Andreas Spitaler, BSc

Modeling of Distributed Parameter Systems using Dynamic Mode Decomposition

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Electrical engineering

submitted to

Graz University of Technology

Supervisor

Dr.techn. Stefan Koch

Co-Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Markus Reichhartinger

Institute of Automation and Control
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Martin Horn

Graz, October 2022

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

Due to high complexity modeling and control of spatially distributed systems described by partial differential equations is a challenging task. Formulating system equations based on physical properties is a standard modeling approach in this area. This thesis focuses on a data driven method for system modeling. So-called Dynamic Mode Decomposition (DMD) is applied first to a simple example based on the one dimensional heat equation. The impact of model order reduction on the model error is investigated. Then a laboratory setting consisting of multiple heaters, a thermal camera and a silicon wafer piece is modeled following the data driven approach. The gained model is used for designing a state controller and a model predictive controller, both for tracking purposes. Simulating and testing the controllers on the laboratory setting results into promising behaviours of the closed loop systems.

Key words. dynamic mode decomposition, system identification, model order reduction, thermal diffusion

Aufgrund der hohen Komplexität stellt die Modellbildung und Regelung verteilt-parametrischer Systeme eine große Herausforderung dar. Typischerweise wird das zu untersuchende System durch physikalische Gleichungen mathematisch beschrieben. In dieser Arbeit wird jedoch eine datenbasierte Methode, die sogenannte Dynamic Mode Decomposition, zur Modellierung herangezogen. Das Konzept wird Anfangs auf ein einfaches Beispiel, basierend auf der eindimensionalen Wärmeleitungs-Gleichung, angewendet. Es wird untersucht, welche Auswirkung die im Ansatz integrierte Modellordnungsreduktion auf den Modellfehler hat. Weiters wird ein Labor-Aufbau, welcher zur Untersuchung der Wärmeausbreitung in einem

Silizium Wafer verwendet wird, modelliert und geregelt. Dieser Aufbau stellt ein Mehrgrößensystem dar. Zur Temperaturregelung wird ein Zustandsregler sowie ein modellprädiktiver Regler entworfen und erprobt. Ziel beider Entwürfe ist es ein vorgegebenes Temperaturprofil entlang der Wafer-Oberfläche zu erreichen. Numerische Simulationen sowie Experimente am Labor-Aufbau liefern vielversprechende Ergebnisse.

Contents

List of Figures

1 Introduction

Topic and context. This thesis deals with data driven modeling and control of systems which dynamical behaviour is governed by partial differential equations (PDEs). A common way of modeling „simple“ systems governed by PDEs is e.g. to discretize the PDE in space and time. Often a linear time-invariant (LTI) system of coupled difference equations results by using diffenence quotients and simplifying nonlinear terms. Another possibility to model more complex distributed parameter systems is the Ritz-based finite element method [14]. Due to fine spatial discretization the resulting systems are usually of high order. Thus, the gained models are computional expensive due to the spatial dependence of the variables of interest. In order to approach model-based controller design models of low order are of interest. A concept overcoming this issue in combination with a method for system order reduction is *Dynamic Mode Decomposition* (DMD) [8]. Another advantage of DMD in modeling complex systems is its straightforward usage. There is no need to formulate differential or difference equations describing the dynamics. A meaningful record of an identification experiment describing the process by the contained input and output data is all that is needed approaching DMD. This thesis makes use of this approach for modeling systems governed by the heat diffusion equation. The dynamical behaviour of a laboratory setting is modeled by applying DMD to a data-set of an identification experiment. The gained model is further used to design controllers. The DMD also provides spatio-temporal coherent structures also known as *modes* dominating the investigated process. The graphical representation of the modes illustrates the strong connection of DMD to Fourier analysis. Due to the mode-based model, choosing the weighting matrix of a linear quadratic regulator or a model predictive control approach may be more complex compared to other models. This results from projecting the state vector into the linear subspace where the controller is acting. Also

the identification experiment needs to be chosen properly in order to get a representative model.

Relevance and importance. This research work contributes to the work of J. Proctor, S. Brunton and N. Kutz [12] by applying the proposed concepts in several examples, experiments and simulations. The work of Qiugang Lu and Victor M. Zavala [10] implementing a model predictive control on a 2D array of heating elements as well as the concept of foreground/background seperation of J. Grosek and N. Kutz [7] are taken on this thesis.

Overview of the structure. This thesis is divided into three main parts. The first part introduces the DMD concept. Here the mathematical background used in the following chapters is discussed and a simple example is given. In the second part an existing laboratory setup is modeled first as a single input system in order to check the performance of the data-based modeling algorithm. The chapter continues with the identification and simulation of the setting using multiple inputs. The next chapter deals with the simulation as well as experimental testing of model-based controllers designed using the obtained system model.

2 Introduction into Dynamic Mode Decomposition

Dynamic mode decomposition is a purely data-driven method for system modeling. It provides a „best fit“ discrete linear time-invariant model of the underlying process. The model consists of coupled spatial temporal modes. Also nonlinear systems can be analyzed using DMD. In this case the result will be a linear approximation. It is also possible to model time-varying systems with DMD by applying it on-the-fly [11]. However, this thesis focuses on offline applications by using data-sets of experiments to gain models for model-based controller design.

DMD is often used to analyze high-dimensional systems like in the field of fluid dynamics. More general, DMD is a powerful method regarding spatially distributed processes based on partial differential equations that can be captured with, e.g., a video camera. The pixel of the gained video record are considered as states which may have different values at every snapshot of the record. Since a video usually contains thousands or millions of pixels the system order is high. Here a crucial benefit of the DMD comes into play. The system order can be reduced dramatically by truncating the model obtained by DMD exploiting singular value decomposition. The loss of information that goes along with the truncation can often be kept small despite the drastic order reduction.

Since DMD is purely data-driven there is no need to formulate physically motivated differential equations describing the system dynamics. Nevertheless, it is cruical to perform well chosen experiments on the system to be identified in order to get meaningful data for the DMD and to gain a useful

state-space representation of the underlying process. This is necessary because the resulting model is only valid within the range of the data-set used for the DMD [8]. Performing validation experiments gives an idea of how the model behaves outside the data-set of the identification experiment.

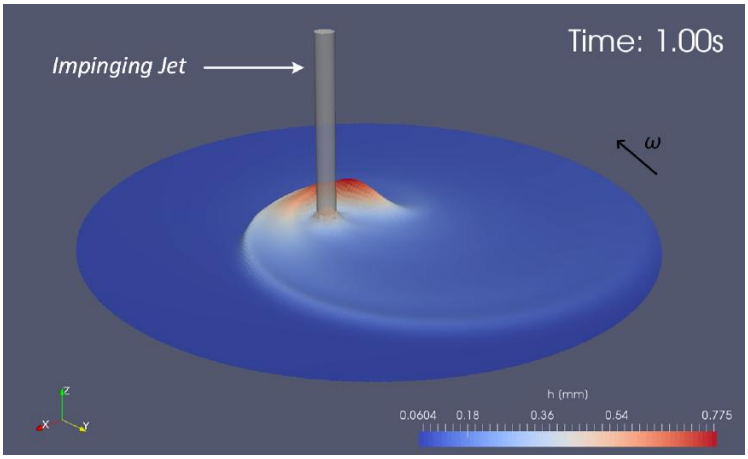


Figure 2.1: Thin film fluid distribution on a silicon wafer ([13])

Another requirement for the DMD analysis is that all states of the system need to be measured over the entire duration of the identification experiment. This condition may be fulfilled by capturing, e.g., a thermal process with a camera since the pixel represent the states in this case.

A motivating example is depicted in Figure 2.1. One could think of modeling how a fluid distributes on a rotating silicon wafer [13] by processing recorded video data of this setting using the DMD. A more extensive introduction to the DMD framework is provided by [8].

2.1 DMD Algorithms

There exist a variety of DMD algorithms. Some examples are the Least Squares DMD [11], the Optimized DMD [2] and the Extended DMD [9].

This thesis focuses on the use of the basic DMD and DMDc whereas DMD is the classic Dynamic Mode Decomposition performing system identification on autonomous systems. DMDc means Dynamic Mode Decomposition with Control and provides a method to identify systems with known input signals [12]. All DMD algorithms use snapshots of the process to be analyzed. Thus the result of the DMD will be a discrete-time state-space model.

2.1.1 DMD - Identification of Autonomous Systems

DMD is based on an investigation of measurement pairs of a discrete dynamical system. It is assumed that

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k \tag{2.1}$$

approximately captures the process dynamics. Here $\boldsymbol{x}_k \in \mathbb{R}^n$ is a measurement vector capturing the states of the discrete dynamical system at every time step k. The order of the system results from the spatial discretization e.g. into pixel using a camera and is denoted by n. The matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ describes how $\boldsymbol{x}$ is transfered from one time step to the next. In general, this can not hold for all pairs of measurements exactly. Especially if a process contains nonlinearities. So the DMD aims to find a best-fit solution for the considered dynamics.

Snapshots of an identification experiment (e.g. a video record) containing all states captured at equidistant time intervals Δt are put into column vectors $\boldsymbol{x}_i$ whereas $i = 1, 2, ..., m$ are the points in time up to the end of the record m refering to each snapshot. Then the matrices

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & ... & \boldsymbol{x}_{m-1} \end{bmatrix}$$

and

$$\boldsymbol{X'} = \begin{bmatrix} \boldsymbol{x}_2 & \boldsymbol{x}_3 & ... & \boldsymbol{x}_m \end{bmatrix}$$

are assembled. The relation between the matrices $\boldsymbol{X}$ and $\boldsymbol{X'}$ can be expressed by

$$\boldsymbol{X'} \approx \boldsymbol{A}\boldsymbol{X}.$$

The reformulation of this relation defines the linear operator

$$\boldsymbol{A} = \boldsymbol{X'}\boldsymbol{X}^{+},$$

where $\boldsymbol{X}^{+}$ stands for the pseudo inverse of $\boldsymbol{X}$. In the next step a truncated Singular Value Decomposition (SVD) is carried out by

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{*} = \begin{bmatrix}\tilde{\boldsymbol{U}} & \tilde{\boldsymbol{U}}_{rem}\end{bmatrix} \begin{bmatrix}\tilde{\boldsymbol{\Sigma}} & 0 \\ 0 & \tilde{\boldsymbol{\Sigma}}_{rem}\end{bmatrix} \begin{bmatrix}\tilde{\boldsymbol{V}}^{*} \\ \tilde{\boldsymbol{V}}^{*}_{rem}\end{bmatrix} \approx \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^{*},$$

where $\tilde{\boldsymbol{U}} \in \mathbb{R}^{n \times r}$ includes all column vectors of $\boldsymbol{U}$ up to the chosen truncation value r. This truncation value provides a degree of freedom in the DMD. A scheme for choosing r can be found in [6]. The matrix $\tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{r \times r}$ is a diagonal matrix with the dominant r singular values of $\boldsymbol{\Sigma}$. The matrix $\tilde{\boldsymbol{V}} \in \mathbb{R}^{n \times r}$ consists of the first r columns of $\boldsymbol{V}$. An approximation $\bar{\boldsymbol{A}} \in \mathbb{R}^{n \times n}$ of the original $\boldsymbol{A}$ matrix can now be computed by using the truncated SVD of $\boldsymbol{X}$ by

$$\boldsymbol{A} \approx \bar{\boldsymbol{A}} = \boldsymbol{X'}\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}^{*}.$$

Expressing $\boldsymbol{A}$ in (2.1) by its approximation leads to

$$\boldsymbol{x}_{k+1} = \boldsymbol{X'}\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}^{*}\boldsymbol{x}_{k}.$$

In order to gain a computationally more efficient model, the state vector $\boldsymbol{x}$ is projected onto a linear subspace by performing the linear transformation

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{U}}^{*}\boldsymbol{x}. \tag{2.2}$$

The measurement vector $\boldsymbol{x}$ can be approximated by

$$\boldsymbol{x} \approx \hat{\boldsymbol{x}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{x}}. \tag{2.3}$$

In (2.1) the measurement vector can be expressed by (2.2). The fact that $\tilde{\boldsymbol{U}}^*\tilde{\boldsymbol{U}} = \boldsymbol{I}$ for $n > r$ is used to simplify the result leading to

$$\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{U}}^*\boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}_k.$$

Reformulating this expression using

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}^*\boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1} \tag{2.4}$$

leads to

$$\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k. \tag{2.5}$$

This means that $\tilde{\boldsymbol{A}}$ is the dynamic matrix of the reduced order system. Relation (2.4) completes the data-driven system modeling. The dynamic modes ϕ of a system are spatial temporal coherent structures dominating the dynamics of the process. The modes are the eigenvectors of $\bar{\boldsymbol{A}}$. The eigenvectors of $\bar{\boldsymbol{A}}$ and $\tilde{\boldsymbol{A}}$ are related via a linear transformation [12]. For large values of n the solution of the eigenvalue problem of $\bar{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$ might be numerically difficult. Using the eigenvectors $\boldsymbol{w} \in \mathbb{R}^r$ of $\tilde{\boldsymbol{A}}$ and the relation

$$\phi = \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{w} \tag{2.6}$$

avoids this issue.

2.1.2 DMDc - Identification of Systems with Known Input Signals

For a discrete-time process providing measurements of all inputs $\boldsymbol{u} \in \mathbb{R}^q$ and all states $\boldsymbol{x} \in \mathbb{R}^n$ at regular time intervals Δt the following relation including the linear operators $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times q}$ can be formulated:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k. \tag{2.7}$$

Similar to DMD, for fixed values of $\boldsymbol{A}$ and $\boldsymbol{B}$ the equation can not hold for every measurement trio $\boldsymbol{x}_{k+1}$, $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$. So the DMDc can only provide best-fit solutions of $\boldsymbol{A}$ and $\boldsymbol{B}$. Beside the state measurement matrices $\boldsymbol{X}$ and $\boldsymbol{X'}$ a matrix $\boldsymbol{\Upsilon}$ is formulated by

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_{m-1} \end{bmatrix}.$$

This matrix describes how the inputs evolve over time. Using the measurement matrix $\boldsymbol{\Upsilon}$, relation (2.7) can be reformulated into matrix form as

$$\boldsymbol{X'} \approx \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}\boldsymbol{\Upsilon}. \tag{2.8}$$

At this point of the DMDc procedure two cases are distinguished: Known and unknown $\boldsymbol{B}$. In this thesis only the case that both operators are unknown is considered. Thus the calculations proceed as follows:

Using $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix}$ and $\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{X}^T & \boldsymbol{\Upsilon}^T \end{bmatrix}^T$ relation (2.8) is reformulated to

$$\boldsymbol{X'} \approx \boldsymbol{G}\boldsymbol{\Omega}.$$

In order to directly calculate the best-fit operators $\boldsymbol{A}$ and $\boldsymbol{B}$ this equation is rewritten to

$$\boldsymbol{G} = \boldsymbol{X'}\boldsymbol{\Omega}^+$$

and

$$\begin{bmatrix} A & B \end{bmatrix} = X' \begin{bmatrix} X \\ Y \end{bmatrix}^{+}$$

where Ω^{+} is the pseudo inverse of Ω. Similar to the approach of DMD an order reduction of the system can be embedded performing a truncation in the SVD of $\Omega \approx \tilde{U}\tilde{\Sigma}\tilde{V}^{*}$. The used truncation value is given by

$$p = r + q,$$

where r is the system truncation value and q is the number of inputs. An approximation of A and B can be calculated using the result of the SVD by

$$\begin{bmatrix} A & B \end{bmatrix} \approx \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^{*}. \tag{2.9}$$

In order to obtain a reduced order model the matrix $\tilde{U}$ is splitted into two parts $\tilde{U}_1 \in \mathbb{R}^{n \times p}$ and $\tilde{U}_2 \in \mathbb{R}^{q \times p}$. The next step requires a SVD and truncation of X' by r. This is achieved by calculating

$$X' = U'\Sigma'V'^{*} = \begin{bmatrix} \hat{U} & \hat{U}_{rem} \end{bmatrix} \begin{bmatrix} \hat{\Sigma} & 0 \\ 0 & \hat{\Sigma}_{rem} \end{bmatrix} \begin{bmatrix} \hat{V}^{*} \\ \hat{V}^{*}_{rem} \end{bmatrix} \approx \hat{U}\hat{\Sigma}\hat{V}^{*},$$

where $\hat{U} \in \mathbb{R}^{n \times r}$, $\hat{\Sigma} \in \mathbb{R}^{r \times r}$ and $\hat{V} \in \mathbb{R}^{m-1 \times r}$. This SVD provides the transformation matrix $\hat{U}$ used to project the state vector x onto the linear subspace $\tilde{x}$ in the manner

$$\tilde{x} = \hat{U}^{*}x. \tag{2.10}$$

The measurement vector x can be approximated by

$$x \approx \hat{x} = \hat{U}\tilde{x}.$$

Seperating the rigth hand side of equation (2.9) using $\tilde{U}_1^*$ and $\tilde{U}_2^*$ leads to the expression

$$\begin{bmatrix} A & B \end{bmatrix} \approx \begin{bmatrix} X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^* & X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^* \end{bmatrix}.$$

In (2.7) the measurement vector can be expressed by (2.10). The fact that $\hat{U}^*\hat{U} = I$ for $n > r$ is used to simplify the result leading to

$$\tilde{x}_{k+1} = \hat{U}^* X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^*\hat{U}\tilde{x}_k + \hat{U}^* X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^* u_k$$

where I represents the identity matrix. Reformulating this expression using

$$\tilde{A} = \hat{U}^* X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^*\hat{U}$$

and

$$\tilde{B} = \hat{U}^* X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^*$$

leads to

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}u_k$$

which provids the reduced order system. In DMDc the dynamic modes ϕ of the system basically have similar relations and meaning as in DMD. Performing an eigendecomposition on $\tilde{A}$ is sufficient. The gained eigenvectors w are used for the relation

$$\phi = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^*\hat{U}w \tag{2.11}$$

providing the r dominant dynamic modes of the process.

2.2 Example: System Identification using DMD

A first example shows how DMD is applied to a given system. The given system is modeled and validated using different reduction values r.

2.2.1 Properties of the System

The considered setting consists of an iron rod having length l. The temperature distribution along the single spatial dimension x is captured by a thermal camera at constant time intervals Δt. The thermal conductivity λ, the density ρ and the heat capacity c_p of iron are listed in Table 2.1.

Table 2.1: Model parameters

Parameter	Value	Unit
c_p	462	$\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
ρ	7874	$\mathrm{kg \cdot m^{-2}}$
λ	55	$\mathrm{W \cdot m^{-1} \cdot K^{-1}}$
l	0.2	m
Δt	2	s
Δx	0.01	m

These material properties are constant over the entire rod. Considering the given setting the one dimensional heat equation

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \tag{2.12}$$

can be used to describe the thermal behaviour of the rod. Here κ is the thermal diffusivity

$$\kappa = \frac{\lambda}{\rho c_p}.$$

The partial differential equation (2.12) is discretized in time by using the forward difference quotient

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{k+1} - T_i^k}{\Delta t}$$

and discretized in space by using the central difference quotient

$$\frac{\partial^2 T}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) \approx \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{(\Delta x)^2}$$

where k denotes the time step and i the discrete position. The spatial grid interval Δx is chosen constant over the entire spatial domain. The discretized PDE can be reformulated to

$$T_i^{k+1} = T_i^k + \kappa \Delta t \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{(\Delta x)^2}. \tag{2.13}$$

Both ends of the rod are taken into account as states of the resulting LTI formulation. This means that T_1^k is located at $x = 0\,\text{cm}$ and T_{21}^k at $x = 20\,\text{cm}$. Thus, the system has $n = 21$ states. At both ends, it is assumed that the rod satisfies the homogeneous Neumann boundary condition

$$\left.\frac{\partial T}{\partial x}\right|_{x=0\text{cm},\, x=20\text{cm}} = 0.$$

This means that there are no losses of energy to the ambience. The Neumann boundary condition modifies equation (2.13) for the first of all spatial indices $i = 1, 2, ..., n$ to

$$T_1^{k+1} = T_1^k + \kappa \Delta t \frac{T_2^k - T_1^k}{(\Delta x)^2}.$$

For the last spatial sample $i = n = 21$ a similar expression results by

$$T_{21}^{k+1} = T_{21}^{k} + \kappa \Delta t \frac{T_{20}^{k} - T_{21}^{k}}{(\Delta x)^2}.$$

The resulting discrete-time system is formulated in terms of an autonomous state-space model by

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k$$

It is assumed that the state vector

$$\boldsymbol{x} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{21} \end{bmatrix}$$

is directly measured and contains the temperature of the rod at the discretization nodes.

2.2.2 System Identification and Validation

A simulation of m time steps using a random initial temperature distribution is performed. The resulting data-set including the states $\boldsymbol{x}_k$ over the whole duration is depicted in Figure 2.2. The data matrices $\boldsymbol{X}$ and $\boldsymbol{X'}$ for the DMD are extracted from this record. The mean value of the shown data-set is subtracted before the DMD is performed. This exactly reduces the DMD to the temporal Discrete Fourier Transform (DFT) [5]. A degree of freedom in the DMD is the model reduction value r. The resulting model is given by equation (2.5). This model converges to the origin of the state-space. So, if the model is used for simulation the mean of the state vector $\boldsymbol{x}$ has to be subtracted from $\boldsymbol{x}$ before transforming it into the subspace using (2.2). The estimated state vector $\hat{\boldsymbol{x}}$ is obtained by performing (2.3) to $\tilde{\boldsymbol{x}}$ and adding the mean of $\boldsymbol{x}$. A simulation using the gained model and the same initial states

as for the identification is performed. The impact of the model reduction onto the model error is investigated in Figure 2.3 by using different values for r. Therefore, the error norm E_k of the state vector is calculated for every time step k by

$$E_k = \sqrt{\boldsymbol{e}_k^T \boldsymbol{e}_k}$$

where

$$\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k.$$

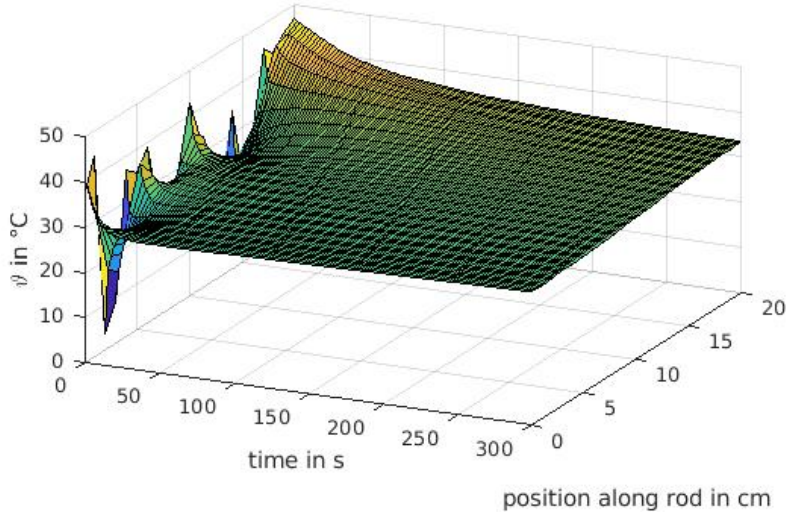


Figure 2.2: Evolution of temperature along a rod over time

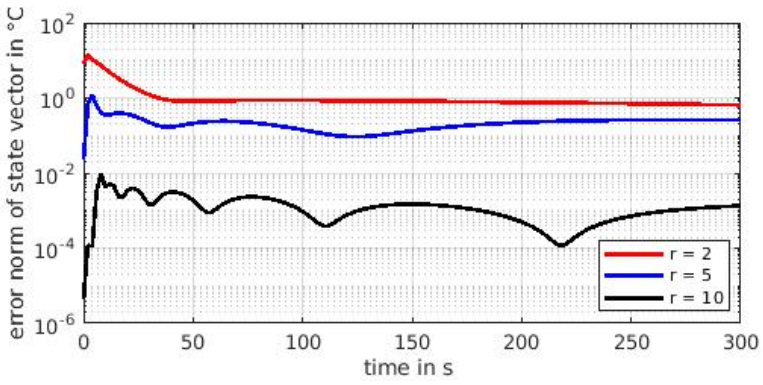


Figure 2.3: Error norm of state vector considering the identification experiment

Validation. To further investigate the quality of the gained system approximation another simulation using a different initial state vector is executed for both the original and the estimated system. The resulting data-set of the original system is shown in Figure 2.4. Again the data-sets of both systems are compared using the error norm E_k. The result can be observed in Figure 2.5.

Discussion. As expected, the error increases by reducing the value of r. The r dominant dynamic modes of the system are computed using (2.6). The resulting curves are depicted in Figure 2.6. The eigenvalues of $\tilde{A}$ are:

Nr	eigenvalue
1	-0.0269
2	0.4264
3	0.8307
4	0.9932
5	0.9246

Table 2.2: Eigenvalues of $\tilde{A}$

The connection between the spatial frequency of each mode and the corresponding eigenvalue is investigated. Comparing both properties of each mode one can observe that they are directly linked to each other. The mode

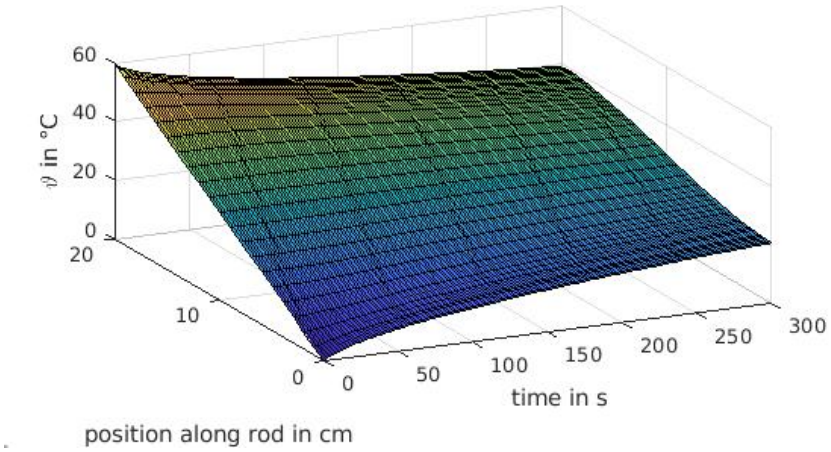


Figure 2.4: Resulting data-set of validation experiment

having the lowest spatial frequency also has the lowest temporal frequency. This is represented by mode 4. In contrast, mode 1 has the highest spatial frequency and the highest temporal frequency. Different modes result if a different data-set gained from the same system is provided to the DMD.

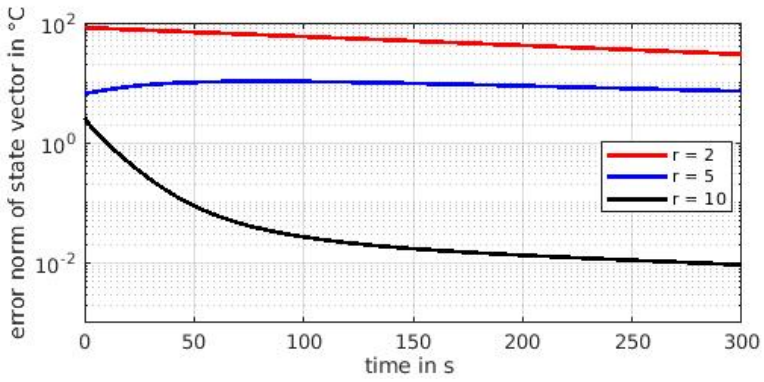


Figure 2.5: Error norm of state vector considering the validation experiment

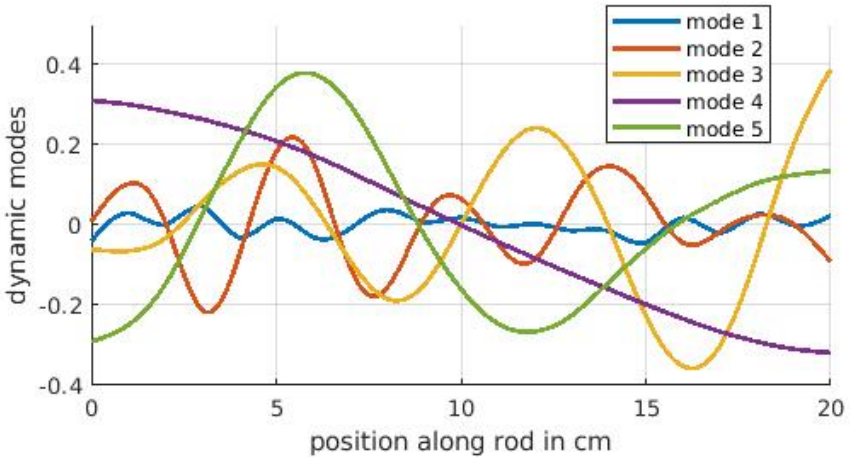


Figure 2.6: Dynamic modes of the model using $r = 5$

3 Modeling of a Laboratory Setting using Dynamic Mode Decomposition with Control

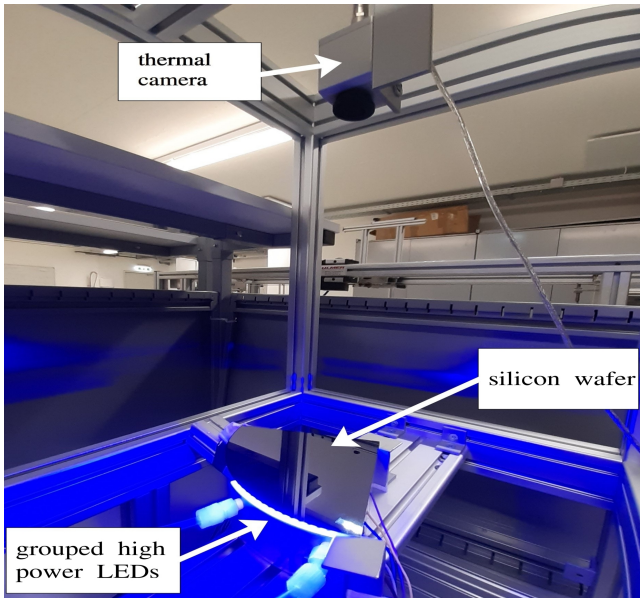


Figure 3.1: Laboratory setting

An existing laboratory setting is modeled by running an identification experiment and using the obtained data in the DMDc. Basically the setting was built to investigate and control thermal effects in a silicon-wafer. As a temperature sensor a thermal camera, detecting the temperature distribution on the surface of the wafer (see Figure 3.1) is installed. It is assumed that the

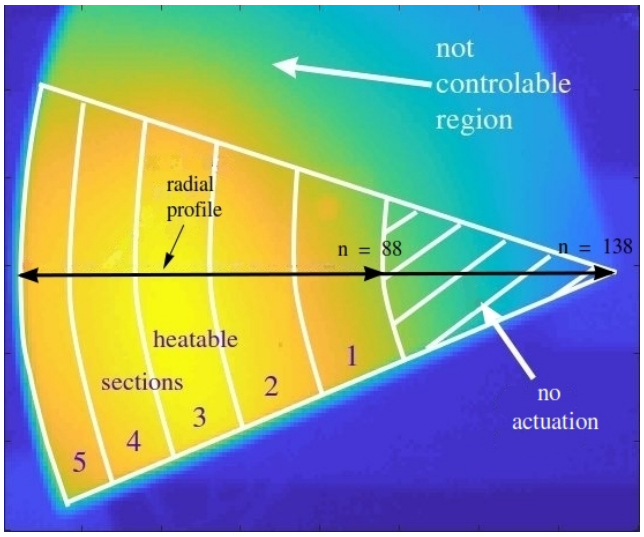


Figure 3.2: Setting seen from the perspective of the thermal camera - top view

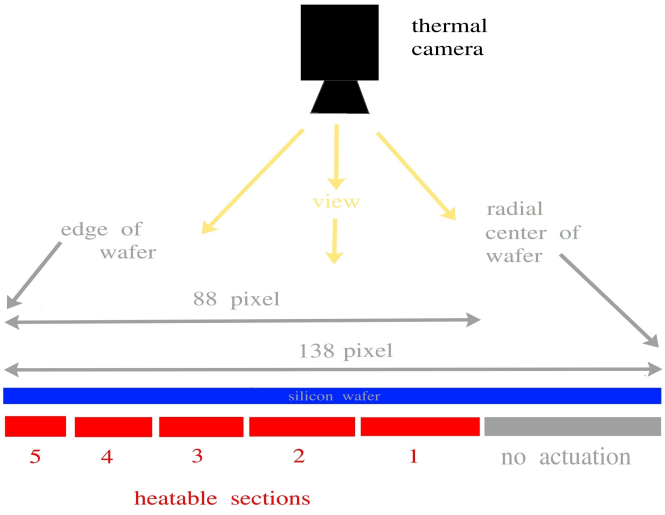


Figure 3.3: Scheme of the setting - side view of the radial profile

setup is rotationally symmetric. Thus, the observation of the wafer along a line from the rotational center to the edge is sufficient. In Figure 3.2 the radial profile is represented by a black line having two different lengths

meaning that there are two different state vectors. The first captures the entire profile including $n = 138$ pixel. The second only captures the heatable sectors of the profile including $n = 88$ pixel. In this regard, the states are directly measured by the camera. The distance between the camera and the surface of the wafer is chosen such that one pixel approximately corresponds to 1mm^2. The actuators of the system are several hundred high power LEDs grouped into six circular sections whereas only the outer five are used. As depicted in Figure 3.2 it is not possible to actuate the innermost section. The control unit of the setting is realized by a conventional personal computer (PC). It gathers the data of the thermal camera and outputs actuating signals using an Arduino Due micro controller. The micro controller provides the supply units of the LED sections with the actuating signals via power switches. In order to operate the system, Matlab® is embedded as the main software component.

3.1 Sampling Time and Time Constants

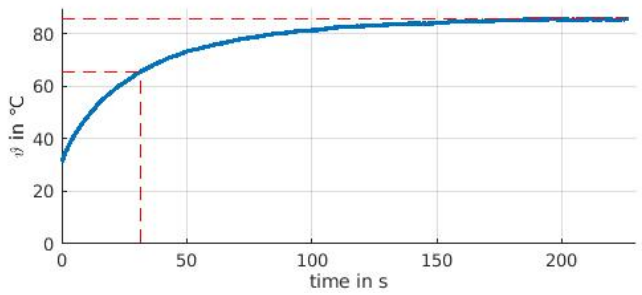


Figure 3.4: Investigating the fastest responding state of the system by step response

For the modeling procedure the time constants of the system are investigated. Figure 3.4 illustrates the step response resulting from a step from 0% to 40% heater power at the outermost section. The shown temperature curve referes to the fastest responding state recorded by the thermal camera. This state is located in the vicinity of the center of the actuated section. The time constant $\tau = 31.72\,\text{s}$ is found by looking for the point in time where 63.2%

of the steady-state temperature is reached. For this calculation the ambient temperature ϑ_a has been subtracted from the steady-state temperature. The maximum allowed samling time Δt_{max} is obtained from the nyquist criterion, i.e.

$$f_c = \frac{1}{\tau} = 0.03153\,\text{Hz}$$

$$f_s > 2f_c = 0.06306\,\text{Hz} \rightarrow \Delta t_{max} = 15.86\,\text{s} \rightarrow \underline{\Delta t_{chosen} = 0.5\,\text{s}}$$

As expected in thermal diffusion processes the time constants are large. The sampling time is chosen as $\Delta t_{chosen} = 0.5\,\text{s}$.

3.2 System Identification using a Single Actuator

In order to investigate if a system identification via DMDc is suitable for the present system, only a single actuator is used at first. Looking at the system from the modeling and control perspective, it is obvious that the autonomous system must converge towards the ambiente temperature ϑ_a. If ϑ_a is not measured it has to be considered as a disturbance acting onto the entire state vector. In the present case ϑ_a is directly recorded by the thermal camera and can be treated as a known external input. In the DMDc there are two possibilities to deal with this effect. In the first approach ϑ_a is treated as an input of the DMDc. In the second it is subtracted from the state data-set.

3.2.1 ϑ_a as Input of the DMDc

Input data for the DMDc is generated by performing an open loop experiment on the laboratory setup. For this experiment only the outermost actuator is used. The actuating signal u as well as the ambiente temperature ϑ_a are shown in Figure 3.5 over the entire duration of the experiment. These signals form the first input matrix $\boldsymbol{\Upsilon}$ of the DMDc. The corresponding record

of the wafer surface temperature is shown in Figure 3.6. It represents the evolution of x over the experiment duration and also provides the matrices $\boldsymbol{X}$ and $\boldsymbol{X}'$ for the DMDc. The reduction value is chosen by $r = 10$ whereas in the present case the original system order is $n = 138$ since only the pixel along a line from the radial center to the edge of the wafer (see Figure 3.2) are considered as states of the system in this case.

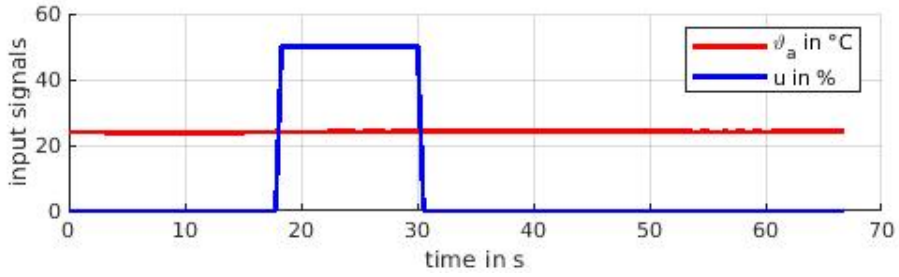


Figure 3.5: Input signals of the system

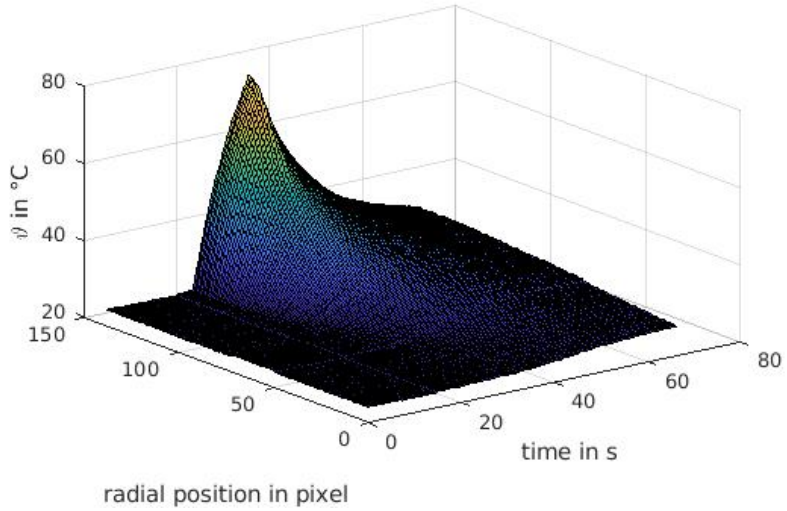


Figure 3.6: Evolution of the states over time and space

Results. As a first evaluation of the gained model the input shape functions of the two input signals ϑ_a and u are calculated by

$$\hat{\boldsymbol{B}} = \hat{\boldsymbol{U}}\tilde{\boldsymbol{B}}. \tag{3.1}$$

This transformation is necessary because the functions of interest refer to the state vector $\boldsymbol{x}$ and not $\tilde{\boldsymbol{x}}$. An input shape function expresses how an input acts onto the state vector. An input may influence some states i.e. with a positive sign, others i.e. with a negative sign or a different intensity. The functions are investigated by having a look at Figure 3.7. Here one can observe that the way how ϑ_a acts onto the states does not make much sense since the ambiente temperature should affect all states equally. The impact of the second input signal u seems to match the reality quite well because the actuated area is located at the outermost region of the wafer. This is adequately represented by the input shape function of u. Due to the unrealistic input shape function of ϑ_a a different approach of providing the data to the DMDc is tried in the following section.

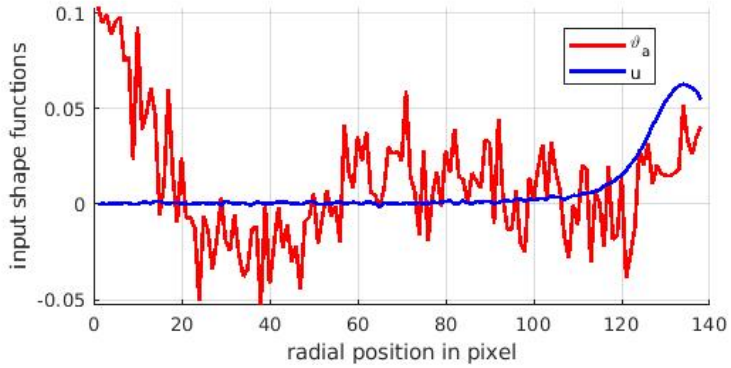


Figure 3.7: Input shape functions

3.2.2 ϑ_a as Base Flow

In contrast to considering ϑ_a as input signal in the DMDc it can also be seen as an equilibrium point or *base flow* [5] [7] of the system. This *base*

flow is now subtracted from the data-set before the DMDc is performed. Thus, only the time series of u is used in $\boldsymbol{Y}$. The matrix $\boldsymbol{X}$ remains the same as in Section 3.2.1. The reduction value again is chosen by $r = 10$. As one can observe in Figure 3.8 the input shape function of u is approximately identical to the one resulting from the approach of Section 3.2.1. Due to the fact that the input shape function of ϑ_a, shown in Figure 3.7, does not match the reality, the approach of using ϑ_a *as base flow* is chosen for the following investigations and also for the controller designs.

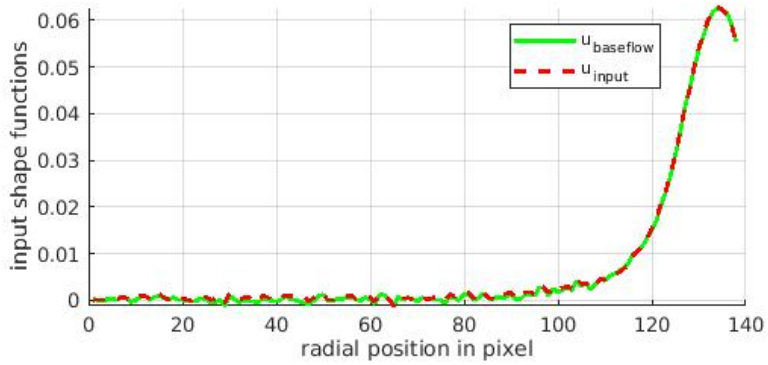


Figure 3.8: Comparing the input shape function of u of both attitudes 3.2.1 and 3.2.2

3.3 System Identification using all Actuators

The input data for the DMDc is now generated by performing an identification experiment using all 5 actuators. An appropriate sequence how to switch them over the experiment duration has to be found. A random pattern like white noise is a common choice in system identification. The pattern used to identify the present setup is depicted in Figure 3.9. The signals are generated by the Matlab command *randi*. In order to not exceed the maximum temperature, breaks need to be taken frequently during the record. Figure 3.10 shows how the resulting temperature of the wafer, including $n = 138$ pixel along a radial profile, evolves over time and space.

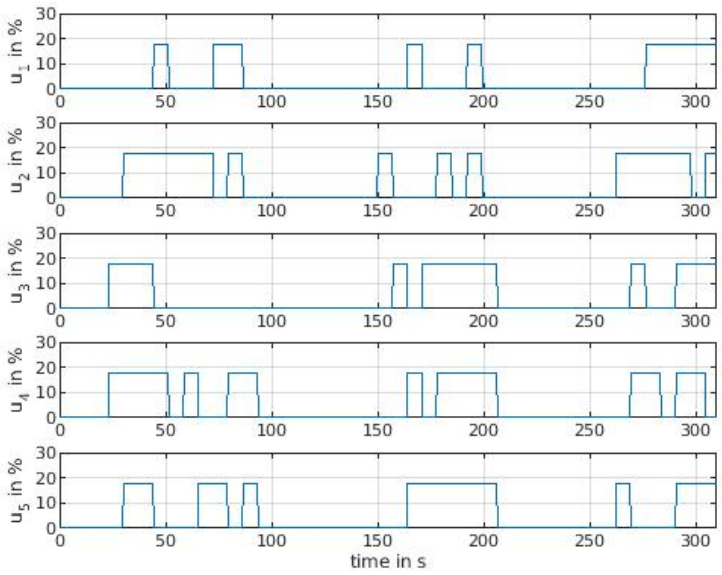


Figure 3.9: Impulse pattern - identification experiment

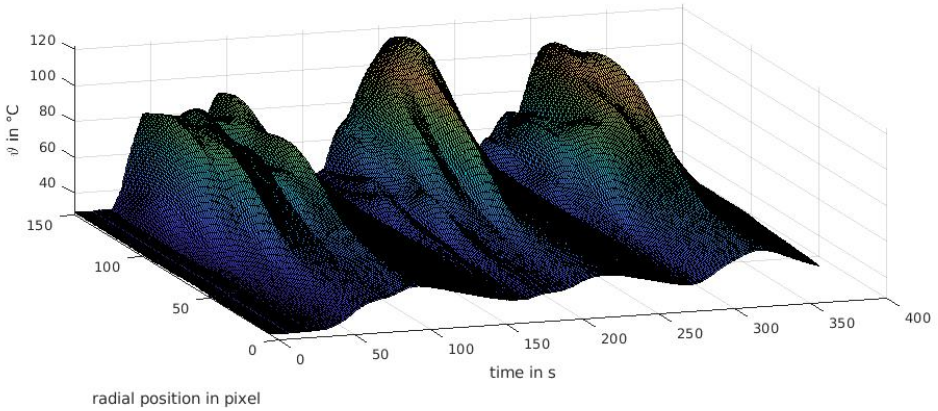


Figure 3.10: Behaviour of the wafer surface (n=138) along a radial line from the center to the edge during the identification experiment

Having a look at Figure 3.10 it is conspicuous that pixel $1-62$ can not be actuated. ***Thus: Any following calculations and controller designs are aimed to control only the actuatable region of the wafer having*** $n = 88$ ***states!***

3.3.1 Applying the DMDc

As mentioned in Section 3.2.2, considering ϑ_a as baseflow leads to better results. Thus, for the identification of the Multiple Input Multiple Output (MIMO) system the value of ϑ_a is subtracted from the entire state data-set of the identification record shown in Figure 4.1. This transformed data-set as well as the data of Figure 3.9 provide the matrices $\boldsymbol{X}$, $\boldsymbol{X'}$ and $\boldsymbol{Y}$ for the DMDc. The system order reduction value is chosen by $r = 10$, whereas the original system order is $n = 88$. The identification returns a discrete-time state-space model

$$\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k \tag{3.2}$$

discribing the dynamical behaviour in terms of a linear time-invariant (LTI) system. The state vector $\tilde{\boldsymbol{x}}_k$ of the reduced order system is computed by

$$\tilde{\boldsymbol{x}}_k = \hat{\boldsymbol{U}}^*(\boldsymbol{x}_k - \boldsymbol{\vartheta}_a) \tag{3.3}$$

and the measured state vector $\boldsymbol{x}_k$ is approximated by

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{U}}\tilde{\boldsymbol{x}}_k + \boldsymbol{\vartheta}_a.$$

Results. The gained input shape functions are calculated by (3.1) and picking the columns out of $\hat{\boldsymbol{B}}$ (see Figure 3.11). The input shape functions seem to match the reality quite well since the center of each heatable section is affected the most by the corresponding group of LEDs. Therefore, the peak of each curve is at the center of each section.

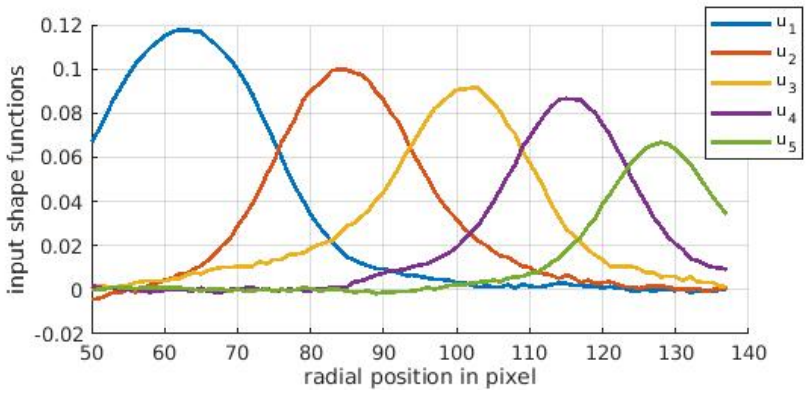


Figure 3.11: Gained input shape functions of the multiple input system

The DMDc analysis also provides the r dominant dynamic modes of the system [12]. To obtain these modes equation (2.11) is solved. The modes are the eigenvectors of the dynamic matrix $\bar{\boldsymbol{A}}$. A benefit using DMD is that this large matrix does not need to be computed. The graphical representation of the modes is depicted in Figure 3.12. The eigenvalues of $\tilde{\boldsymbol{A}}$ are:

Nr	eigenvalue
1	0.2361
2	0.4457
3	0.5613
4	0.6314
5	0.7331

Nr	eigenvalue
6	0.8235
7	0.9905
8	0.9672
9	0.8920
10	0.9388

Table 3.1: Eigenvalues of $\tilde{A}$

A validation experiment is performed in order to get an impression if the model is useful outside the data-set of the identification experiment. For this purpose a pseudo random actuating signal is used. The evolution of the state vector $\boldsymbol{x}$ during the validation experiment is depicted in Figure 3.13. Also the corresponding simulation data using the model can be found here. The error norm of the state vector is calculated following the scheme of Section 2.2.2 providing a qualitative investigation of the gained model in Figure 3.13. The conclusion can be drawn that the gained LTI model

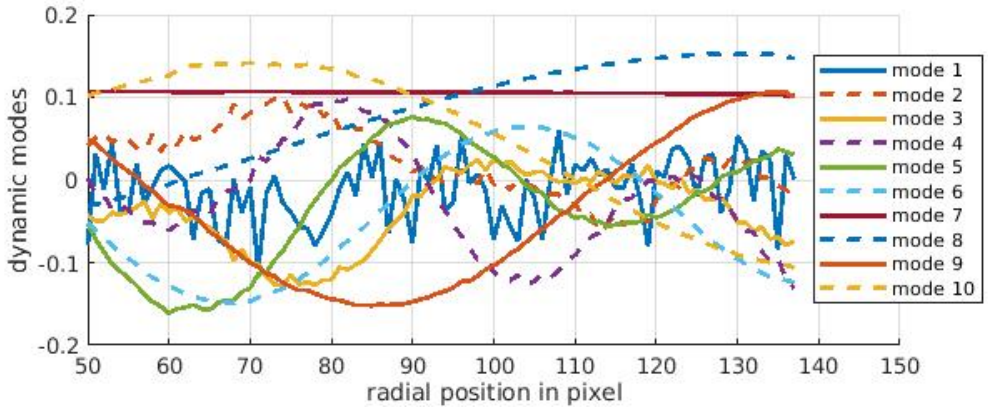
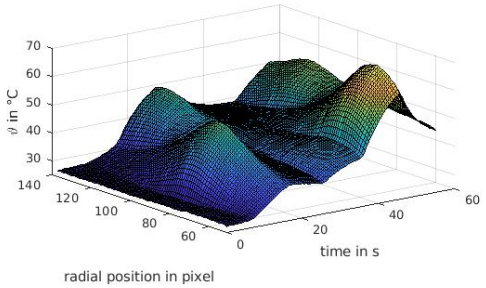
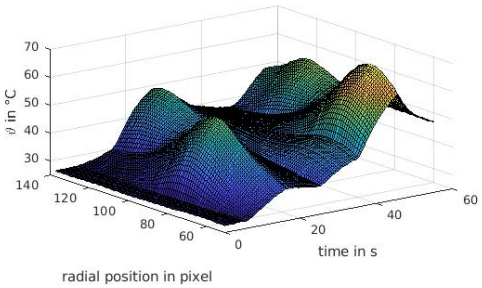


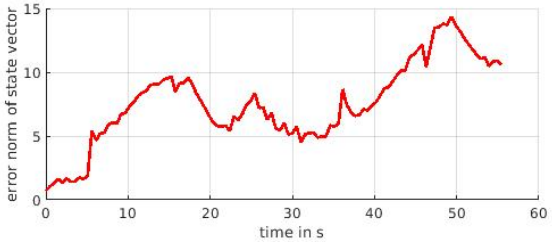
Figure 3.12: Dynamic modes of the system



(a) Measured behaviour of real system



(b) Behaviour of model



(c) Error norm of state vector using $r = 10$

Figure 3.13: Validation experiment

approximates the process sufficiently to approach model based controller design.

4 Controller design

A desireable goal of the controller design is tracking or stabilization of the temperature distribution along the wafer to a given constant temperature profile. In order to achieve that goal, different control approaches can be chosen. In this section model-based approaches are exploited. First a state feedback controller based on the Linear Quadratic Regulator (LQR) [1] approach including tracking is designed. The chapter continues implementing and simulating a Model Predictive Controller (MPC) [4].

4.1 Controllability

In order to check the controllability of the system resulting from DMDc the matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are used for the following investigation. The controllability matrix

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} \tilde{\boldsymbol{B}} & \tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} & \tilde{\boldsymbol{A}}^2\tilde{\boldsymbol{B}} & \dots & \tilde{\boldsymbol{A}}^{r-1}\tilde{\boldsymbol{B}} \end{bmatrix}$$

is computed via *Kalman's criterion of controllability*. If $\boldsymbol{\mathcal{C}}$ has full row rank the system is controllable. In the present case the row rank equals the reduced system order r. Thus, the system is fully controllable.

4.2 Linear Quadratic Regulator

In order to use a state feedback controller for the present setting, the state vector $\boldsymbol{x}$ needs to be transformed by using equation (3.3). The resulting

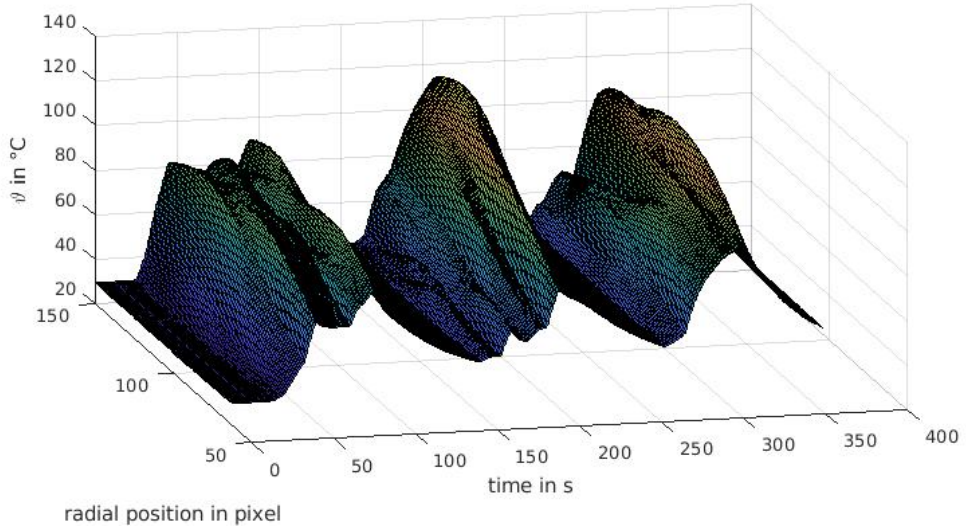


Figure 4.1: State vector x (n=88) during the identification experiment

state vector $\tilde{x}$ can now be used as an input for the tracking state feedback controller

$$u_k = V\tilde{x}_{ref,k} - K\tilde{x}_k \tag{4.1}$$

where the transformed reference state vector $\tilde{x}_{ref,k}$ is calculated by applying (3.3) to the reference state vector $x_{k,ref}$. Thus the control target is $x_{ref,k}$. An overall scheme of the control loop structure can be found in Figure 4.2. The feedback matrix K is computed by the Matlab command *dlqr*. The system model matrices $\tilde{A}$ and $\tilde{B}$ as well as the weighting matrix of the states Q and the weighting matrix of the input signals R are the parameters.

Beside the preceding reduction value r and the record length m the matrices Q and R represent the degrees of freedom in the entire controller design including modeling. Their values are chosen as

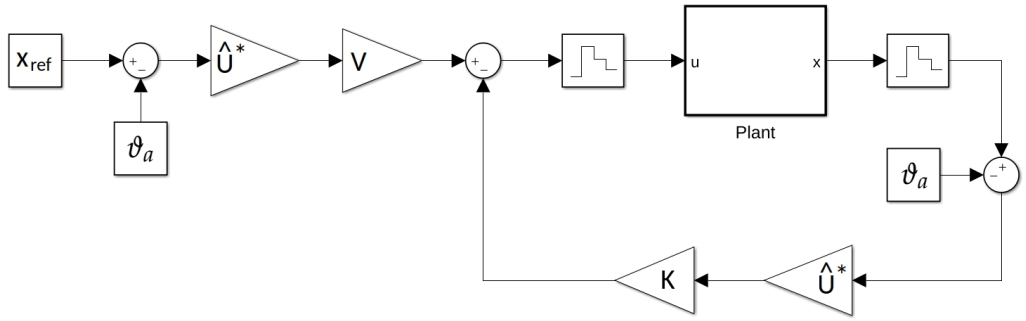


Figure 4.2: Block scheme of closed loop using an LQR

$$Q = \begin{bmatrix} 0.005 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

An advantage of using an LQR approach instead of Ackermann's formula is the more intuitive placement of the eigenvalues via a cost function. Since the model gained from the DMDc is based on coupled modes, tuning Q and R is not straightforward. Thus the present choice of Q is created following a trial-and-error procedere.

4.2.1 Calculation of V

In order to gain a tracking controller the matrix $\boldsymbol{V}$ has to be computed. Therefor the state-space model (3.2) is reformulated using the feedback law (4.1) to

$$\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{B}}\boldsymbol{K}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{V}\tilde{\boldsymbol{x}}_{ref,k}.$$

In steady-state $\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{x}}_k$ holds and it is assumed that the transformed control target

$$\tilde{\boldsymbol{x}}_{ref,k} = \hat{\boldsymbol{U}}^*(\boldsymbol{x}_{ref,k} - \boldsymbol{\vartheta}_a)$$

is reached. This means that

$$\tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_{ref,k}.$$

This results into the reformulation

$$\tilde{\boldsymbol{x}}_{ref,k} = (\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\boldsymbol{K} + \tilde{\boldsymbol{B}}\boldsymbol{V})\tilde{\boldsymbol{x}}_{ref,k}$$

and further to

$$\boldsymbol{0} = (\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\boldsymbol{K} + \tilde{\boldsymbol{B}}\boldsymbol{V} - \boldsymbol{I})\tilde{\boldsymbol{x}}_{ref,k}$$

where $\boldsymbol{I}$ is the identity matrix. The equation is only valid if

$$\boldsymbol{0} = \tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\boldsymbol{K} + \tilde{\boldsymbol{B}}\boldsymbol{V} - \boldsymbol{I}.$$

This is further reshaped to

$$\tilde{\boldsymbol{B}}\boldsymbol{V} = \tilde{\boldsymbol{B}}\boldsymbol{K} + \boldsymbol{I} - \tilde{\boldsymbol{A}} = -(\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\boldsymbol{K} - \boldsymbol{I})$$

and finally

$$\boldsymbol{V} = -\tilde{\boldsymbol{B}}^{+}(\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\boldsymbol{K} - \boldsymbol{I}),$$

whereas $\tilde{\boldsymbol{B}}^{+}$ is the pseudo inverse of $\tilde{\boldsymbol{B}}$.

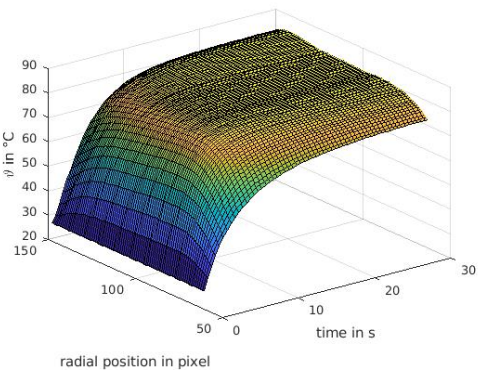
4.2.2 Simulation of the Closed Loop

For the simulation of the closed loop depicted in Figure 4.2 the plant is realized by an LTI model, generated following the scheme explained in Section 3.3.1 using $r = n = 88$. This results into the most exact model of the process so far. The results of a closed loop simulation are shown in Figure 4.3. Here, a value of $80°C$ along the entire spatial domain is chosen as a control target for $t > 0\,\mathrm{s}$. Therefore, no limitation of the actuating signals is implemented in the simulation. Figure 4.3 shows that the control signal $\boldsymbol{u}$ only exceeds a possible upper limit of 60% heater power for a short periode of time at the beginning of the reference step. Having a look at the error of the state vector in steady-state one can observe that the states near the edge and the center of the wafer can not be actuated in a way that they match the set-point temperature due to the fact that there are no LEDs in this area. Nevertheless the controller seems promissing for trying it at the laboratory setup.

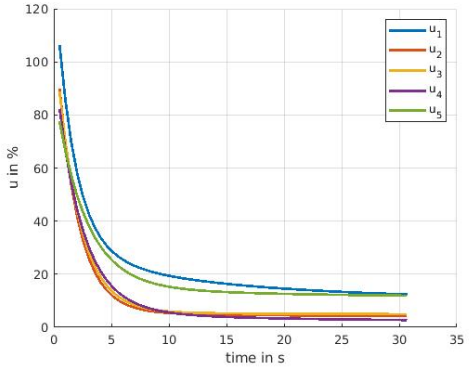
4.2.3 Closed Loop Experiment using the Laboratory Setting

The actuating signals are limited to $0 - 60\%$ since the LEDs will be destroyed if powered with more than 60% for a time range exceeding a couple of seconds. Furthermore the power supply units are not able to run below 10%. To be able to use this important zone the heaters are switched off completely by relais. Thus, the actual input signals $u_{sat,i}$ follow the scheme

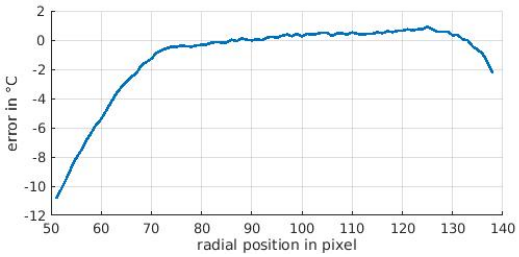
$$u_{sat,i} = \begin{cases} 0\% & for \quad u_i < 5\% \\ 10\% & for \quad 5\% \leq u_i < 10\% \\ u_i & for \quad 10\% \leq u_i < 60\% \\ 60\% & for \quad 60\% \leq u_i \end{cases}$$



(a) State vector over time



(b) Actuating signals

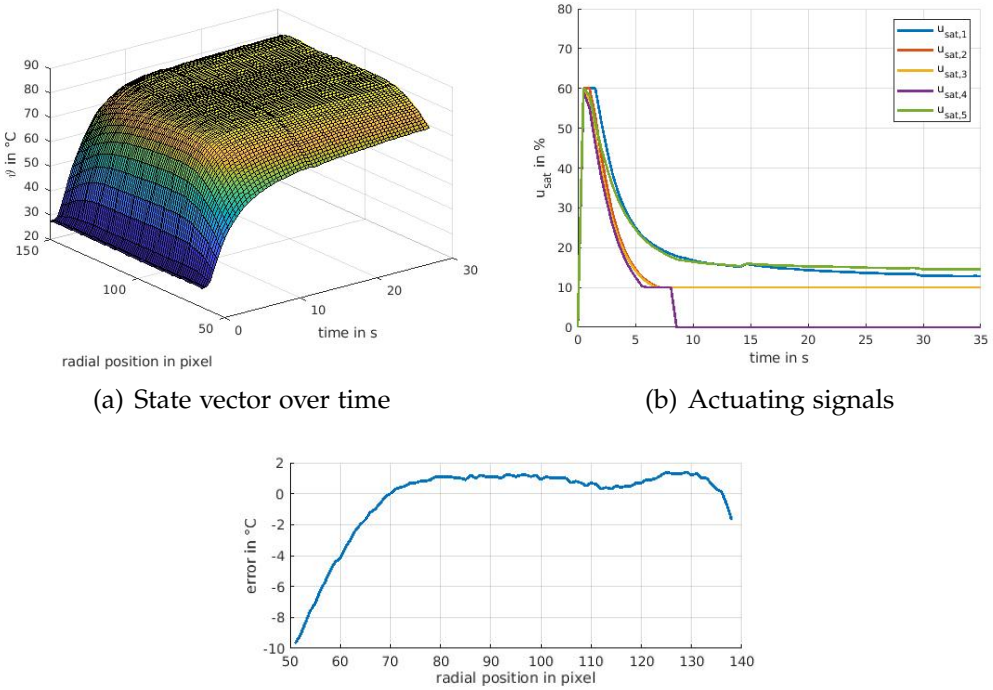


(c) Remaining error at steady-state

Figure 4.3: Simulation - step response of closed loop system

which is also represented by the actuating signals in Figure 4.4. A more continuous representation of the range $0 - 10\%$ could be achieved by translating $u_{sat,i}$ to a pulse-width modulation signal. Since the power is switched by conventional relais having limited number of switching events this is not approached here.

Results. Comparing Figure 4.3 and Figure 4.4 one can observe that the results gained from the experiment match with the plots of the closed-loop simulation. The region of nonlinearity at $0 - 10\%$ heater power is entered by $u_{sat,3}$ and $u_{sat,4}$ when $\boldsymbol{x}_k$ is at the vicinity of the setpoint $\boldsymbol{x}_{ref,k}$.



(a) State vector over time

(b) Actuating signals

(c) Remaining error at steady-state

Figure 4.4: Experiment - step response of closed loop system

4.3 Model Predictive Control

The model gained using DMDc can also be exploited for designing a MPC. For a 2D thermal diffusion problem this has been done in [10]. A benefit of using MPC is the option of embedding constraints. These can be used to restrict the values of the actuating signals to the present limits since beneath 0% and upon 60% the corresponding $u_{sat,i}$ is in saturation.

4.3.1 Reference Signal

Using an MPC the reference signal is a matrix $\boldsymbol{L} \in \mathbb{R}^{(N_p \cdot n) \times t_{sim}}$ whereas t_{sim} is the number of simulation steps and N_p is the prediction horizon. In

the present case the reference is selected as a constant temperature of 80°C over the entire space and time. Thus the matrix $\boldsymbol{L}$ is completely filled with this constant values. Since the reference is constant in time, $\boldsymbol{L}$ is reduced to a time-independent reference state vector $\bar{\boldsymbol{x}}_{ref} \in \mathbb{R}^{(N_p \cdot n) \times 1}$ including the desired surface temperature distribution until N_p. The reference $\bar{\boldsymbol{x}}_{ref}$ is transformed by modifying relation (3.3) to

$$\bar{\tilde{\boldsymbol{x}}}_{ref} = \hat{\boldsymbol{U}}^* (\bar{\boldsymbol{x}}_{ref} - \boldsymbol{\vartheta}_a) \tag{4.2}$$

where $\hat{\boldsymbol{U}} \in \mathbb{R}^{(N_p \cdot n) \times (N_p \cdot r)}$ is the block diagonal matrix

$$\hat{\boldsymbol{U}} = \begin{bmatrix} \hat{\boldsymbol{U}} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{U}} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \hat{\boldsymbol{U}} \end{bmatrix}.$$

As an output (4.2) provides the reference input $\bar{\tilde{\boldsymbol{x}}}_{ref}$ of the MPC (see Figure 4.5).

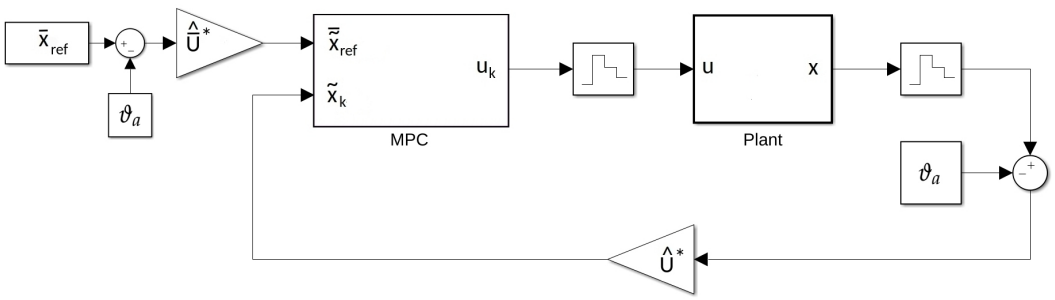


Figure 4.5: Block scheme of closed loop using MPC

4.3.2 Prediction

The MPC is designed using the reduced order system. This is also represented by the block scheme in Figure 4.5. The state vector $\tilde{\boldsymbol{x}}_k$ is predicted

for every time step until the end of the prediction horizon N_p is reached. Therefor the relation

$$\bar{\tilde{\boldsymbol{x}}}_{k+1} = \boldsymbol{F}\tilde{\boldsymbol{x}}_k + \boldsymbol{G}\boldsymbol{u}_{k-1} + \boldsymbol{H}\,\Delta\bar{\boldsymbol{u}}_k \tag{4.3}$$

is used. Here $\bar{\tilde{\boldsymbol{x}}}_{k+1} \in \mathbb{R}^{(N_p \cdot r)\times 1}$ includes the predicted state vector $\hat{\tilde{\boldsymbol{x}}} \in \mathbb{R}^{r\times 1}$ for all prediction steps up to N_p. The input $\Delta\bar{\boldsymbol{u}}_k \in \mathbb{R}^{(q\cdot N_c)\times 1}$ of the equation contains the variation of the actuating signal $\Delta\boldsymbol{u} \in \mathbb{R}^{q\times 1}$ up to the control horizon N_c. Since $\Delta\bar{\boldsymbol{u}}_k$ is an input of (4.3) it represents the degrees of freedom here. The parameter matrices $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{H}$ need to be calculated using the state-space model of the plant by

$$\boldsymbol{F} = \begin{bmatrix} \tilde{\boldsymbol{A}} \\ \tilde{\boldsymbol{A}}^2 \\ \vdots \\ \tilde{\boldsymbol{A}}^{N_p} \end{bmatrix},$$

$$\boldsymbol{G} = \begin{bmatrix} \tilde{\boldsymbol{B}} \\ (\tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} \\ \vdots \\ (\tilde{\boldsymbol{A}}^{N_p-1} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} \end{bmatrix},$$

and

$$\boldsymbol{H} = \begin{bmatrix} \tilde{\boldsymbol{B}} & 0 & \dots & 0 \\ (\tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & \tilde{\boldsymbol{B}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\tilde{\boldsymbol{A}}^{N_c-1} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & (\tilde{\boldsymbol{A}}^{N_c-2} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & \dots & \tilde{\boldsymbol{B}} \\ (\tilde{\boldsymbol{A}}^{N_c} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & (\tilde{\boldsymbol{A}}^{N_c-1} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & \dots & (\tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} \\ \vdots & \vdots & \ddots & \vdots \\ (\tilde{\boldsymbol{A}}^{N_p-1} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & (\tilde{\boldsymbol{A}}^{N_p-2} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} & \dots & (\tilde{\boldsymbol{A}}^{N_p-N_c} + \cdots + \tilde{\boldsymbol{A}} + \boldsymbol{I})\tilde{\boldsymbol{B}} \end{bmatrix}.$$

4.3.3 Optimization Problem

The objective function used in the MPC approach is quadratic and includes the control error of the reduced order system up to the prediction horizon N_p and the variation of the actuating signal $\boldsymbol{\Delta u}_k$ up to the control horizon N_c. The objective is given by

$$J = \sum_{i=1}^{N_p} \left(\hat{\tilde{\boldsymbol{x}}}_{k+i} - \tilde{\boldsymbol{x}}_{ref,i}\right)^T \boldsymbol{Q}_i \left(\hat{\tilde{\boldsymbol{x}}}_{k+i} - \tilde{\boldsymbol{x}}_{ref,i}\right) + \sum_{i=1}^{N_c} \boldsymbol{\Delta u}_{k+i-1}^T \boldsymbol{R}_i \, \boldsymbol{\Delta u}_{k+i-1}$$

where $\tilde{\boldsymbol{x}}_{ref,i} \in \mathbb{R}^{r\times 1}$ is picked segment wise out of $\bar{\boldsymbol{x}}_{ref}$ for every step i up to N_p. The matrices $\boldsymbol{Q}_i \in \mathbb{R}^{N_p \times N_p}$ and $\boldsymbol{R}_i \in \mathbb{R}^{N_c \times N_c}$ are weighting matrices. The objective and the constraints are formulated in terms of a quadratic program [3] by

$$\begin{aligned} \underset{\boldsymbol{\Delta \bar{u}}_k \in \mathbb{R}^{m\cdot N_c}}{min} \quad & \boldsymbol{\Delta \bar{u}}_k^T (\boldsymbol{H}^T \boldsymbol{Q} \boldsymbol{H} + \boldsymbol{R}) \, \boldsymbol{\Delta \bar{u}}_k + 2 \, \boldsymbol{\Delta \bar{u}}_k^T \boldsymbol{H}^T \boldsymbol{Q} \bar{\boldsymbol{e}}_k \\ s.t. \quad & \boldsymbol{W} \, \boldsymbol{\Delta \bar{u}}_k \leq \bar{\boldsymbol{w}} \end{aligned} \tag{4.4}$$

where

$$\bar{\boldsymbol{e}}_k = \boldsymbol{F} \tilde{\boldsymbol{x}}_k + \boldsymbol{G} \boldsymbol{u}_{k-1} - \bar{\tilde{\boldsymbol{x}}}_{ref}.$$

The matrix

$$\boldsymbol{W} = \begin{bmatrix} -\boldsymbol{M} \\ \boldsymbol{M} \end{bmatrix}$$

and the vector

$$\bar{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{L} \, \boldsymbol{u}_{k-1} - \bar{\boldsymbol{u}}_{min} \\ \bar{\boldsymbol{u}}_{max} - \boldsymbol{L} \, \boldsymbol{u}_{k-1} \end{bmatrix}$$

are needed for the constraints given by (4.4). Here $\bar{\boldsymbol{u}}_{min} \in \mathbb{R}^{(q \cdot N_c) \times 1}$ stands for the lower limit and $\bar{\boldsymbol{u}}_{max} \in \mathbb{R}^{(q \cdot N_c) \times 1}$ for the upper limit of the actuating signal. The values are chosen constant over the control horizon N_c by $\bar{\boldsymbol{u}}_{min} = \mathbf{0}$ and $\bar{\boldsymbol{u}}_{max} = \mathbf{60}$. The lower triangular matrix

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{I} & \boldsymbol{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}$$

and the matrix $\boldsymbol{L} = [\boldsymbol{I} \quad \boldsymbol{I} \quad \dots \quad \boldsymbol{I}]^T$ are needed to assemble $\boldsymbol{W}$ and $\bar{\boldsymbol{w}}$. The matrix $\boldsymbol{I} \in \mathbb{R}^{q \times q}$ represents the identity matrix. The tuning parameters of this MPC formulation are the horizons N_c and N_p as well as the matrices $\boldsymbol{Q} \succ 0$ and $\boldsymbol{R} \succ 0$.

For solving the optimization problem numerically the open source toolbox *Yalmip* is embedded into the Matlab simulation environment. A quadratic programming solver is provided by choosing the option *quadprog* in the Yalmip settings. Figure 4.6 shows the inner structure of the MPC used in Figure 4.5. Here the embedded Matlab function block performs the optimization using Yalmip in every time step. The used optimization variables are represented by $\boldsymbol{\Delta}\bar{\boldsymbol{u}}_k \in \mathbb{R}^{(q \cdot N_c) \times 1}$.

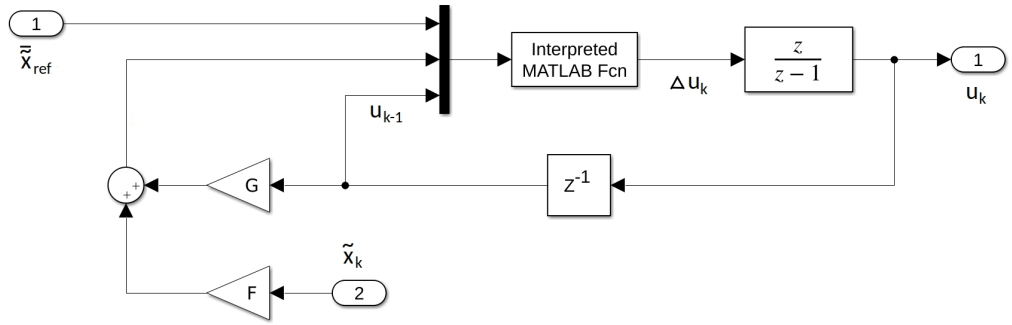


Figure 4.6: Inner structure of the MPC

4.3.4 Simulation of the Closed Loop

A block diagram representing the closed loop is shown in Figure 4.5. The model embedded in the MPC is generated following the scheme of Section 3.3.1 using $r = 8$. In order to keep the computational effort of the optimization low, a small value for r is chosen here. The modes of the MPC-model are depicted in Figure 4.7. The corresponding eigenvalues are:

Nr	eigenvalue
1	$0.5471 + 0.0475j$
2	$0.5471 - 0.0475j$
3	0.9905
4	0.9679
5	0.9236
6	0.6318
7	0.7547
8	0.8393

Table 4.1: Eigenvalues of $\tilde{A}$

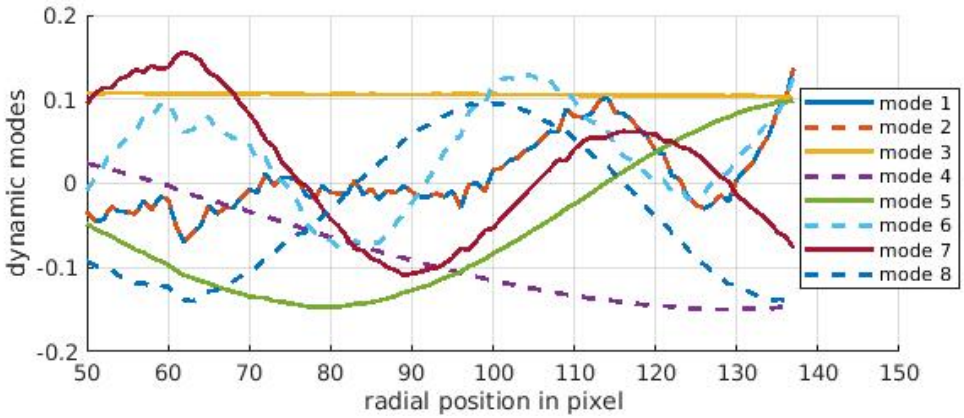


Figure 4.7: Modes of the system using r=8

For the simulation of the closed loop using the present MPC the plant shown in Figure 4.5 is realized the same way as the plant of Section 4.2.2. The parameters of the MPC are chosen by $N_c = 8$, $N_p = 10$, $\boldsymbol{R} = \boldsymbol{I}$ where $\boldsymbol{I} \in \mathbb{R}^{q \times q}$ is the identity matrix and

$$\boldsymbol{Q} = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.45 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 \end{bmatrix}.$$

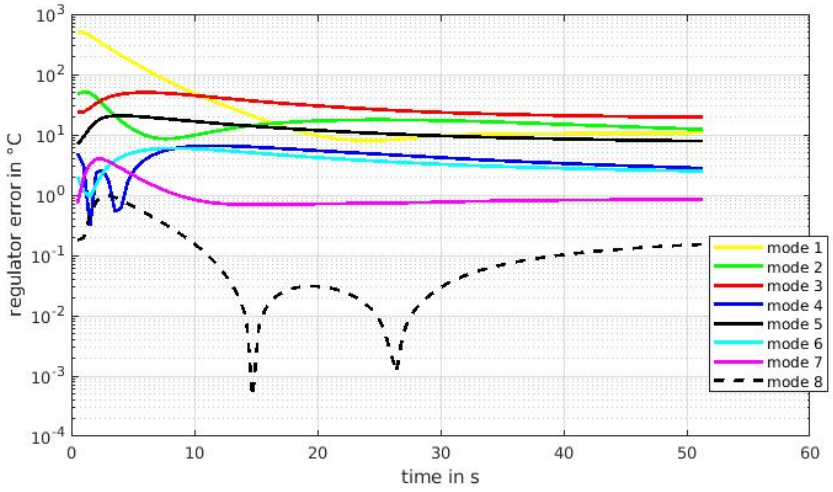
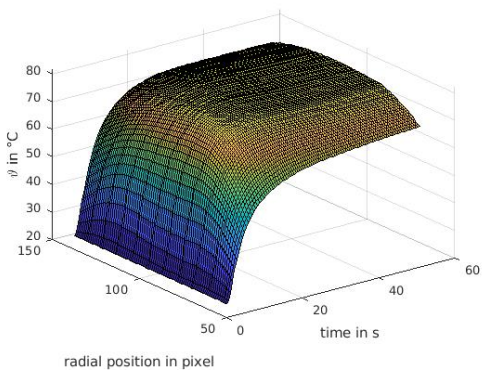
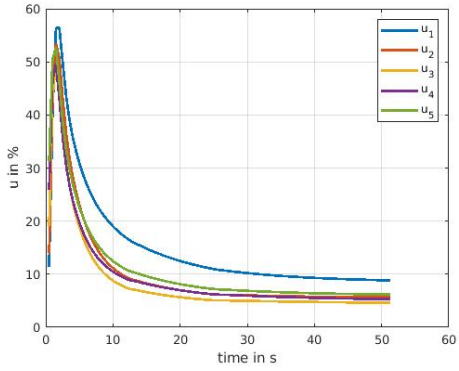


Figure 4.8: Control error considering the modes

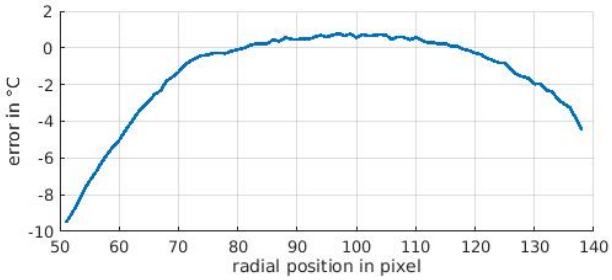
The results of the simulation are depicted in Figure 4.8 and 4.9. The actuating signals do not exceed the range of $0 - 60\%$ heater power. The remaining error considering $\boldsymbol{x}$ at steady-state can still be improved by tuning the parameters. Since the MPC works with the reduced order model, the parameter matrix $\boldsymbol{Q}$ is the weighting for the error of the modes. This matrix



(a) Temperature over time



(b) Actuating signals



(c) Remaining error considering $\boldsymbol{x}$ at steady-state

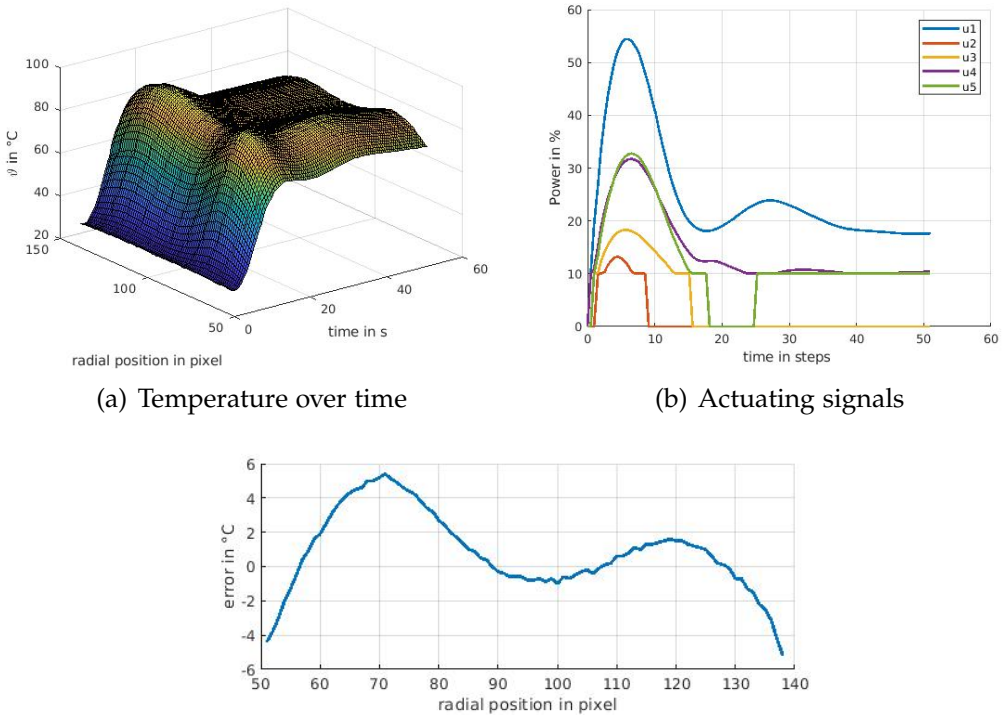
Figure 4.9: Simulation - step response of closed loop system using MPC

has been tuned empirically. The evolution of the control error $\tilde{\boldsymbol{x}}_{ref,k} - \tilde{\boldsymbol{x}}_k$ is investigated in Figure 4.8. Since the modes are coupled and since they are all linked to every state of $\boldsymbol{x}$ having a non-zero value in dimension y in Figure 4.7 the control error of the reduced order system never vanishes for $\tilde{\boldsymbol{x}}$.

4.3.5 Closed Loop Experiment using the Laboratory Setting

Having a look at Figure 4.10 one can observe that an overshoot occurs if a reference step to 80°C over the entire state-vector is performed. Comparing these results to the results of the closed loop simulation the difference

appears conspicious. Multiple possible reasons could already be excluded. So the reason for the deviation is still an issue to be declared in the future.



(a) Temperature over time

(b) Actuating signals

(c) Remaining error considering x at steady-state

Figure 4.10: Experiment - step response of closed loop system using MPC

5 Conclusion and Outlook

This research aimed to apply the data-based system modeling concept Dynamic Mode Decomposition first to an illustrative example and further to a more complex process realized by a laboratory setting consisting of a silicon wafer heatable by LEDs and observed by a thermal camera. Two model-based controllers were implemented using the gained model of the process. Based on the good results in model validation as well as closed loop simulations and experiments it can be concluded that this approach of system modeling is convenient to model this distributed parameter system. The emdedded order reduction proves to be an essential feature especially when using the computational expensive Model Predictive Control. By manipulating the input data-sets of the Dynamic Mode Decomposition, this thesis has shown how to treat constant known external inputs and the mean of data-sets. The context of the contained spatial and temporal frequencies is pointed out. Also the input shape functions describing how every actuator of the setting acts onto the wafer surface was constructed using the model gained from Dynamic Mode Decomposition. This resulting estimation seems to match the reality quite well.

A qualitative comparison to a physically motivated model may be pointed out as a future aim considering the input shape functions. In general an analytic linear or even nonlinear model of the laboratory setting is of interest since the plant used in the simulations was generated by Dynamic Mode Decomposition until now. Discretizing the one dimensional heat diffusion equation in space and time in the cylindrical coordinate system followed by an identification experiment on the setting would be a convenient approach. Further the used plant can be extended by capturing the nonlinearity of the actuating signals. A more realistic plant will also result from taking into account the entire radial profile through the wafer. This captures the

homogeneous Neumann boundary condition in the center of the wafer. The setting itself can be improved by including a heating section in the radial center. This will improve the quality and possibilities in modeling and control. Choosing the parameters m and n of the data-set and the parameter r of the Dynamic Mode Decomposition in terms of [6] also may improve the resulting model. The entire method of combining Dynamic Mode Decomposition with Model Predictive Control can be implemented on a single wafer machine using a partition into smaller circular rings. This results into a higher number of actuators. The prediction approach enables to react onto known future process steps of the machine e.g. a fluid of certain temperature is spiked onto the rotating wafer. Further research may investigate if the MPC compensates this variation of the system better than a robust state controller.

Bibliography

[1] Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007 (cit. on p. 31).

[2] Travis Askham and J. Nathan Kutz. "Variable Projection Methods for an Optimized Dynamic Mode Decomposition." In: *SIAM Journal on Applied Dynamical Systems* 17.1 (2018), pp. 380–416. DOI: 10.1137/M1124176 (cit. on p. 4).

[3] Roscoe A Bartlett et al. "Quadratic programming algorithms for large-scale model predictive control." In: *Journal of Process Control* 12.7 (2002), pp. 775–795 (cit. on p. 40).

[4] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013 (cit. on p. 31).

[5] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. "Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses." In: *Journal of nonlinear science* 22.6 (2012), pp. 887–915 (cit. on pp. 13, 24).

[6] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank." In: *Psychometrika* 1.3 (1936), pp. 211–218 (cit. on pp. 6, 48).

[7] Jacob Grosek and J Nathan Kutz. "Dynamic mode decomposition for real-time background/foreground separation in video." In: *arXiv preprint arXiv:1404.7592* (2014) (cit. on pp. 2, 24).

[8] J. Nathan Kutz et al. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA, USA: SIAM-Society for Industrial and Applied Mathematics, 2016. ISBN: 1611974496 (cit. on pp. 1, 4).

[9] Qianxiao Li et al. "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.10 (2017), p. 103111 (cit. on p. 4).

[10] Qiugang Lu and Victor M. Zavala. "Image-based model predictive control via dynamic mode decomposition." In: *Journal of Process Control* 104 (2021), pp. 146–157. ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2021.06.009 (cit. on pp. 2, 37).

[11] Daiki Matsumoto and Thomas Indinger. "On-the-fly algorithm for dynamic mode decomposition using incremental singular value decomposition and total least squares." In: *arXiv preprint arXiv:1703.11004* (2017) (cit. on pp. 3, 4).

[12] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. "Dynamic Mode Decomposition with Control." In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161. DOI: 10.1137/15M1013857 (cit. on pp. 2, 5, 7, 28).

[13] Petr Vita. "Thin Film Fluid Flow Simulation on Rotating Discs." PhD thesis. May 2016. DOI: 10.13140/RG.2.2.20264.78089 (cit. on p. 4).

[14] Edward L Wilson and Robert E Nickell. "Application of the finite element method to heat conduction analysis." In: *Nuclear engineering and design* 4.3 (1966), pp. 276–286 (cit. on p. 1).