.A. We can then use MC simulation, with randomness imposed by (2.13), to obtain a collection of generalized coordinate pairs (θ, ϕ) . These pairs can be allocated to discrete-surfaces through accept-reject algorithms based on the coordinates of the discrete-surface boundaries as expressed in (θ, ϕ) with relation to the considered infinitesimal area. However, we need to obtain the fractions of particles that are exchanged between discrete surfaces. Therefore, for the discrete-surface containing the considered infinitesimal area, we take the average over a sampling of infinitesimal areas belonging to this discrete-surface.

The constant δ is a timescale correction for the scattering phenomena. The speed of the dynamics is not directly apparent as it can be dependent on multiple physical factors that are difficult to characterize, such as atomic vibration time and binding energy. We can fit δ empirically when suitable measurements are available (such as the AAS measurement that we will implement in Chapter 3).

2.3.1 Validation of MC simulation based method

In order to verify our MC simulation method, we will make a comparison with (Cale and Raupp 1990). We compare the likeliness of transfer from an infinitesimal point on the side of the cylinder to discrete surface rings on the bottom, that span $\varphi = [0, 2\pi]$ in the angular dimension and various bounds of \bar{r} in the radial dimension. In these simulations, we consider a cylinder with height $\bar{H} = 10$ and radius $\bar{R} = 5$.

For various heights $\bar{h} \in [0, 10]$ of the infinitesimal along the cylinder side, the likeliness of transfer to four discrete rings (satisfying $\bar{r} = [0, 1.25], [1.25, 2.5], [2.5, 3.75]$ and $[3.75, 5]$, respectively, in the radial dimension) on the cylinder bottom is calculated and displayed in Fig. 2.2. We find that the results agree with each other, where

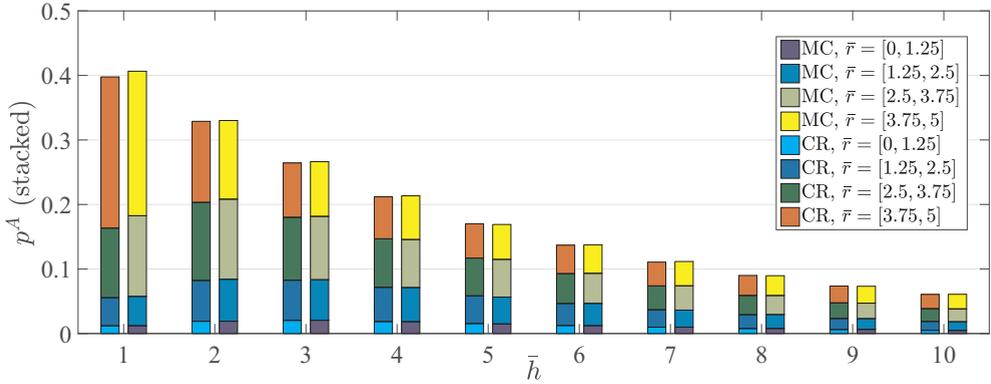


Figure 2.2: Validation of MC based method for obtaining the transfer matrix

Results of the MC simulation in comparison with the analytic integration presented in (Cale and Raupp 1990) are shown. The comparison is made for several infinitesimal points along the side of the cylinder, whose height values \bar{h} are displayed on the x-axis. The probability shown is given for the four discrete surface rings on the bottom of the cylinder, with values of \bar{r} spanning the bounds given by the legend. The figure shows strong similarities between the results from the MC simulation and the analytical solutions.

the most distinct differences are found close to the corner where $\bar{r} = 5$ for the bottom or $\bar{h} = 0$ for the side, which can be considered as a limit case.

2.4 Numerical controller simulations for FMF dynamics

In this section, we discuss two relevant control problems for a FMF process such as we have modeled by (2.5). In order to achieve robustness and guaranteed performance against uncertainties and variations in the process, we propose a point-wise min-norm controller which can take into account desirable particle flux dynamics on relevant surfaces or desired partial pressures. It replaces the current open-loop setting by enabling a feedback controller in the process that ensures that the actual state trajectories evolve in a small tube around the desired trajectory. It is similar to the model-based reference tracking control problem except that we require a guaranteed level of closeness to the desired trajectory. To this end, we will start by introducing our point-wise min-norm controller design. After this is done, we will present implementation of such a controller for flux magnitudes through a numeric simulation. In this simulation, we will consider the dynamical model as is in line with our work in (Dresscher et al. 2017). This example will accordingly consider slightly different dynamics as presented in (2.5), but results are still relevant. We implement the point-wise min-norm controller that is restricted by both a control Lyapunov function and a control barrier function, which provides a guaranteed transient behaviour prop-

erty. We furthermore consider a finely discretized cylinder geometry, having 256 discrete surfaces, through model reduction. The units used in this simulation are arbitrary, e.g. cylinder dimensions and flux magnitudes are not directly realistic. To improve this, we will also present a second numerical example, one that is dedicated to controlling the partial pressures. This partial pressure example uses a more simple controller, with only a control Lyapunov function, causing us to lose the guaranteed transient behaviour property. This is done for simplicity, as the example is primarily meant to (i) bridge the discrepancy between the applied model in (Dresscher et al. 2017) and this thesis and (ii) provide an implementation for a model realization that we have validated for the vapor pressure regime in practice and which is therefore considerably more realistic. This example is accordingly applied to the dynamics that we have presented in Section 2.2 and considers the same cylinder geometry and dynamics that we use (and find) for our experiments in Chapter 3. Both simulations are performed under the assumption that no further sorption occurs, e.g. the boundary is saturated and we are below vapor pressure levels. We furthermore assume that we can directly apply a flux input. These assumptions cause the resulting system to be linear.

2.4.1 Point-wise min-norm controller design

We will now present our control design for stabilization with guaranteed transient behaviour, which we will implement fully in Section 2.4.2 and partially (without guaranteed transient behaviour) in Section 2.4.3. The requirement of stabilization with guaranteed transient behavior shares the same principle as the stabilization with guaranteed safety as proposed in (Romdlony and Jayawardhana 2016b). In order to make this more precise, we let $\mathcal{P} \subset \mathbb{R}^n$ define a closed set around the desired state trajectory x_d , i.e., $x_d(t) \in \mathcal{P}$ for all $t \geq 0$. We also assume that x_d converges to a desired operating point $x^* \in \mathbb{R}^n$. In this case, the problem of stabilization with guaranteed transient behavior for our system in (2.5) can be stated as follows.

Stabilization with guaranteed transient behavior problem: For the system in (2.5) and given the set \mathcal{P} , design a control law $u = k(x)$ such that $x(t) \in \mathcal{P}$ for all t and $x(t) \rightarrow x^*$ as $t \rightarrow \infty$,

where x are the states. If we consider the complement set of \mathcal{P} as the set of unsafe state \mathcal{D} , i.e., $\mathcal{D} = \mathbb{R}^n \setminus \mathcal{P}$, then the above problem can be recast to the stabilization with guaranteed safety problem (Romdlony and Jayawardhana 2016b) as given below.

Stabilization with guaranteed safety problem: For the system in (2.5) and given the set of unsafe state \mathcal{D} , design a control law $u = k(x)$ such that $x(t) \notin \mathcal{D}$ for all t and $x(t) \rightarrow x^*$ as $t \rightarrow \infty$.

In the sequel, we assume the above dual problem when designing the control law for a FMF process. In the current state-of-the-art, there are basically two different

approaches in solving the stabilization with guaranteed safety problem. The first one is based on the construction of control Lyapunov-Barrier function and the usage of Sontag's universal control law as pursued in (Romdlony and Jayawardhana 2014, Romdlony and Jayawardhana 2016b); the other one is based on a point-wise min-norm controller, employing both a control Lyapunov function and a control barrier function (Ames et al. 2014). As proposed in (Romdlony and Jayawardhana 2014, Romdlony and Jayawardhana 2016b), we can construct the control Lyapunov-Barrier function by combining a control Lyapunov function with a control barrier function, which is not trivial. Since it is not trivial to combine explicitly these functions, we will consider, in this paper, the implementation of the latter approach.

Let us now briefly recall the concept of a control barrier function (CBF) and the point-wise min-norm control formulation as proposed in (Ames et al. 2014) using quadratic programming and adopted to our systems' formulation.

Consider a candidate barrier function $B : \mathbb{R}^n \rightarrow \mathbb{R}$ which is C^1 , e.g. continuously differentiable, and satisfies

$$\mathcal{D} = \{x \in \mathbb{R}^n : B(x) \geq 0\}, \quad (2.14)$$

$$\partial\mathcal{D} = \{x \in \mathbb{R}^n : B(x) = 0\} \quad \text{and} \quad (2.15)$$

$$\text{Int}(\mathcal{D}) = \{x \in \mathbb{R}^n : B(x) > 0\}. \quad (2.16)$$

where $\partial\mathcal{D}$ denotes the boundary of \mathcal{D} and $\text{Int}(\mathcal{D})$ denotes the interior of \mathcal{D} .

For describing the control Lyapunov function and control barrier function, we need to introduce a few more notations as follows. For a given general affine nonlinear system

$$\begin{aligned} \dot{x} &= \hat{f}(x) + \hat{g}(x)u \\ y &= \hat{h}(x), \end{aligned} \quad (2.17)$$

with output y and sufficiently smooth functions \hat{f}, \hat{g} and \hat{h} . We denote $\mathcal{L}_{\hat{f}}\mathcal{V}(x)$ as the Lie derivative of a function $\mathcal{V} : \mathbb{R}^n \rightarrow \mathbb{R}$ along the vector field \hat{f} , i.e., $\mathcal{L}_{\hat{f}}\mathcal{V}(x) := \frac{\partial\mathcal{V}(x)}{\partial x}\hat{f}(x)$. Similarly, $\mathcal{L}_{\hat{g}}\mathcal{V}(x) := \frac{\partial\mathcal{V}(x)}{\partial x}\hat{g}(x)$. Furthermore, let $U \subset \mathbb{R}^m$ be the set of admissible inputs and $X \subset \mathbb{R}^n$ the set of admissible states.

Exponentially safe control barrier function (ES-CBF): A C^1 function $B : \mathbb{R}^n \setminus \mathcal{D} \rightarrow \mathbb{R}$ satisfying (2.14)-(2.16) is an ES-CBF if there exist constants $c_1, c_2, c_3, \kappa_1 > 0$ such that

$$-c_1\|x\|_{\partial\mathcal{D}}^2 - \kappa_1 \leq B(x) \leq -c_2\|x\|_{\partial\mathcal{D}}^2 \quad (2.18)$$

$$\inf_{\mu \in U} \left[\mathcal{L}_{\hat{f}}B(x) + \mathcal{L}_{\hat{g}}B(x)\mu + c_3B(x) \right] \leq 0 \quad (2.19)$$

for all $x \in \mathbb{R}^n \setminus \mathcal{D}$, where $\|x\|_{\partial\mathcal{D}}$ denotes the shortest Euclidean distance between the point x and the boundary of \mathcal{D} .

Both conditions of ES-CBF as given above are different to the ones considered in (Ames et al. 2014, Xu et al. 2015), where, in these papers, (2.18) is replaced by

$$\frac{1}{\alpha_1(\|x\|_{\partial\mathcal{D}})} \leq B(x) \leq \frac{1}{\alpha_2(\|x\|_{\partial\mathcal{D}})} \quad (2.20)$$

with \mathcal{K} functions α_1, α_2 and (2.19) is replaced by

$$\inf_{\mu \in U} \left[\mathcal{L}_{\hat{f}}B(x) + \mathcal{L}_{\hat{g}}B(x)\mu - \frac{\kappa_2}{B(x)} \right] \leq 0, \quad (2.21)$$

with $\kappa_2 > 0$. One can see immediately that the conditions (2.20) and (2.21) are strong, in the sense that the corresponding barrier function B has a singularity property at the boundary of \mathcal{D} which may constrain the design of B . The use of (2.18)-(2.19) has also another nice robustness property, namely, input-to-state safety. We refer the interested reader to (Romdlony and Jayawardhana 2016a) for the details on input-to-state safety. Another advantage of assuming (2.18)-(2.19) is that we can readily use many numerical tools for polynomials, such as, sum-of-squares tools, for constructing B .

Similarly, we define an exponential stability control Lyapunov function (ES-CLF) following (Ames et al. 2014).

Exponentially stable control Lyapunov function (ES-CLF): A C^1 function $\mathcal{V} : \mathbb{R}^n \rightarrow \mathbb{R}$ is an ES-CLF if there exist constants $c_1, c_2, c_3 > 0$ such that

$$c_1\|x\|^2 \leq \mathcal{V}(x) \leq c_2\|x\|^2 \quad (2.22)$$

$$\inf_{\mu \in U} [\mathcal{L}_{\hat{f}}\mathcal{V}(x) + \mathcal{L}_{\hat{g}}\mathcal{V}(x)\mu + c_3\mathcal{V}(x)] \leq 0 \quad (2.23)$$

for all $x \in \mathcal{P}$.

For solving the stabilization with guaranteed transient behavior problem w.r.t. the desired set of states \mathcal{P} , we will adopt a point-wise min-norm programming using both ES-CLF $\mathcal{V}(x)$ and ES-CBF $B(x)$. A review on the use of point-wise min-norm controller based on ES-CLF for the stabilization of continuous-time nonlinear systems can be found in (Primbs et al. 1999). We will adopt the point-wise min-norm controller for combining both functions similar to the one pursued in (Ames et al. 2014, Xu et al. 2015).

When we want to combine these two functions, extra care has to be taken since the functionality of ES-CLF and ES-CBF can be conflicting with each other. Indeed, if the inequalities for ES-CBF and ES-CLF hold globally, then the fulfillment of the ES-CBF inequality (c.f., (2.19)) ensures that the distance to \mathcal{D} grows indefinitely due to (2.18), which contradicts to the use of ES-CLF that ensures asymptotic convergence of the state to the origin. These two functions can be combined in a domain outside the neighborhood of the origin. In the neighborhood of the origin, the control should be dictated by the ES-CLF. In view of this, in the following discussion, we

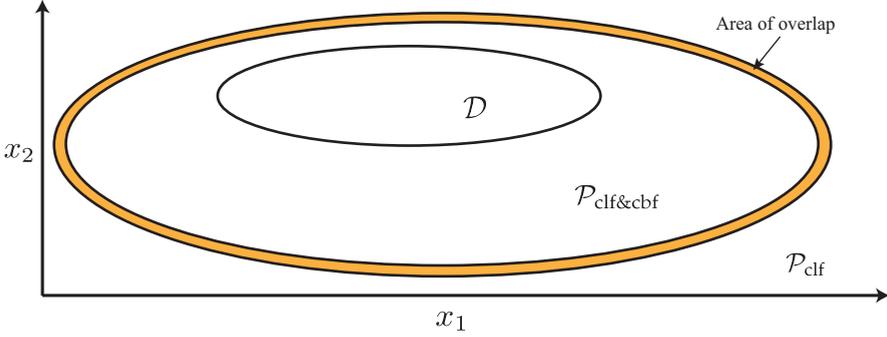


Figure 2.3: Illustration showing the partitioned safe and unsafe sets

We show partitioning of the safe area \mathcal{P} in $\mathcal{P}_{\text{clf}\&\text{cbf}}$ and \mathcal{P}_{clf} , given an unsafe set \mathcal{D} , for a system with two states. The partitioned safe areas overlap in the coloured area. Having the overlap prevents the occurrence of Zeno behavior. Restricting the domain where the ES-CBF is active furthermore helps us to avoid conflict between the ES-CBF and the ES-CLF.

will partition \mathcal{P} into $\mathcal{P}_{\text{clf}\&\text{cbf}}$ and \mathcal{P}_{clf} such that they overlap and their boundaries are not intersecting with each other. We show an example of such a partition in Fig. 2.3.

Point-wise min-norm quadratic program (QP) for stabilization with guaranteed transient behavior.

Let \mathcal{V} be an ES-CLF and B be a ES-CBF where $\mathcal{D} = \mathbb{R}^n \setminus \mathcal{P}$. For given constants $\gamma_{\text{clf}}, \gamma_{\text{cbf}} > 0$ and for all $x \in \mathcal{P}_{\text{clf}\&\text{cbf}}$, we define the ES-CLBF (Control Lyapunov-Barrier Function) control law $u = k_{\text{clf}\&\text{cbf}}(x)$ by the solution of the following QP problem:

$$\text{QP1: } k_{\text{clf}\&\text{cbf}}(x) = \underset{u}{\text{argmin}} \frac{1}{2} u^\top F(x) u + H(x)^\top u \quad (2.24)$$

$$\text{s.t. } \begin{cases} \mathcal{L}_{\hat{f}} \mathcal{V}(x) + \mathcal{L}_{\hat{g}} \mathcal{V}(x) u + \gamma_{\text{clf}} \mathcal{V}(x) \leq 0, \\ \mathcal{L}_{\hat{f}} B(x) + \mathcal{L}_{\hat{g}} B(x) u + \gamma_{\text{cbf}} B(x) \leq 0, \end{cases} \quad (2.25)$$

where $F, H \succeq 0$ are smooth functions such that $\frac{1}{2} u^\top F(x) u + H(x)^\top u$ is convex. Similarly, for all $x \in \mathcal{P}_{\text{clf}}$, we define the ES-CLF control law $u = k_{\text{clf}}(x)$ by the solution of

$$\text{QP2: } k_{\text{clf}}(x) = \underset{u}{\text{argmin}} \frac{1}{2} u^\top F(x) u + H(x)^\top u \quad (2.26)$$

$$\text{s.t. } \mathcal{L}_{\hat{f}} \mathcal{V}(x) + \mathcal{L}_{\hat{g}} \mathcal{V}(x) u + \gamma_{\text{clf}} \mathcal{V}(x) \leq 0. \quad (2.27)$$

Let $K_{\text{clf}}(x)$ define the admissible input set for a given x such that the ES-CLF inequality holds, i.e.,

$$K_{\text{clf}}(x) := \{ \mu \in U \mid \mathcal{L}_{\hat{f}} \mathcal{V}(x) + \mathcal{L}_{\hat{g}} \mathcal{V}(x) \mu + \gamma_{\text{clf}} \mathcal{V}(x) \leq 0 \} \quad (2.28)$$

and similarly, let $K_{\text{cbf}}(x)$ define the admissible input set for a given x such that the ES-CBF inequality holds, i.e.,

$$K_{\text{cbf}}(x) := \{\mu \in U \mid \mathcal{L}_{\hat{f}}B(x) + \mathcal{L}_{\hat{g}}B(x)\mu + \gamma_{\text{cbf}}B(x) \leq 0\}. \quad (2.29)$$

Then, we can see that the first necessary and sufficient condition for the feasibility of QP1 is that

$$K_{\text{clf}}(x) \cap K_{\text{cbf}}(x) \neq \emptyset \quad (2.30)$$

holds for all $x \in \mathcal{P}_{\text{clf}\&\text{cbf}}$.

Another implicit assumption in the feasibility of QP1 is the Artstein's-like condition for the control Lyapunov function and for the control barrier function. In this case, we require that for the ES-CLF \mathcal{V} , there exists $\gamma_{\text{clf}} > 0$ such that

$$\mathcal{L}_{\hat{f}}\mathcal{V}(x) \leq -\gamma_{\text{clf}}\mathcal{V}(x) \quad \forall x \text{ s.t. } \mathcal{L}_{\hat{g}}\mathcal{V}(x) = 0. \quad (2.31)$$

Similarly, for the ES-CBF B , there exists $\gamma_{\text{cbf}} > 0$ such that

$$\mathcal{L}_{\hat{f}}B(x) \leq -\gamma_{\text{cbf}}B(x) \quad \forall x \text{ s.t. } \mathcal{L}_{\hat{g}}B(x) = 0. \quad (2.32)$$

Based on the ES-CLBF and the ES-CLF control laws, we can propose a simple hybrid automaton controller (as considered in (Romdlony and Jayawardhana 2015)) to stabilize the origin with guaranteed transient behavior as follows. We consider two automata where the domains are given by $\mathcal{P}_{\text{clf}\&\text{cbf}}$ and \mathcal{P}_{clf} , while the guards are given by the boundary of $\mathcal{P}_{\text{clf}\&\text{cbf}}$ and \mathcal{P}_{clf} , respectively. Depending on the active automaton, we implement either $u = k_{\text{clf}\&\text{cbf}}(x)$ or $u = k_{\text{clf}}(x)$. The reset map will be identity, i.e., there is no resetting of state variables. The jump only occurs when the boundary of the active domain is reached, in which case, we switch to the other control law. Since we assume that the domains have overlap with no intersecting boundaries, we ensure that there will be a minimum dwell-time that prevents Zeno behavior to occur.

Theorem 1. *The control laws $u = k_{\text{clf}\&\text{cbf}}(x)$ and $u = k_{\text{clf}}(x)$ are locally piecewise Lipschitz continuous and the proposed hybrid automata controller solves the problem of stabilization with guaranteed transient behavior.*

PROOF. The proof of local piecewise Lipschitz continuity follows the same arguments as the proof of Theorem 8 and 11 in (Xu et al. 2015), which is based on the use of Karush-Kuhn-Tucker condition for optimality to the QP problem with convex cost function and nonlinear affine constraints. In this case, the closed-form of the $k_{\text{clf}\&\text{cbf}}(x)$ and $k_{\text{clf}}(x)$ can be expressed as a function of the KKT multipliers and the Lie derivatives of \mathcal{V} and B .

For the closed-loop system, from (2.25) and (2.27), we have that \mathcal{V} satisfies

$$\dot{\mathcal{V}}(x(t)) \leq -\gamma_{\text{clf}}\mathcal{V}(x) \quad (2.33)$$

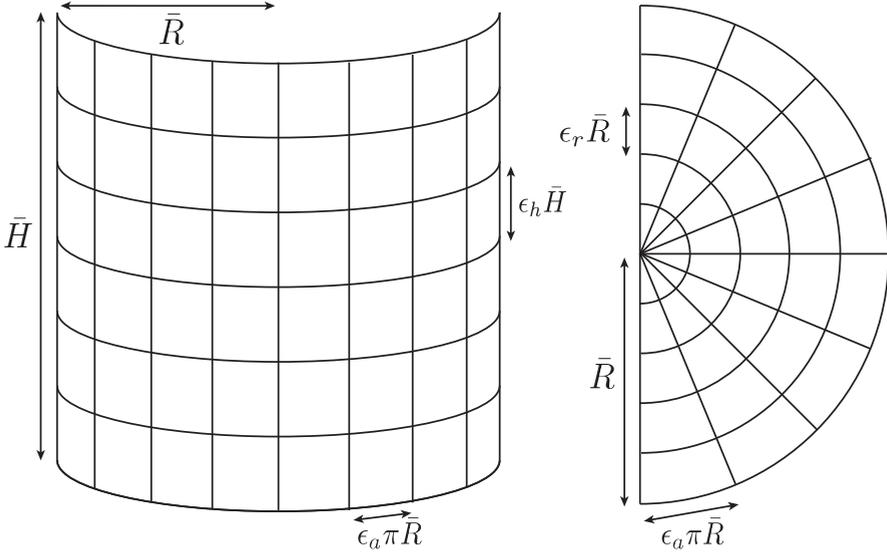


Figure 2.4: Discretization of cylinder for numerical flux control example

For our example geometry of a cylinder, we implement a discretization as shown in this figure. By choosing values for ϵ_a , ϵ_r and ϵ_h , we create a uniform grid in two dimensions on the top, side and bottom of the cylinder.

for all $x(t) \in \mathcal{P}$ in both discrete state of automaton. This implies immediately that \mathcal{V} is a common Lyapunov function that shows the convergence of x to the origin exponentially. On the other hand, when $x(t) \in \mathcal{P}_{\text{clf}\&\text{cbf}}$, we have that B satisfies

$$\dot{B}(x(t)) \leq -\gamma_{\text{cbf}} B(x) \quad (2.34)$$

which, together with the lower and upper bound of B , implies that x will never enter \mathcal{D} . Therefore, we achieve stabilization with guaranteed transient behavior. \square

2.4.2 Numerical flux control example

Let us now present a numerical simulation of a model similar to (2.5), as presented in (Dresscher et al. 2017), and the point-wise min-norm controller for flux control with guaranteed safety. We consider a cylindrical reaction chamber with height $\bar{H} = 10$ and radius $\bar{R} = 5$. We will first describe the spatial discretization of our cylinder example as shown in Fig. 2.4. We take $\epsilon_a = \frac{1}{8}$, $\epsilon_r = \frac{1}{4}$ and $\epsilon_h = \frac{1}{8}$. Accordingly, we define lines along the surfaces of the cylinder, such that these lines act as boundaries for the discrete surfaces. In cylindrical coordinates, the lines span $([0, \bar{R}], 0, i_1 \epsilon_a \pi)$, $([0, \bar{R}], \bar{H}, i_1 \epsilon_a \pi)$ and $(\bar{R}, [0, \bar{H}], i_1 \epsilon_a \pi)$, where $i_1 = 0, 1, \dots, 2\epsilon_a^{-1}$, for the cylinder bottom, top and side, respectively. The lines in the height dimension on the side of the

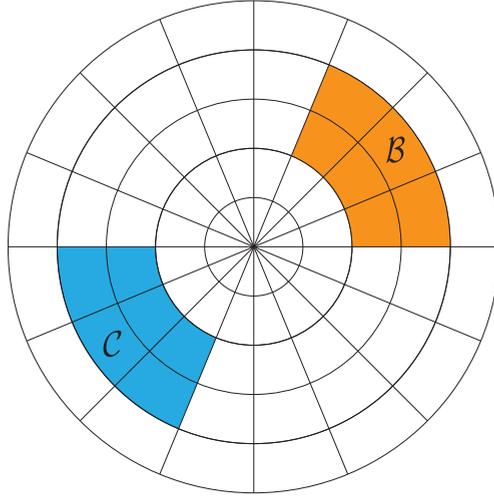


Figure 2.5: Allocation of inputs and outputs for flux controller

We show the allocation of the \mathcal{B} and \mathcal{C} vectors with respect to the system as in (2.5). The surface shows the discretized bottom of the cylinder for our example geometry. For our numerical example, vectors $\mathcal{B} \in \mathbb{R}^{256}$ and $\mathcal{C} \in \mathbb{R}^{256}$ both have values of $\frac{1}{6}$ for the six colored surfaces they are associated to and are 0 for all other surfaces.

cylinder then span $(\bar{R}, i_2 \epsilon_h \bar{H}, [0, 2\pi])$, where $i_2 = 0, 1, \dots, \epsilon_h^{-1}$. The lines in the radial dimension on the bottom and top span $(i_3 \epsilon_r \bar{R}, 0, [0, 2\pi])$ and $(i_3 \epsilon_r \bar{R}, \bar{H}, [0, 2\pi])$, respectively, where $i_3 = 0, 1, \dots, \epsilon_r^{-1}$. The number of discrete surface elements is then given by $n = 2\epsilon_h^{-1}\epsilon_a^{-1} + 4\epsilon_r^{-1}\epsilon_a^{-1} = 256$.

As a first step, we compute the matrix A . We use Monte-Carlo simulations to obtain transfer fractions between discrete surfaces, which form matrix p^A , with the densities of departing angles as in (2.13). We use the discrete surface boundaries, expressed in the θ and ϕ coordinates with respect to the considered infinitesimal as accept-reject conditions. As a next step, we relate an infinitesimal to a discrete area by taking the average transfer fractions of sampled infinitesimal areas belonging to that surface. We have then found the surface-to-surface transfer matrix p^A , corresponding to (2.12), and are left with multiplying with δ to obtain A . For this example, we let $\delta = 1$ and $p^A = A$.

Let us shortly describe the remaining system dynamics as in (2.5) that we consider for this example. For the leakage matrix L , we consider $L = -\frac{0.25}{256} \mathbf{1}^{256 \times 256}$, where $\mathbf{1}^{256 \times 256}$ is a 256×256 matrix of ones. We will furthermore operate under a saturated boundary assumption, e.g. no further sticking occurs and $f = 0$. As this example was originally published in (Dresscher et al. 2017), with a model slightly different from (2.5), we have that $(A - L)g(u(t), t) = \mathcal{B}u(t)$ and $h(x(t), T_a(t)) = \mathcal{C}x(t)$. These differences have only a small impact on the dynamics. For the output, we now

assume that we have a measurement of fluxes on a specific surface, instead of partial pressure. For \mathcal{B} and \mathcal{C} , we then assign a value of $\frac{1}{6}$ to six discrete space elements on the bottom of the cylinder, as shown and explained in Fig. 2.5.

Controller design

For facilitating the control design where we use a quadratic program formulation, we first need to obtain a reduced-order model. To this end, we apply balanced truncation (Antoulas 2005). Through analysis of the Hankel singular values, we reduce the vector of states x from \mathbb{R}^{256} as in (2.5) to \mathbb{R}^2 . The corresponding reduced system significantly deviates in high frequencies from the original system as is common with this method. However, we obtain a good maximum \mathcal{H}_∞ norm difference between the two systems of 1.9900×10^{-4} . We will use this reduced system for the controller design, where the controller is subsequently applied to the high-order system. A different reduction method can be considered if the controller design is inadequate for the high-order system.

Our goal is to converge to an equilibrium state vector x^* in such a way that we have guaranteed transient behavior, namely $x \in \mathcal{P}$. To realize this, we will incorporate u as a state s.t. $\hat{u} := \dot{u} \in \mathbb{R}$, where \hat{u} is a new input. By doing this, we obtain a relative degree of 2 which will enable us to design an ES-CBF for shaping both y and \dot{y} as will be discussed later. Hence, with reference to (2.17), the resulting reduced order dynamics with $x \in \mathbb{R}^3$ can be expressed in the canonical observable form as

$$\begin{aligned} \dot{x} &= \hat{f}(x) + \hat{g}(x)\hat{u} = \hat{A}x + \hat{B}\hat{u} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & \hat{a}_1 \\ 0 & 1 & \hat{a}_2 \end{bmatrix} x + \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ 0 \end{bmatrix} \hat{u}, \\ y &= \hat{h}(x) = \hat{C}x = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x, \end{aligned} \quad (2.35)$$

with $\hat{a}_1 = -0.1051$, $\hat{a}_2 = -0.5334$, $\hat{b}_1 = 5.5067 \times 10^{-5}$ and $\hat{b}_2 = 3.6872 \times 10^{-4}$. Furthermore, we consider the desired equilibrium as $x^* = \begin{bmatrix} 105 & 533 & 1000 \end{bmatrix}^\top$.

Subsequently, to facilitate the implementation of the point-wise min-norm control with guaranteed transient behavior for the reduced-order model in (2.35) we consider our candidate ES-CLF as follows

$$\mathcal{V}(x - x^*) = \frac{1}{2}(x - x^*)^\top (x - x^*), \quad (2.36)$$

such that the corresponding constraint to the ES-CLF in (2.25) becomes

$$\frac{\partial \mathcal{V}(x - x^*)}{\partial x} (\hat{A}x + \hat{B}\hat{u}) \leq -\gamma_{\text{clf}} \mathcal{V}(x - x^*). \quad (2.37)$$

Furthermore, we consider an ES-CBF candidate which will effectively prevent the measured output $y = x_3$ from growing rapidly by providing an upper limit for

$\dot{y} = x_2 + \hat{a}_2 x_3$. This ES-CBF candidate is then described by

$$B(x) = -\frac{(x_3 - 100)^2}{600} - (x_2 + \hat{a}_2 x_3 - 115)^2 + 100^2, \quad (2.38)$$

in terms of x_2 and x_3 , such that (2.38) contains the set of the unsafe states \mathcal{D} and has the desired effect of providing a maximum for \dot{y} . Hence, the corresponding constraint to the ES-CBF in (2.25) yields

$$\frac{\partial B(x)}{\partial x}(\hat{A}x + \hat{B}\hat{u}) \leq -\gamma_{\text{cbf}}B(x) \quad (2.39)$$

and is providing the desired transient behavior guarantees.

Let us now define the two domains and guards for the two automata. For the ES-CLBF automaton, we choose $\mathcal{P}_{\text{clf}\&\text{cbf}} = \{x \mid \dot{y} > 8\} = \{x_2, x_3 \mid x_2 + \hat{a}_2 x_3 > 8\}$ with guard condition $\dot{y} \leq 8$. For the ES-CLF automaton, we choose $\mathcal{P}_{\text{clf}} = \{x \mid \dot{y} < 9\}$ with guard condition $\dot{y} \geq 9$. Consequently, combining (2.37) and (2.39), we obtain the ES-CLBF for the stabilizing flux controller with guaranteed transient behavior for the reduced order FMF process in (2.35) by solving the realization of (2.24) - (2.27) for this example, given by

$$\text{QP1e: } k_{\text{clf}\&\text{cbf}}(x) = \arg \min_{\hat{u}} \hat{u}^\top F(x)\hat{u} + H(x)^\top \hat{u} \quad (2.40)$$

$$\text{s.t. } \begin{cases} \frac{\partial \mathcal{V}(x-x^*)}{\partial x}(\hat{A}x + \hat{B}\hat{u}) \leq -\gamma_{\text{clf}}\mathcal{V}(x-x^*), \\ \frac{\partial B(x)}{\partial x}(\hat{A}x + \hat{B}\hat{u}) \leq -\gamma_{\text{cbf}}B(x), \end{cases} \quad (2.41)$$

for the ES-CLBF automaton and

$$\text{QP2e: } k_{\text{clf}}(x) = \arg \min_{\hat{u}} \hat{u}^\top F(x)\hat{u} + H(x)^\top \hat{u} \quad (2.42)$$

$$\text{s.t. } \frac{\partial \mathcal{V}(x-x^*)}{\partial x}(\hat{A}x + \hat{B}\hat{u}) \leq -\gamma_{\text{clf}}\mathcal{V}(x-x^*), \quad (2.43)$$

for the ES-CLF automaton. Furthermore, we let $\gamma_{\text{clf}} = 0.03$ and $\gamma_{\text{cbf}} = 100$, $F(x) = 1$, and $H(x)$ be a convex combination of the ES-CLF and the ES-CBF, such that the trajectories avoid the boundaries. The simulation results are presented in the sequel.

Simulation results and discussion

The performance of the controlled system can be observed in the y - \dot{y} plane shown in Fig. 2.6 and in the output trajectory depicted in Fig. 2.7. We have compared the performance of a controller that utilizes solely the ES-CLF control law with a controller that utilizes both the ES-CLF and the ES-CLBF control laws, for QP1e (2.40)-(2.41) and QP2e (2.42)-(2.43). While the former achieves the stabilization goals, the latter realizes better convergence speed performance. The ES-CLF alone could not achieve a similar convergence speed without becoming unsafe. Furthermore, the ES-CBF will always prevent the system trajectory from becoming unsafe, since it guarantees that the solution of QP1e in (2.40)-(2.41) becomes infeasible due to violation of condition (2.30) if a trajectory should enter the unsafe set.

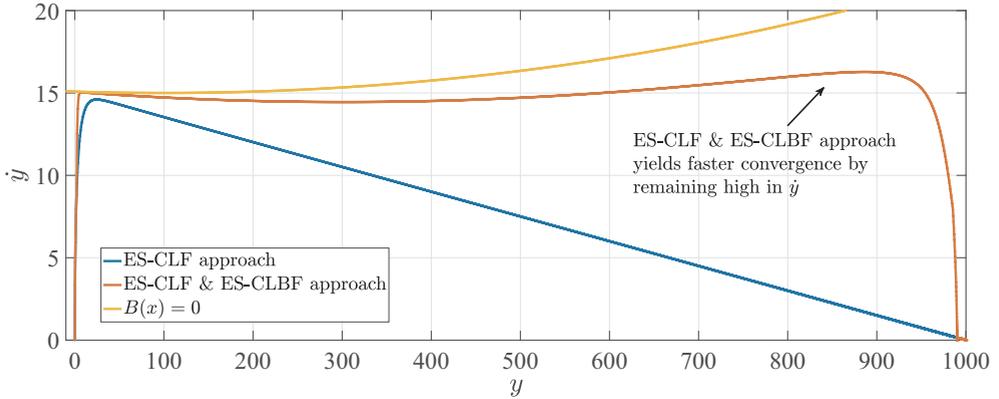


Figure 2.6: Phase plot of numerical flux control example

We show the controlled system performance in terms of y ($= x_3$) and \dot{y} ($= x_2 + \hat{a}_2 x_3$), for QP1e and QP2e given by (2.40)-(2.43). For the ES-CLF approach, the only control law used is the ES-CLF. For the ES-CLF & ES-CLBF approach, both the ES-CLF and the ES-CLBF control laws are used. The line $B(x) = 0$ is the plot of the barrier function B (equal to zero).

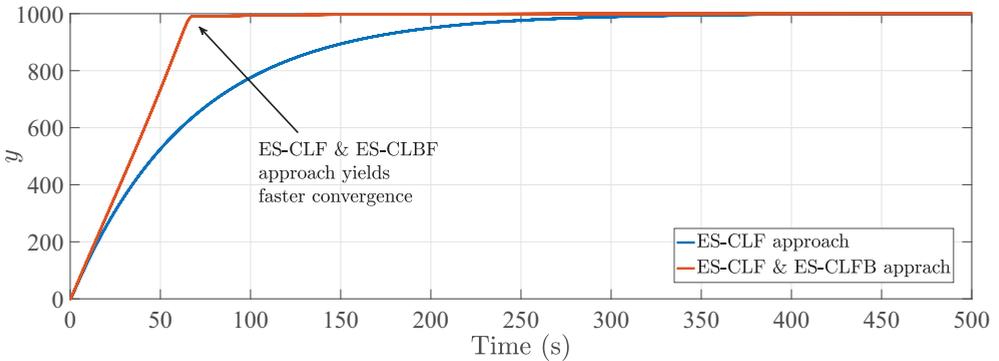


Figure 2.7: Time trajectory of numerical flux control example

The controlled system performance in terms of $y(t)$ for QP1e and QP2e given by (2.40)-(2.43) is shown. For the ES-CLF approach, the only control law used is the ES-CLF. For the ES-CLF & ES-CLBF approach, both the ES-CLF and the ES-CLBF control laws are used.

2.4.3 Numerical partial pressure control example

For our second example in this chapter, we will consider the system dynamics that are in accordance with the observations that we make in Chapter 3. How these dynamics are obtained will be explained extensively there. We hence consider a cylinder with height $\bar{H} = 0.102$ and radius $\bar{R} = 0.062$, in meters, having a volume $V = 1.23 \times 10^{-3} \text{m}^3$. We discretize this cylinder in four surfaces, where ω_1 is the cylinder bottom, ω_2 is the lower half of the side of the cylinder, ω_3 is the upper half

of the side and ω_4 is the top of the cylinder. By solving the integrals provided in (Cale and Raupp 1990), we find

$$p^A = \begin{bmatrix} 0 & 0.1373 & 0.3350 & 0.5277 \\ 0.1373 & 0.3300 & 0.1977 & 0.3350 \\ 0.3350 & 0.1977 & 0.3300 & 0.1373 \\ 0.5277 & 0.3350 & 0.1373 & 0 \end{bmatrix}. \quad (2.44)$$

For the other components in (2.5), we have the universal gas constant $R = 8.3145 \frac{\text{J}}{\text{mol K}}$, the molar mass of sodium $M = 0.0230 \frac{\text{kg}}{\text{mol}}$, $\delta = 0.8112$, $A = \delta p^A$ and

$$\ell = \begin{bmatrix} 0 & 0 & 0.025 & 0.05 \\ 0 & 0 & 0.05 & 0.075 \\ 0.025 & 0.05 & 0.1 & 0.1 \\ 0.05 & 0.075 & 0.1 & 0.1 \end{bmatrix}. \quad (2.45)$$

We are left with assigning f, g, L and T_a . Since we still assume a saturated boundary, we take $f = 0$ as before. Let g then arrange the allocation of incoming flux to the bottom surface, e.g. $g(x)u = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top u$. For L , we will consider leakage which can correspond to a small hole of 6mm in the top of the cylinder. We then have the approximated fraction of leakage given by the surface of the hole divided by the total surface of the top of the cylinder, which is subsequently multiplied by δ . This results in

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0019 \end{bmatrix}, \quad (2.46)$$

with the leakage only applying to the flux on the surface ω_4 at any given time. Lastly, we will assume that $T_a = 473.15$ degrees Kelvin is constant.

We accordingly obtain the linear system dynamics described by

$$\dot{x} = \hat{f}(x) + \hat{g}(x)u = \hat{A}x + \hat{B}u = (A - \delta I - L)x + (A - L) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u, \quad (2.47)$$

$$y = \hat{h}(x) = \hat{C}x = \frac{R}{V} \sqrt{\frac{T_a(t)\pi M}{8R}} \mathbf{1}^\top (p^A \cdot \ell)x, \quad (2.48)$$

where u is input flux in $\frac{\text{mol}}{\text{s}}$ and y is pressure in Pascal.

Controller design

Our goal is now to converge to an equilibrium output y^* . To facilitate this goal, we will implement a point-wise min-norm control for the model in (2.47)-(2.48). For this

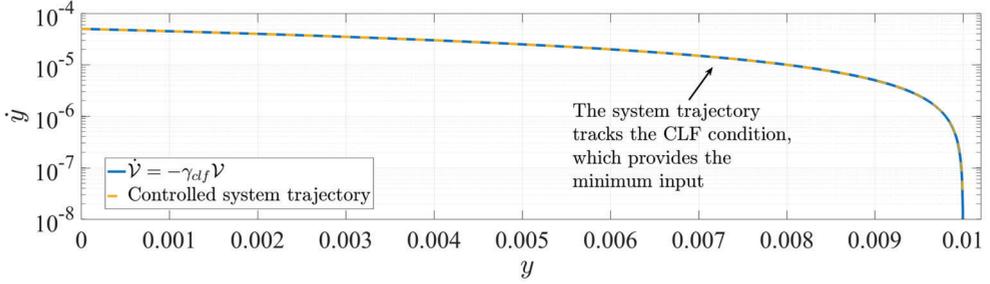


Figure 2.8: Phase plot of numerical partial pressure control example

We show the phase plot for our partial pressure control example in Section 2.4.3. The trajectory follows the constraint imposed by our ES-CLF. This is the optimal trajectory for our quadratic program, which aims to minimize the input at all time while converging to $y^* = 0.01$ Pascal.

purpose, we consider a candidate ES-CLF as

$$\mathcal{V}(y - y^*) = \frac{1}{2}(y - y^*)^2, \quad (2.49)$$

which yields the constraint to the ES-CLF as

$$\frac{\partial \mathcal{V}(y - y^*)}{\partial y} \frac{\partial y}{\partial x} (\hat{A}x + \hat{B}u) \leq -\gamma_{\text{clf}} \mathcal{V}(y - y^*), \quad (2.50)$$

or equivalently

$$\hat{C}(\hat{A}x + \hat{B}u) \leq -\frac{1}{2}\gamma_{\text{clf}}(\hat{C}x - y^*). \quad (2.51)$$

We accordingly obtain the quadratic program as

$$\text{QP: } k_{\text{clf}}(x) = \arg \min_u u^\top F(x)u + H(x)^\top u \quad (2.52)$$

$$\text{s.t. } \hat{C}(\hat{A}x + \hat{B}u) \leq -\frac{1}{2}\gamma_{\text{clf}}(\hat{C}x - y^*), \quad (2.53)$$

with $F(x)$ very large and $H(x) = 0$.

Simulation results and discussion

The performance of the controlled system in the $y - \dot{y}$ plane is shown in Fig. 2.8 and the input and output trajectories are shown in Fig. 2.9. The results show that the proposed point-wise min-norm control law can effectively steer the system to the desired equilibrium, where $y^* = 0.01$ Pascal. More importantly, we show that our modeling framework is suitable for controller design under saturated boundary assumption and with direct input flux actuation for the realistic process dynamics that we verify in the next chapter.

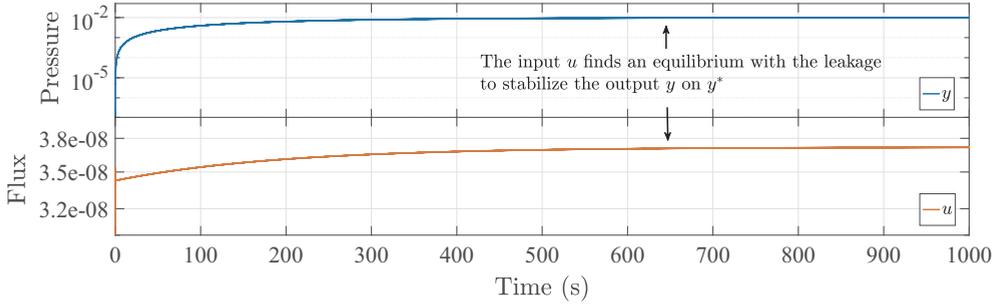


Figure 2.9: Time trajectory of numerical partial pressure control example

The input and output signals for our example in Section 2.4.3 is shown. The system shows an appropriate convergence to the set-point $y^* = 0.01$ Pascal. At this set-point, a constant input is required to maintain equilibrium under leakage conditions.

2.5 Concluding remarks

In this chapter, we have introduced the dynamics of gas flow in the FMF regime and derived a flux-based dynamical model. We have discussed the various model components extensively and have validated a key model component using results from the literature. Lastly, we have numerically implemented a state-of-the-art controller design to show the suitability of our flux-based dynamical model for control purposes.

The flux-based model framework that we have introduced provides a starting point for model identification and model-based controller design. The framework is flexible and can potentially be applied to various types of chemical processes that contain FMF dynamics. The numerical analysis performed in this chapter indicate that such a model is suitable for controller design, but the simulations are performed for special cases and practical application can yield different results.

The next step in the development of this model is to test its predictive capacity experimentally. We will do so in the next chapter.

2.A Derivation of the scattering pdf

This appendix contains the derivation of the probability density function λ , given in (2.13). Starting from (2.1), let us consider a (half) unit sphere. We accordingly obtain $\cos(\theta_{d\omega_2}) = 1$ and $d_E(d\omega_1, d\omega_2) = 1$, which causes the surface $d\omega_2$ to become equal to the solid angle used in (Knudsen 1915). We accordingly have

$$p_{d\omega_1}(d\omega_2) = \frac{\cos(\theta_{d\omega_1})}{\pi} d\omega_2, \quad (2.54)$$

where $d\omega_2$ can be interpreted as a solid angle. For this solid angle, and with reference to Fig. 2.1, we then have

$$d\omega_2 = \sin(\theta_{d\omega_1}) d\theta_{d\omega_1} d\phi_{d\omega_1}, \quad (2.55)$$

and substitution yields

$$p_{d\omega_1}(d\omega_2) = \frac{1}{\pi} \cos(\theta_{d\omega_1}) \sin(\theta_{d\omega_1}) d\theta_{d\omega_1} d\phi_{d\omega_1} = \frac{1}{2\pi} \sin(2\theta_{d\omega_1}) d\theta_{d\omega_1} d\phi_{d\omega_1}, \quad (2.56)$$

which satisfies

$$\int_0^{2\pi} \int_0^{\frac{\pi}{2}} \frac{1}{2\pi} \sin(2\theta_{d\omega_1}) d\theta_{d\omega_1} d\phi_{d\omega_1} = 1. \quad (2.57)$$

We accordingly obtain

$$\vartheta(\theta_{d\omega_1}, \phi_{d\omega_1}) = \frac{1}{2\pi} \sin(2\theta_{d\omega_1}). \quad (2.58)$$

Furthermore, we have that the probability of transfer to a horizontal cut of the unit sphere is given by

$$\vartheta_\theta(\theta_{d\omega_1}) = \int_0^{2\pi} \frac{1}{2\pi} \sin(2\theta_{d\omega_1}) d\phi_{d\omega_1} = \sin(2\theta_{d\omega_1}), \quad (2.59)$$

and for the angular cut we have

$$\vartheta_\phi(\phi_{d\omega_1}) = \int_0^{\frac{\pi}{2}} \frac{1}{2\pi} \sin(2\theta_{d\omega_1}) d\theta_{d\omega_1} = \frac{1}{2\pi} \quad (2.60)$$

Integrating yields the CDFs

$$\Theta_\theta(\theta_{d\omega_1}) = -\frac{1}{2} \cos(2\theta_{d\omega_1}) + \frac{1}{2}, \quad (2.61)$$

where we obtain the last part due to the required CDF property $\Theta_\theta : [0, \frac{\pi}{2}] \rightarrow [0, 1]$, and

$$\Theta_\phi(\phi_{d\omega_1}) = \frac{\phi_{d\omega_1}}{2\pi}, \quad (2.62)$$

which satisfies $\Theta_\phi : [0, 2\pi] \rightarrow [0, 1]$. Inverting yields the sampling distribution for the Monte-Carlo algorithm.

