

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

THE APPLICATION OF PROPER ORTHOGONAL DECOMPOSITION TO THE CONTROL OF TUBULAR REACTORS

Promotor :
Prof. dr. ir. B. De Moor

Co-Promotors :
Prof. dr. ir. J. Espinosa
Prof. dr. ir. J. Vandewalle

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Oscar Mauricio AGUDELO

November 2009



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

THE APPLICATION OF PROPER ORTHOGONAL DECOMPOSITION TO THE CONTROL OF TUBULAR REACTORS

Jury:
Prof. dr. ir. Y. Willems, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. J. Espinosa, co-promotor (UNAL)
Prof. dr. ir. J. Vandewalle, co-promotor
Prof. dr. ir. J. Suykens
Prof. dr. ir. J. Van Impe
Prof. dr. ir. A.C.P.M. Backx (TU/e)
Prof. dr. ir. I. Smets

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Oscar Mauricio AGUDELO

November 2009

©Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 978-94-6018-133-7

D/2009/7515/116

To my mother Luz Marina, my greatest teacher

Foreword

This thesis is the product of my research activities during my doctoral studies at the SCD/SISTA research division of the Electrical Engineering Department of the Katholieke Universiteit Leuven. It has been an incredible journey of learning, full of challenges and nice experiences where I have not only grown as researcher but also as human being.

I want to thank my promotor Prof. Bart De Moor for his continuous support and guidance along my doctoral studies. I am also very grateful to my co-promotor Prof. Joos Vandewalle, who gave me the opportunity to join the research group as a predoctoral student. My gratitude goes as well to my co-promotor Prof. Jairo Espinosa, for his unconditional assistance, guidance and constant feedback since I was finishing my master studies in Ibagué, Colombia. I also want to thank Prof. Moritz Diehl and Dr. Michel Baes, for playing an important role during part of my research. I have to say that working together with them has been a very enriching experience for me. Many thanks to Prof. Johan Suykens, Prof. Jan Van Impe, Prof. Ton Backx and Prof. Ilse Smets for being part of the jury of this thesis. I would like to acknowledge the help that I received from the administrative staff, namely Ida Tassens and Ilse Pardon during all these years. I extend my gratitude to all my colleagues and friends, who made this journey a memorable and enjoyable experience.

Furthermore, I want to express my gratitude to all the people that in one way or another have made my stay in Belgium quite pleasant. Especially I want to thank Jairo Espinosa and his nice family, for considering me as another family member and for making me feel like at home. From the deepest part of my heart, Jairo, Claudia, Laurita, Jairito, Don Jairo and Doña alba, thank you to all of you. At the end of my second Ph.D year I had to undergo a knee surgery, and it was a quite tough moment for me. At that point I

have to say that the moral support of my family from Colombia and the help of Don Jairo in Belgium allowed me to go through. Many thanks Don Jairo for your unconditional support, hospitality and friendship.

Last but not least, I want to thank my mother Luz Marina, my father Luis Oscar, my brother Jorge Ernesto and my sister Carolina, for accompanying me from the distance in this journey, and for cheering me up in the most difficult times. *Gracias madre, por ser la luz que Dios puso a mi lado para no extraviar el camino.*

Oscar Mauricio Agudelo Mañozca

Leuven, November 2009.

Abstract

This dissertation considers two main research topics. First, this thesis explores the applicability of Proper Orthogonal Decomposition (POD) and Galerkin projection in the design of Model Predictive Control (MPC) schemes for tubular chemical reactors. These processes pose very interesting control problems, since their behavior is modeled by highly nonlinear Partial Differential Equations (PDEs), and they require the satisfaction of both their input (physical limitation of the actuators) and state constraints (e.g., the temperature inside the reactor must be below a given value in order to avoid the formation of byproducts). In this study, POD is used together with Garlerkin projection for reducing the high-dimensionality of the discretized systems used to approximate the PDEs that model the reactors. Then, based on the resulting reduced-order models, Kalman filters and predictive controllers are designed. Although a significant model order reduction can be obtained with POD and Galerkin projection, these techniques do not reduce the number of state constraints (linear inequality constraints) which is typically very large. In this thesis we propose two methods to tackle this problem. In the first method we use univariate polynomials to approximate part of the basis vectors derived with the POD technique, and then we apply the theory of positive polynomials to find good approximations of the state constraints by Linear Matrix Inequalities (LMIs). In the second method, we exploit the similarities between the coefficients of consecutive state constraints for developing a greedy algorithm that selects a small number of constraints from the complete set. This algorithm reduces dramatically the number of state constraints, and consequently the memory needed for storing them and the time required for solving the optimization problem.

The second main research subject of this thesis is related to speeding up the evaluation of reduced-order models derived by POD from nonlinear high-dimensional systems. Unlike the Linear Time Invariant (LTI) case,

the model-order reduction by POD and Galerkin projection does not conduce to an important computational saving when the high-dimensional models under consideration are nonlinear or Linear Time Variant (LTV). Therefore, this thesis introduces two methods for coping with this situation. The first method takes advantage of the input-output nonlinear mapping capabilities, and the fast on-line evaluation of Multi-Layer Perceptrons (MLPs) for accelerating the evaluation of the POD models. The second method exploits the polynomial nature of POD models derived from input-affine high-dimensional systems with polynomial nonlinearities, in order to generate compact and efficient formulations that can be evaluated much faster. Moreover, in this study it is shown how the use of sequential feature selection algorithms can provide a significant boost in the computational saving. Although this method is not as general as the first one, it might be applied to models with non-polynomial nonlinearities, provided that the nonlinearities can be approximated by low degree polynomials. In addition, conditions for guaranteeing the local stability of these POD models with polynomial nonlinearities are discussed.

Notation

Variables and Symbols

$\mathbf{x}^T$	Transpose of the vector $\mathbf{x}$
$\mathbf{A}^T$	Transpose of the matrix $\mathbf{A}$
$\mathbf{A}_{ij}$ or $\mathbf{A}(i,j), \mathbf{A} \in \mathbb{R}^{m \times n}$	Element at the ith row and jth column of $\mathbf{A}$
$\mathbf{A}(i,:), \mathbf{A} \in \mathbb{R}^{m \times n}$	ith row of $\mathbf{A}$
$\mathbf{A}(:,j), \mathbf{A} \in \mathbb{R}^{m \times n}$	jth column of $\mathbf{A}$
$\mathbf{A}^{\mathrm{H}}$	Conjugate transpose of the matrix $\mathbf{A}$
$\mathbf{I}_n$	Identity matrix of size $n \times n$
$\mathbf{I}$	Identity matrix
$\|\mathbf{x}\|_2,\ \mathbf{x} \in \mathbb{R}^n$	L_2-norm or Euclidean norm of a vector : $\sqrt{\mathbf{x}^T\mathbf{x}}$
$\|\mathbf{x}\|_{\mathbf{Q}},\ \mathbf{x} \in \mathbb{R}^n, \mathbf{Q} \in \mathbb{R}^{n \times n}$	Weighted norm: $\sqrt{\mathbf{x}^T\mathbf{Q}\mathbf{x}}$
$\langle \mathbf{x}, \mathbf{y} \rangle,\ \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$	Euclidean inner product between two vectors: $\mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x}$
$[\mathbf{x}; \mathbf{z}], \mathbf{x}, \mathbf{z} \in \mathbb{R}^n$	Stacked vectors : $[\mathbf{x}^T, \mathbf{z}^T]^T \in \mathbb{R}^{2n}$
$\{\mathbf{x}_i\}_{i=1}^p$	Data set with p elements : $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\}$
$\mathbf{1}$	Vector where all components are equal to 1

Acronyms

ARE	Algebraic Riccati Equation
BDS	Bidirectional Search
BMI	Bilinear Matrix Inequality
CTR	Continuous Tubular Reactor
CSTR	Continuous Stirred-Tank Reactor
KLD	Karhunen-Loève Decomposition
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
LTV	Linear Time Variant
LRS	Plus-L Minus-R Selection
MEMS	Micro-Electro-Mechanical System
MLP	Multi-Layer Perceptron
MPC	Model Predictive Control
MPE	Missing Point Estimation
MSE	Mean Squared Error
NSDP	Nonlinear SemiDefinite Program or Programming
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PFR	Plug Flow Reactor
POD	Proper Orthogonal Decomposition
POM	Proper Orthogonal Mode
POV	Proper Orthogonal Value
P-POD	Polynomial POD model
PRBNS	Pseudo Random Binary Noise Signals
PRMNS	Pseudo Random Multilevel Noise Signals
QP	Quadratic Programming or Program
RHC	Receding Horizon Control
SBS	Sequential Backward Selection
SDP	SemiDefinite Program or Programming
SFS	Sequential Forward Selection
SOS	Sum Of Squares
SQP	Sequential Quadratic Programming
SSE	Sum Squared Error
SVD	Singular Value Decomposition

Contents

Foreword **i**

Abstract **iii**

Notation **v**

Contents **vii**

1 General Introduction **1**

1.1 Introduction and motivation 1

1.2 Objectives 5

1.3 Chapter by chapter overview 5

1.4 Contribution of this thesis 9

2 Proper Orthogonal Decomposition and Predictive Control **13**

2.1 Introduction 13

2.2 Proper orthogonal decomposition 14

2.2.1 General procedure 15

2.2.2 Model reduction 18

2.3 Model predictive control 21

2.3.1 Predictive control principle 22
2.3.2 Estimation of the states 24
2.4 Example: Temperature control in a one–dimensional bar . . . 27
2.4.1 Heat transfer in a one-dimensional bar 27
2.4.2 Discretization 28
2.4.3 Model reduction using POD 30
2.4.4 MPC control scheme without a disturbance model . . 33
2.4.5 MPC control scheme with a disturbance model 38
2.5 Conclusions 42

3 Control of a Non-isothermal Tubular Reactor 45

3.1 Introduction 45
3.2 Tubular chemical reactor 46
3.2.1 Plug flow reactor model 47
3.2.2 Operating profiles 49
3.2.3 Linear model 53
3.3 Model reduction using POD 58
3.4 Predictive control schemes 65
3.4.1 First MPC control scheme (MPC-NTC) - Formulation in terms of the POD coefficients 65
3.4.2 Second MPC control scheme (MPC-PV) - Formulation in terms of physical variables 67
3.5 Simulation results 70
3.5.1 Tests on a reactor with axial dispersion 83
3.6 Conclusions 91

4 Constraint Handling 95

4.1 Introduction 95

4.2 POD-based MPC controller with temperature constraints . . 96

4.3 Positive polynomial approach 99

4.3.1 Fundamentals 99

4.3.2 Approximation of the temperature constraints by means of positive polynomials 102

4.3.3 Semidefinite representability of positive polynomials on an interval 104

4.3.4 Simulation results 108

4.4 Greedy selection algorithm 117

4.4.1 Simulation results 123

4.5 Conclusions 128

5 Performance Improvement in Model Simulation 131

5.1 Introduction 131

5.2 Nonlinear heat transfer in a one-dimensional bar 133

5.3 Nonlinear POD model of the system 135

5.4 Acceleration of POD models by using neural networks 139

5.5 Polynomial POD models 145

5.5.1 Calculation of the coefficients of the polynomials . . . 147

5.5.2 Reduction of the number of monomials 149

5.6 Polynomial POD models with stability guarantee 152

5.6.1 Semidefinite problem formulation 153

5.6.2 Nonlinear semidefinite problem formulation 154

5.6.3 Numerical example 155

5.7 Validation and simulation results 160
5.8 Conclusions 167

6 General Conclusions **169**
6.1 Concluding remarks 169
6.2 Future research 172

Bibliography **175**

Curriculum Vitae **187**

Publications by the author **189**

Chapter 1

General Introduction

1.1 Introduction and motivation

This thesis explores the application of a technique known as Proper Orthogonal Decomposition (POD), in the design of Model Predictive Control (MPC) strategies for tubular chemical reactors. Additionally, this dissertation develops new methods for improving the performance in simulation of models derived by POD from nonlinear high-dimensional systems.

Tubular chemical reactors are distributed parameter systems that typically are modeled by coupled nonlinear Partial Differential Equations (PDEs) which are derived from mass and energy balance principles. One way of addressing the control of these infinite-dimensional systems is by approximating the PDEs by a large number of Ordinary Differential Equations (ODEs). Afterwards, given the high-dimensionality of the resulting systems, reduced order models are derived to make possible the control design. Figure 1.1 shows this general control design framework. In this dissertation, the reduced order models are found by means of POD and Galerkin projection. Proper orthogonal decomposition is a data driven technique where a suitable set of orthonormal basis vectors are computed from simulation or experimental data. These basis vectors, which are organized in order of relevance, capture the spatial dynamics of the original systems. The reduced order models are obtained by projecting (Galerkin Projection) the high-dimensional models on the space spanned by the most relevant basis

vectors. The advantage of using these two techniques is the incorporation of simulated or experimental data as well as the existing physical relationships from the original model [64].

Model predictive control is a popular control method for handling input and state constraints within an optimal control setting. In MPC, the control actions are obtained by solving continually, on-line, a finite-horizon constrained open-loop optimal control problem. The popularity of this approach resides largely in its ability to handle, among other issues, multivariable interactions, constraints on controls and states, and optimization requirements. The use of this control strategy in tubular reactors is of special interest since this control methodology has demonstrated that it can push the plants towards their limits of performance while satisfying both the input (constraints in the actuators) and the state constraints (e.g., the temperature inside the reactor must be within a predefined range).

Tubular reactors typically operate under steady state conditions in order to efficiently produce high product volumes of a consistent quality. Nevertheless, transient operation regimes are also used to minimize the off-spec material during transitions, when reactors are employed for producing different kind of products. In this dissertation, the POD-based MPC controllers have the goal of rejecting the disturbances that affect the nominal operation of the reactors, under steady state regimes. For a complete literature review about model based control and optimization of tubular reactors, readers are referred to [95, page 43].

From the studies presented in [33–36], a general and practical framework for robust control synthesis for transport reaction process, which also encompass tubular reactors, has been established. However, the drawback of this framework is that it does not include the input and output constraints of the process under consideration. Consequently, the research efforts have been recently focused on the use of model predictive control strategies, which are characterized by dealing with the input and state constraints of a process in a very natural way. Thus, predictive controllers have been devised in [46, 133] for hyperbolic systems (convection-reaction processes, e.g., a tubular reactor where a plug-flow behavior is assumed), and in [44–46] for parabolic systems (Diffusion-reaction processes, e.g., a tubular reactor with axial diffusion/dispersion).

The well-known success of POD in many applications for deriving reduced-order models for simulation and control purposes, motivates its use in

this thesis for the development of alternative predictive control strategies (POD-based predictive control systems) for tubular chemical reactors. As it was mentioned before, these predictive control schemes should push the reactors to their limits of performance while satisfying the input and output constraints.

There is however, an important aspect of the POD-based predictive controllers that should be addressed at the moment of their implementation: the reduction of the number of state/output constraints which typically is very large, since it is given by the number of discretization points multiplied by the prediction horizon of the controllers. This large set of constraints consumes a significant amount of memory due to the large size of the matrices storing it, which by the way are not sparse. Furthermore, this large amount of constraints increase the computational time required for solving the optimization problem of the MPC. Clearly, methods that can cope with the reduction of the number of state/output constraints are necessary in order to generate more efficient POD-based predictive controllers.

Leaving aside the topic of POD-based predictive controllers for tubular reactors, another problem that motivates this thesis is the one related with the performance improvement in simulation of nonlinear reduced-order models derived by means of POD and Galerkin projection. Although a large model-order reduction can be achieved with these techniques, such reduction does not lead to a significant computational saving when nonlinear or Linear Time Variant (LTV) models are considered. This limitation is due to the necessity of having the full spatial information from the original high-dimensional systems, at the moment of evaluating the reduced-order models. In [10–12] a general method known as Missing Point Estimation (MPE) is introduced for coping with this problem. The method achieves a computational saving by conducting the Galerkin projection on some pre-selected state variables or points of the spatial domain instead of the complete set. The remaining state variables are estimated from the POD basis vectors. Although it has been reported that this technique can save considerable computation effort, the speeding up of nonlinear POD models is still an open problem that might be addressed from a different angle. Methods that exploit the nature of the nonlinearities, although more specific than the MPE, might provide more accurate reduced-order models that can be evaluated much faster.

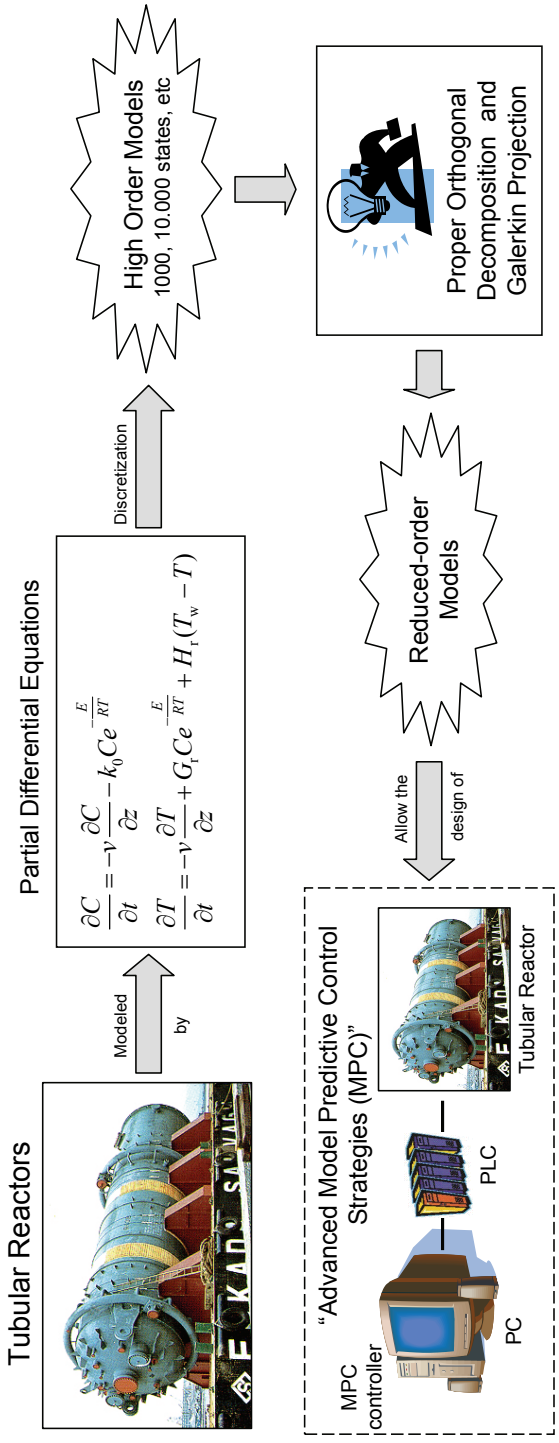


Figure 1.1: Control design framework.

1.2 Objectives

The main objectives of this dissertation are summarized in the following lines.

- To explore the applicability of proper orthogonal decomposition in the design of model predictive control schemes for tubular chemical reactors.
- To propose methods for reducing the number of state/output constraints of POD-based model predictive controllers.
- To derive alternative techniques for speeding up the evaluation of nonlinear or Linear Time Variant (LTV) POD models.

1.3 Chapter by chapter overview

This thesis is organized in 6 chapters. Figure 1.2 presents an overview of them as well as the way they relate to each other.

A brief description of each chapter is given as follows.

- **Chapter 2:** This chapter starts by introducing the fundamentals of proper orthogonal decomposition. Subsequently, the chapter describes how POD and Galerkin projection are used for deriving reduced order models from high-dimensional systems. The basics of model predictive control and Kalman filtering are also given here. The chapter concludes with a detailed example, where all these techniques are used together in the control of the temperature profile of a one-dimensional bar.
- **Chapter 3:** This chapter addresses the control of a non-isothermal tubular chemical reactor by using POD and predictive control techniques. After describing the system to be controlled, this chapter introduces an optimization algorithm for deriving the operating profiles in steady state of the reactor. The algorithm is described in detail and some numerical results are presented. Based on the POD model obtained from the linearized equations of the system

around the operating profiles, two MPC control schemes are proposed for the reactor. Their control goal is to keep the process at the operating profiles despite the disturbances in the feed flow, while maintaining the temperature inside the reactor below a given value in order to prevent undesirable side reactions. The basic difference between the MPC schemes is in their formulations. One of them is formulated in terms of the POD coefficients (MPC-NTC) and the other one in terms of physical variables (MPC-PV). In addition, the second one incorporates in its formulation the temperature constraint of the reactor and imposes it to some selected points of the spatial domain. This scheme also incorporates a mechanism for handling the possible infeasibilities that can arise. At the end of this chapter, some simulation results are presented in addition to a detailed comparison regarding the performance along several tests of the proposed control schemes. The pros and cons of each control system are also discussed.

- **Chapter 4:** This chapter starts by presenting an extension of the MPC-NTC controller proposed in the previous chapter. The new controller incorporates in its formulation the temperature constraint of the reactor and uses a slack variable approach with ℓ_∞-norm and time-dependent weights for dealing with the infeasibilities that might emerge [68]. Since POD only reduces the number of states and not the number of temperature constraints which is very large, the optimization algorithm within the MPC requires a considerable amount of memory and it also demands more computational effort for finding the optimal solution. In this chapter, two methods for reducing the number of temperature constraints are proposed. In the first method, the large set of inequality constraints (temperature constraints) is approximated by using the theory of positive polynomials [1]. This approximation conduces to a reduction in the number of constraints by replacing the large number of inequality constraints by a few linear matrix inequalities and a small number of linear equalities. The basics of this positive polynomials theory are also discussed very briefly. In the second method, a greedy algorithm is used for selecting a reduced set of constraints from the full set [5]. The algorithm exploits the similarities between the coefficients of consecutive temperature constraints, which tend to be alike as consequence of the smoothness of the most relevant basis vectors. Here it is shown that the greedy algorithm can be used for finding a suitable set of points for the MPC-PV controller proposed in the previous chapter. In addition,

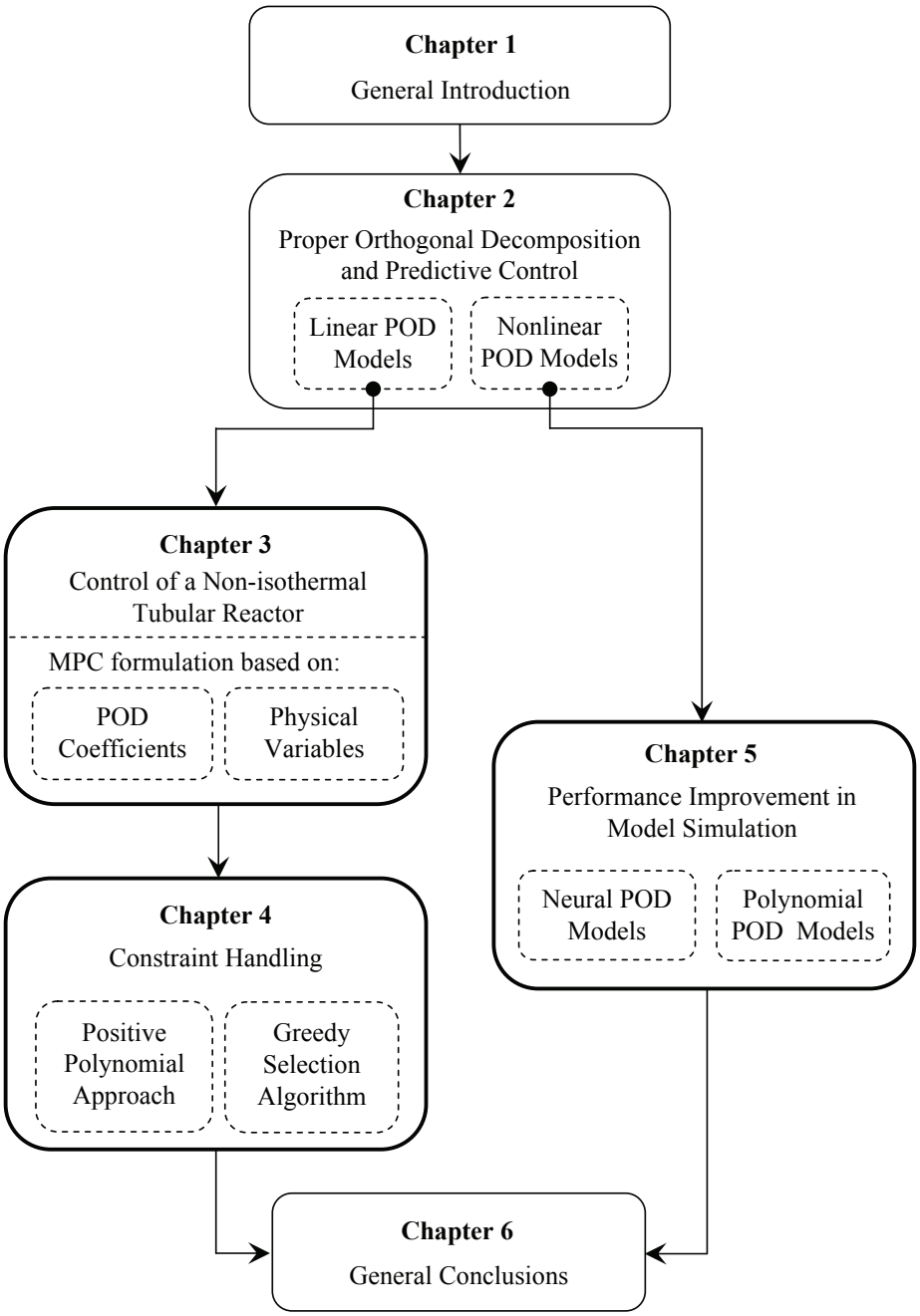


Figure 1.2: Overview and connection between the different chapters in this thesis. The arrows suggest the reading order of the chapters.

an improved formulation of this controller is discussed. Based on the polynomial approximation of the temperature constraints, and based on the reduced set of constraints found by the greedy algorithm, two new MPC controllers are presented in this chapter. The chapter includes several simulation results of all the control schemes as well as a discussion about their performance and the advantages and disadvantages of the methods proposed for reducing the number of constraints.

- **Chapter 5:** This chapter presents two methods for speeding up the evaluation of nonlinear POD models, which typically do not provide a significant computational gain with respect to the high-dimensional systems from which they are derived. This limitation comes from the fact that the vector function of the resulting POD models is still in terms of the high-dimensional vector function of the original models. In the first method proposed to tackle this problem, a multilayer perceptron is employed for approximating the nonlinear vector function of the reduced order models [7]. Provided that the output of a trained multilayer perceptron can be computed in a very short time, a significant computational saving can be expected. The second method is mostly intended for accelerating nonlinear POD models derived from input-affine high-dimensional systems with polynomial nonlinearities. It is shown that by taking advantage of the polynomial nature of the resulting POD models, a compact and efficient representation of the nonlinear vector function can be built, which significantly reduces the time required for evaluating the POD models. Given that the number of monomials of these polynomial representations can be very large and could compromise the computational saving, in this chapter a sequential feature selection algorithm is employed for selecting the most relevant monomials (suboptimal solution) in order to boost the computational gain. In order to guarantee the local stability of POD models with polynomial nonlinearities, an eigenvalue constraint is derived from Lyapunov's theory. Given that the inclusion of this constraint conduces to a non-smooth and non-convex optimization problem, in this chapter two approaches are proposed for dealing with this difficulty. In one method a semidefinite optimization problem has to be solved whereas in the second one the solution of a nonlinear semidefinite optimization problem must be found. The pros and cons of each of these approaches are discussed through a numerical example. In order to explain the

techniques introduced in this chapter for speeding up the evaluation of nonlinear POD models, the nonlinear heat transfer problem in a one-dimensional bar is used (it has nonlinearities of polynomial type). The last part of the chapter presents the simulation and validation results of the different POD models developed for the bar as well as a detailed discussion about the advantages and disadvantages of the techniques proposed. These techniques are also compared with an existing method known as missing point estimation [11, 12].

- **Chapter 6:** In this chapter the general conclusions of this dissertation are presented as well as some future research subjects.

1.4 Contribution of this thesis

The main contributions of this dissertation can be summarized as follows.

- In Chapter 2 by using a didactic and illustrative example, namely, the control of the temperature profile in a one-dimensional bar [2], we present a tutorial about the application of POD and Galerkin projection in the derivation of reduced-order models, which are the basis for developing predictive control schemes for high-dimensional processes, like the ones resulting from the discretization of partial differential equations.

- In Section 3.2.2 we propose an optimization algorithm (a sort of Sequential Quadratic Programming solver) for deriving the steady state operating profiles of a non-isothermal tubular reactor where a plug flow behavior is assumed [3]. The optimization problem solved by the algorithm, considers both the input and state constraints of the process and its cost function takes into account both the squared deviations of the concentration at the reactor outlet with respect to zero (terminal cost), and the squared deviations of the temperature along the reactor regarding the temperature of the feed flow (integral cost). To sum up, the proposed algorithm solves a multi-objective optimization where two conflictive objectives, the terminal and integral costs, are combined by a weighted sum in the cost function.

- Along Chapters 3 and 4, several POD-based predictive control schemes are developed for the tubular reactor considered in this dissertation.

A list of them is given as follows.

- MPC-NTC (see Section 3.4.1): MPC controller whose formulation is in terms of the POD coefficients and does not incorporate the temperature constraint of the reactor in its formulation [3].
- MPC-PV (check Section 3.4.2): MPC controller whose formulation is in terms of physical variables. It imposes the temperature constraint of the system on some selected points of the spatial domain [4].
- MPC-QP (see Section 4.2): This is an extension of the MPC-NTC controller where the temperature constraint of the reactor is taken into account. It is characterized for dealing with a very large number of linear inequality constraints [1].
- MPC-SDP (given by (4.10) in Section 4.3.3): This is a variation of the MPC-QP controller in which the large set of linear inequality constraints is replaced by few Linear Matrix Inequalities (LMIs) and equality constraints [1].
- MPC-QP-RS (defined by (4.13) in Section 4.4): This controller is also an adaptation of the MPC-QP controller in which the large set of linear inequality constraints is substituted by a reduced set of inequalities that has been found by the greedy selection algorithm introduced in Section 4.4 [5].

In addition, at the end of Section 4.4, we discuss an improved formulation of the MPC-PV controller.

- In Chapter 4, we propose two techniques for reducing the number of state/output constraints of POD-based predictive controllers, which typically is quite large. In our first approach, we use the theory of positive polynomials for approximating the feasible region delimited by the state/output constraints of the process [1]. This approximation leads to a reduction in the number of constraints by substituting many inequalities by a small number of LMIs and a few equality constraints. In our second approach, we exploit the fact that the coefficients of consecutive constraints are similar in order to formulate a greedy algorithm which chooses a reduced set of constraints from the complete set [5]. These techniques are applied to some of the predictive controllers proposed for the reactor treated in this dissertation.

- In Chapter 5, we introduce two methods for accelerating the evaluation of nonlinear POD models, given that their computational gain is

compromised by the fact that they need the original full spatial information of the high-dimensional models from which they are calculated. The first method takes advantage of the input-output nonlinear mapping capabilities, and the fast on-line evaluation of multilayer perceptrons for speeding up the evaluation of the POD models. From the approaches proposed, this is the most general one. Our second method is characterized by exploiting the polynomial nature of the POD models derived from input-affine high-dimensional systems with polynomial nonlinearities, in order to generate compact and efficient formulations that can be evaluated much faster. Although this method is not as general as the first one, it might be applied to models with non-polynomial nonlinearities, provided that the nonlinearities can be approximated by low degree polynomials.

- In Section 5.5.2, we show how to use the sequential feature selection theory for obtaining an extra boost in the computational gain of POD models with polynomial nonlinearities.

- Based on Lypaunov's theory, in Section 5.6 we propose constrained optimization problems that guarantee the local stability of the resulting polynomial POD models.

Chapter 2

Proper Orthogonal Decomposition and Predictive Control

2.1 Introduction

This chapter is dedicated to introducing the reader to Proper Orthogonal Decomposition (POD), Galerkin projection, Model Predictive Control (MPC) and Kalman Filtering. We will use these techniques along this thesis to design control schemes for processes described by Partial Differential Equations (PDEs) like tubular chemical reactors, for example.

The general procedure is as follows; first we discretize the PDE or PDEs modeling the process, usually this leads to a high-dimensional system that is not adequate for control design. Therefore we use POD and Galerkin projection for deriving a reduced order model that can be used in the design of MPC control schemes. Typically the state of the reduced order model is not measured as well as the disturbances that affect the process, and this information is required in the on-line implementation of the MPC algorithm. Hence, we use a Kalman filter (optimal estimation techniques) for estimating the unknown variables. This filter is also based on the reduced order model of the high-dimensional system.

The chapter is structured as follows. Section 2.2 introduces the proper

orthogonal decomposition technique and shows how it can be used in conjunction with Galerkin Projection for deriving reduced order models of high-dimensional systems. In Section 2.3 the basis of model predictive control are presented as well as the fundamentals of the Kalman filter. Finally we conclude this chapter with a very didactic and detailed example where we apply all the techniques presented in previous sections to the control of the temperature profile of a one-dimensional bar.

2.2 Proper orthogonal decomposition

Proper orthogonal decomposition and Galerkin projection are two well-known techniques that have been used together for deriving reduced order models of high-dimensional systems. These high-dimensional systems are typically obtained after discretizing in space the partial differential equations that model many processes. In this method an orthonormal basis for modal decomposition is extracted from an ensemble of data (called snapshots) obtained in the course of experiments or numerical simulations [93, 134]. The basis functions calculated with the POD technique are commonly called either empirical eigenfunctions, empirical basis functions, empirical orthogonal functions, Proper Orthogonal Modes (POMs) or basis vectors [91]. The POD method not only provides an orthonormal basis, but also a measure of the importance of each basis vector. This measure of importance is sometimes referred to as Proper Orthogonal Value (POV) [93]. Now, if we select the most relevant basis vectors and project (Galerkin projection) the original high-dimensional model on the space spanned by this subset, then we can obtain a reduced order model of the process. The most striking feature of the POD method is its optimality: it provides the most efficient way of capturing the dominant components of an infinite-dimensional process with only a finite number of "modes", and often surprisingly few "modes" [66].

Depending on the field of application, POD is also known by other names such as Karhunen-Loève Decomposition (KLD) or Expansion [94], Principal Component Analysis (PCA) [74], and Singular Value Decomposition (SVD) [9, 32] among others. POD has been developed by several people [91]. Lumley [98] traced the idea of POD back to independent investigations by Kosambi (1943) [82], Loeve (1945), Karhunen (1946), Pougachev (1953) and Obukhov (1954). Nevertheless, if we consider the history of PCA and

SVD, then we can not forget the work of Pearson who introduced PCA in 1901 [118], and we have to mention the contributions of Beltrami (1873) [19], Jordan (1874) [75, 76], Sylvester (1889) [141–143], Schmidt (1907) [130] and Weyl (1912) [154], who were responsible for establishing the existence of the singular value decomposition and developing its theory [139]. POD has been applied successfully in many engineering fields. It has been widely used in studies of turbulence [22, 23, 31, 79, 97, 134, 136], and also has been used in vibration analysis [37, 51], data analysis or compression as in characterization of human faces [80, 135], damage detection [129], signal analysis, map generation by robots [110], process identification, control in chemical engineering [1, 3–5, 78, 100, 146], model reduction of microelectromechanical systems (MEMS) [92], etc. There have been applications of POD to both optimization [47, 99, 100, 146] and feedback control design [1–5,12,17,64,71,72,78,84,85]. Besides in [14,15], a method for reducing controllers for systems described by PDEs using POD is discussed. A list of additional examples regarding the application of POD can be found in [23, 66]. Concerning the PDEs to which POD has been applied, we have among others: the incompressible/compressible Navier-Stokes equations [16, 59, 73, 121], the heat equation (Parabolic PDE) [2, 11, 26, 64, 156], the Burgers equation [31, 85], the wave equation (Hiperbolic PDE) [11], the Boussinesq equation [49], and the Helmholtz equation (Eliptic PDE) [151].

In general, POD can be interpreted or realized in three different ways, namely, Karhunen-Loève Decomposition (KLD), Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) [91, 93]. In this thesis the POD technique will be interpreted as an application of SVD. The reader is referred to [91] for a detailed discussion about the equivalence of the SVD, PCA and KLD interpretations of POD as well as their particular characteristics.

2.2.1 General procedure

Let $\mathbf{x}(t) \in \mathbb{R}^N = [x_1(t), x_2(t), \ldots, x_N]^T$ be the state vector of a given dynamical system, and let $\mathbf{X} \in \mathbb{R}^{N \times N_d}$ with $N_d \geq N$ be the so-called snapshot matrix

$$\mathbf{X} = [\mathbf{x}(t_1), \mathbf{x}(t_2), \ldots, \mathbf{x}(t_{N_d})] = \begin{bmatrix} x_1(t_1) & x_1(t_2) & \cdots & x_1(t_{N_d}) \\ x_2(t_1) & x_2(t_2) & \cdots & x_2(t_{N_d}) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(t_1) & x_N(t_2) & \cdots & x_N(t_{N_d}) \end{bmatrix}$$

containing a finite number of samples or snapshots of the evolution of $\mathbf{x}(t)$ at $t = t_1, t_2, \ldots, t_{N_d}$. In POD we start by observing that each snapshot can be written as a linear combination of a set of ordered orthonormal basis vectors (POD basis vectors) $\boldsymbol{\varphi}_j \in \mathbb{R}^N$, $\forall j = 1, 2, \ldots, N$:

$$\mathbf{x}(t_i) = \sum_{j=1}^{N} a_j(t_i)\boldsymbol{\varphi}_j, \ \forall i = 1, 2, \ldots, N_d \tag{2.1}$$

$$a_j(t_i) = \left\langle \mathbf{x}(t_i), \boldsymbol{\varphi}_j \right\rangle = \boldsymbol{\varphi}_j^T \mathbf{x}(t_i), \ \forall j = 1, 2, \ldots, N,$$

where $a_j(t_i)$ is the coordinate of $\mathbf{x}(t_i)$ with respect to the basis vector $\boldsymbol{\varphi}_j$ (it is also called time-varying coefficient or POD coefficient) and $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. Since the first n most relevant basis vectors capture most of the energy in the data collected, we can construct an nth order approximation of the snapshots by means of the following truncated sequence

$$\mathbf{x}_n(t_i) = \sum_{j=1}^{n} a_j(t_i)\boldsymbol{\varphi}_j, \ \forall i = 1, 2, \ldots, N_d, \quad n \ll N. \tag{2.2}$$

In POD, the orthonormal basis vectors are calculated in such a way that the reconstruction of the snapshots using the first n most relevant basis vectors is optimal in the sense that the Sum Squared Error (SSE) E_n between $\mathbf{x}(t_i)$ and $\mathbf{x}_n(t_i)$, $\forall i = 1, \ldots, N_d$,

$$E_n = \sum_{i=1}^{N_d} \|\mathbf{x}(t_i) - \mathbf{x}_n(t_i)\|_2^2 \tag{2.3}$$

is minimized. Herein $\|\cdot\|_2$ denotes the L_2-norm or Euclidean Norm. In other words, the POD basis vectors are the ones that solve the following constrained optimization problem:

$$\min_{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_n} \sum_{i=1}^{N_d} \left\| \mathbf{x}(t_i) - \sum_{j=1}^{n} \left\langle \mathbf{x}(t_i), \boldsymbol{\varphi}_j \right\rangle \boldsymbol{\varphi}_j \right\|_2^2 \tag{2.4}$$

subject to

$$\varphi_i^T \varphi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

The constraint in (2.4) imposes the orthonormality condition of the basis vectors. The orthonormal basis vectors that solve (2.4) can be found by

calculating the singular value decomposition of the snapshot matrix $\mathbf{X}$ (see [85, 91] for details about the derivation of the solution). If we write (2.1) using a matrix formulation,

$$\underbrace{[\mathbf{x}(t_1), \ldots, \mathbf{x}(t_{N_d})]}_{\mathbf{X}} = \underbrace{[\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_N]}_{\boldsymbol{\Phi} \in \mathbb{R}^{N \times N}} \overbrace{\begin{bmatrix} a_1(t_1) & \cdots & a_1(t_{N_d}) \\ a_2(t_1) & \cdots & a_2(t_{N_d}) \\ \vdots & \ddots & \vdots \\ a_N(t_1) & \cdots & a_N(t_{N_d}) \end{bmatrix}}^{\boldsymbol{\Gamma} \in \mathbb{R}^{N \times N_d}} \tag{2.5}$$

$$\mathbf{X} = \boldsymbol{\Phi}\boldsymbol{\Gamma}, \quad \boldsymbol{\Phi}^T\boldsymbol{\Phi} = \mathbf{I}_N.$$

then we obtain the proper orthogonal decomposition of $\mathbf{X}$ [9]. The matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ which contain the orthonormal basis vectors and the evolution of the POD coefficients respectively, are found from the SVD of the snapshot matrix $\mathbf{X}$ that is given by

$$\mathbf{X} = \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^T$$

where $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_N] \in \mathbb{R}^{N \times N}$ and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_{N_d}] \in \mathbb{R}^{N_d \times N_d}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N_d}$ is a matrix which contains the singular values $\sigma_i, \forall i = 1, 2, \ldots, N$ of $\mathbf{X}$ in a decreasing order on its main diagonal. The matrix $\boldsymbol{\Gamma}$ containing the evolution of the POD coefficients is then equal to the matrix product between $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}^T$. The orthonormal POD basis vectors are just the left singular vectors of $\mathbf{X}$. The minimum value of the SSE is given by the following summation,

$$E_n = \sum_{j=n+1}^{N} \sigma_j^2. \tag{2.6}$$

The singular values of $\mathbf{X}$ are positive real numbers that are ordered in a decreasing way, $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_N \geq 0$. These values quantify the importance of the basis vectors in capturing the information present in the data. Therefore, the first POD basis vector is the most relevant one and last POD basis vector is the least important element.

For the application of POD to concrete problems, the choice of the n most relevant basis vectors is certainly of central importance. A criterion commonly used for choosing n based on heuristic considerations is the so-called energy criterion [48]. In this criterion we check the ratio between the modeled energy and the total energy contained in $\mathbf{X}$,

$$\bar{P}_n = \frac{\sum_{j=1}^{n} \sigma_j^2}{\sum_{j=1}^{N} \sigma_j^2}, \quad n = 1, \ldots, N. \tag{2.7}$$

The ratio $\bar{P}_n$ is used to determine the truncation degree of the selected POD basis vectors. The number of POD basis elements should be chosen such that the fraction of the first singular values in (2.7) is large enough to capture most of the information in the data [11]. An ad-hoc rule often applied is that n has to be determined for $\bar{P}_n = 0.99$ [66]. The closer $\bar{P}_n$ to 1, or similarly the closer $1 - \bar{P}_n$ to 0, the better the approximation of $\mathbf{X}$ will be.

Given that the POD basis vectors only reflect the information provided by the snapshots, the generation of the snapshot matrix $\mathbf{X}$ is of vital importance in the model reduction process by using POD. We have to keep in mind that this technique attempts to capture the spatial dynamics (typically the state vector $\mathbf{x}(t)$ comes from the discretization in space of a PDE) of a system via the POD basis vectors and the temporal dynamics through the POD coefficients. So, we must try to get representative data of the process around the operating conditions on which we want to find its reduced order model.

2.2.2 Model reduction

For explaining the ideas and procedures in this section, we are going to suppose that the dynamical behavior of the high-dimensional system from which we want to find a reduced order model, is described by the following nonlinear model in state space form,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) &= \mathbf{g}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \end{aligned} \tag{2.8}$$

where $\mathbf{x}(t) \in \mathbb{R}^N$ is the state vector which acts as a memory containing all the information about the past of the system that is necessary to predict the future behavior, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the vector containing the outputs of the system, and $\mathbf{f}$ and $\mathbf{g}$ are vector-valued functions or maps of appropriate dimensions. The order of (2.8) is given by the number of state variables, that is, N.

Model reduction aims to approximate (2.8) by a lower complexity model, that is, a model with less number of states and therefore less state equations. When POD is used for this purpose, we can basically distinguish two steps:

- The derivation and selection of the n most relevant basis vectors $[\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_n]$ from an ensemble of simulation or experimental data (time snapshots) of the process described by (2.8) and,
- The derivation of the dynamical model for the POD coefficients $a_j(t), \forall j = 1, 2, \ldots, n$ associated to the selected basis vectors. The POD coefficients would be the states of the reduced order model.

It should be clear that the magnitude of the model-order reduction depends on the difference between the number of selected basis vectors and the order of the high-dimensional process. As it was explained in the previous section, the derivation of the basis vectors is performed by calculating the SVD of an ensemble of data called the snapshot matrix $\mathbf{X}$ and the selection of the most important basis vectors is carried out through the energy criterion. Notice that the reduced order model would exist in the low-dimensional space spanned by the selected POD basis vectors.

The derivation of the dynamical model for the POD coefficients can be done in two ways, by using the Galerkin projection or by means of system identification techniques. For the system identification case, we have to postulate a model structure for the relation between the process inputs $\mathbf{u}(t)$ and the POD coefficients $a_j(t), \forall j = 1, 2, \ldots, n$ and determine the unknown parameters in this model based on the data sets $\{\mathbf{u}(t_k)\}_{k=1}^{N_d}$ and $\{a_1(t_k), a_2(t_k), \ldots, a_n(t_k)\}_{k=1}^{N_d}$. The data set $\{\mathbf{u}(t_k)\}_{k=1}^{N_d}$ contains the inputs that were applied to the process in the generation of the snapshot matrix $\mathbf{X}$. The data set of the POD coefficients is nothing else than the first n rows of the matrix $\boldsymbol{\Gamma} = \boldsymbol{\Sigma}\Psi^T$. Notice that this data set can also be generated by using this relation: $a_j(t_i) = \langle \mathbf{x}(t_i), \boldsymbol{\varphi}_j \rangle = \boldsymbol{\varphi}_j^T \mathbf{x}(t_i),\ \forall j = 1, 2, \ldots, n$, and $\forall i = 1, 2, \ldots, N_d$. Once the unknown model parameters are estimated, a reduced order model is available that can predict the time evolution of the POD coefficients from a given time trajectory of the process input $\mathbf{u}(t)$. In [70–72] subspace identification techniques [113] are used together with POD in the derivation of a reduced order model of an industrial glass feeder.

The Galerkin projection [11, 16, 24, 73, 90, 121] is the most common way of deriving the dynamical model for the POD coefficients, and it will be the

method used in this thesis.

Let us define a residual function $R(\dot{\mathbf{x}}, \mathbf{x})$ for Equation (2.8) as follows:

$$R(\dot{\mathbf{x}}, \mathbf{x}) = \dot{\mathbf{x}}(t) - \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right), \tag{2.9}$$

and let $R(\dot{\mathbf{x}}_n, \mathbf{x}_n)$ be the residual when the state vector $\mathbf{x}(t)$ is approximated by its nth order approximation

$$\mathbf{x}_n(t) = \sum_{j=1}^{n} a_j(t)\boldsymbol{\varphi}_j = \boldsymbol{\Phi}_n \mathbf{a}(t), \quad n \ll N$$

where $\boldsymbol{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_n]$ and $\mathbf{a}(t) = [a_1(t), a_2(t), \ldots, a_n(t)]^T$. In the Galerkin projection, the projection of the residual $R(\dot{\mathbf{x}}_n, \mathbf{x}_n)$ on the space spanned by the basis vectors $\boldsymbol{\Phi}_n$ vanishes, that is,

$$\left\langle R(\dot{\mathbf{x}}_n, \mathbf{x}_n), \boldsymbol{\varphi}_j \right\rangle = 0, \quad \forall j = 1, 2, \ldots, n, \tag{2.10}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. This means that the residual $R(\dot{\mathbf{x}}_n, \mathbf{x}_n)$ is not correlated to $\boldsymbol{\varphi}_j, \forall j = 1, 2, \ldots, n$ at all. Moreover, the orthogonality of the residual to the span of the basis vectors implies that the residual is minimal [11]. Therefore, in order to find the model for the POD coefficients, we replace $\mathbf{x}(t)$ by its nth order approximation $\mathbf{x}_n(t) = \boldsymbol{\Phi}_n \mathbf{a}(t)$ in the state equation of (2.8),

$$\boldsymbol{\Phi}_n \dot{\mathbf{a}}(t) = \mathbf{f}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right)$$

and then we apply the inner product criterion (2.10) as follows,

$$\begin{aligned} \left\langle \boldsymbol{\Phi}_n \dot{\mathbf{a}}(t), \boldsymbol{\varphi}_j \right\rangle &= \left\langle \mathbf{f}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right), \boldsymbol{\varphi}_j \right\rangle, \quad \forall j = 1, 2, \ldots, n \\ \boldsymbol{\Phi}_n^T \boldsymbol{\Phi}_n \dot{\mathbf{a}}(t) &= \boldsymbol{\Phi}_n^T \mathbf{f}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right) \end{aligned}$$

and given that $\boldsymbol{\Phi}_n^T \boldsymbol{\Phi}_n = \mathbf{I}_n$ because of the orthonormality of the basis vectors, we have that the model for the POD coefficients reduces to

$$\dot{\mathbf{a}}(t) = \boldsymbol{\Phi}_n^T \mathbf{f}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right).$$

Finally, the reduced order model of (2.10) with only n states has the following form,

$$\begin{aligned} \dot{\mathbf{a}}(t) &= \boldsymbol{\Phi}_n^T \mathbf{f}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) &= \mathbf{g}\left(\boldsymbol{\Phi}_n \mathbf{a}(t), \mathbf{u}(t)\right). \end{aligned} \tag{2.11}$$

We can use this reduced order model for control design purposes or for carrying out faster simulations.

2.3 Model predictive control

Model Predictive Control (MPC), also referred to as Receding Horizon Control (RHC) or moving horizon control, is a control strategy where a finite horizon open-loop optimal control problem is solved on-line at each sampling time using the current state of the plant as the initial state, in order to get a sequence of future control actions from which only the first one is applied to the plant. The fact of solving on-line an optimization problem where commonly plant constraints are included, makes MPC different from conventional control which uses a pre-computed control law [104]. MPC has been widely adopted by the industrial process control community and implemented successfully in many applications. Several reasons are attributed to this success [145]. First of all, the MPC algorithms can handle in a very natural way constraints on both process inputs (manipulated variables or control actions) and process outputs values (controlled variables), which often have a significant impact on the quality, effectiveness and safety of the production. Additionally, the MPC controllers can take into account the internal interactions within the process, thanks to the multivariable models on which they are typically based. This make the MPC algorithms a quite suitable option for multivariable process control. Another reason of the success of MPC is the fact that the principle of operation is comprehensible and relatively easy to explain to process operators and engineers. This is an important aspect at the moment of introducing new techniques into industrial practice.

MPC was originally developed to meet the specialized control needs of power plants and petroleum refineries, and its application was first reported in the seventies [38, 125]. Nowadays, the MPC technology can be found in a wide variety of application areas including chemicals, food processing, automotive, and aerospace applications. A recent survey that provides an overview of commercially available model predictive control technology can be found in [120]. Several past reviews regarding theoretical and practical aspects of MPC are offered in [56, 89, 103, 104, 107, 109, 123, 126].

Linear MPC refers to a family of MPC schemes in which linear models are used to predict the system dynamics, even though the dynamics of the closed-loop system are nonlinear due to the presence of constraints. Along this thesis we will deal with MPC controllers based on discrete-time Linear Time Invariant (LTI) models in state space form:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{2.12}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

where $\mathbf{x}(k) \in \mathbb{R}^{n_x}$, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ and $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ are the state, input and output vectors respectively and $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ are the matrices defining the system dynamics.

In the next subsection we will present very briefly the principle of operation of an MPC controller and the formulation that will be used in this thesis.

2.3.1 Predictive control principle

The predictive control principle is as follows. Based on the measurement or estimation of the state $\mathbf{x}(k)$ of the process at time k, the controller predicts the future dynamic behavior of the plant $\{\mathbf{x}(k+1), \mathbf{x}(k+2), \ldots, \mathbf{x}(k+N_\mathrm{p})\}$ over a prediction horizon N_p, and determines (over a control horizon $N_\mathrm{c} \leq N_\mathrm{p}$) a sequence of future control actions $\{\mathbf{u}(k), \mathbf{u}(k+1), \ldots, \mathbf{u}(k+N_\mathrm{c}-1)\}$ such that a predetermined open-loop performance objective function J is optimized. Then only the first element of this sequence is applied to the plant. At the next sampling time $(k+1)$ a new measurement or estimation of the state is obtained and the whole procedure is repeated, with the prediction and control horizons of the same length N_p and N_c but shifted by one step forward. This is known as the principle of Receding Horizon Control (RHC) and it is depicted in Figure 2.1. It is important to remark that the future control actions are calculated assuming that $\mathbf{u}(k+N_\mathrm{c}-1) = \mathbf{u}(k+N_\mathrm{c}) = \cdots = \mathbf{u}(k+N_\mathrm{p}-1)$. Typically, the prediction horizon is set in such a way that the difference between the prediction and control horizons is at least equal to the largest settling time of the process. This criterion is commonly used in industry for guaranteeing the stability of the closed-loop system when the process to be controlled is stable.

For tracking problems, an MPC controller typically tries to minimize the following performance objective function J at each time instant k,

$$J = \sum_{i=1}^{N_\mathrm{p}} \|\mathbf{x}_\mathrm{ref}(k+i) - \mathbf{x}(k+i)\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_\mathrm{c}-1} \|\Delta\mathbf{u}(k+i)\|_{\mathbf{R}}^2 \tag{2.13}$$

subject to the model (2.12) of the plant and the input and state constraints of the process. Here $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x} \succeq 0$ and $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$ are positive semidefinite and definite weighting matrices, $\|\mathbf{v}\|_{\mathbf{Q}}^2$ denotes $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, $\mathbf{x}_\mathrm{ref}(k+i)$ is the reference vector of $\mathbf{x}(k+i)$ and $\Delta\mathbf{u}(k+i) = \mathbf{u}(k+i) - \mathbf{u}(k+i-1)$.

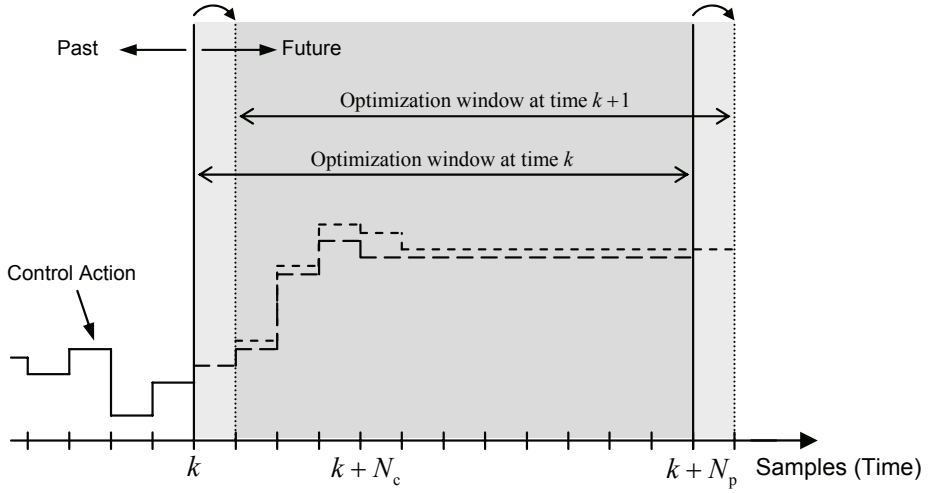


Figure 2.1: The principle of Receding Horizon Control (RHC). At each time instant k an optimal control sequence is calculated after which only the first element of such a sequence is applied to the plant.

Notice that in the cost function J we use $\Delta\mathbf{u}(k)$ instead of $\mathbf{u}(k)$. This is necessary for having an integral action in the controller that guarantees an offset free tracking [128]. The optimization problem that is solved by the MPC controller at each sampling time k is then formally defined as follows:

$$\min_{\mathbf{x}_{N_\mathrm{p}},\Delta\mathbf{u}_{N_\mathrm{c}}} \sum_{i=1}^{N_\mathrm{p}} \|\mathbf{x}_{\mathrm{ref}}(k+i) - \mathbf{x}(k+i)\|^2_{\mathbf{Q}} + \sum_{i=0}^{N_\mathrm{c}-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}} \tag{2.14a}$$

subject to

$$\mathbf{x}(k+i+1) = \mathbf{A}\mathbf{x}(k+i) + \mathbf{B}\mathbf{u}(k+i), \qquad i = 0,\ldots,N_\mathrm{p}-1, \tag{2.14b}$$
$$\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), \qquad i = 0,\ldots,N_\mathrm{c}-1, \tag{2.14c}$$
$$\mathbf{u}(k+i) = \mathbf{u}(k+i-1), \qquad i = N_\mathrm{c},\ldots,N_\mathrm{p}-1, \tag{2.14d}$$
$$\mathbf{x}(k+i) \in \mathcal{X}, \qquad i = 1,\ldots,N_\mathrm{p}, \tag{2.14e}$$
$$\mathbf{u}(k+i) \in \mathcal{U}, \qquad i = 0,\ldots,N_\mathrm{c}-1, \tag{2.14f}$$

with

$$\mathbf{x}_{N_\mathrm{p}} = [\mathbf{x}(k+1); \mathbf{x}(k+2); \ldots; \mathbf{x}(k+N_\mathrm{p})] \tag{2.15}$$
$$\Delta\mathbf{u}_{N_\mathrm{c}} = [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_\mathrm{c}-1)] \tag{2.16}$$

where (2.14e) and (2.14f) are the state and input constraints and $\mathcal{X}$ and $\mathcal{U}$ are convex sets. Notice that by using (2.12), constraints on the outputs can always be rewritten as state constraints $\mathbf{C}\mathbf{x}(k) \in \mathcal{Y}, \forall k$, where $\mathcal{Y}$ is a convex set. A convex set is defined as follows [29]:

Definition 2.1. *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is convex iff for any two points $\mathbf{x}_1$,$\mathbf{x}_2 \in \mathcal{S}$ all convex combinations of these points also lie within the set $\mathcal{S}$:*

$$(1-\theta)\mathbf{x}_1 + \theta\mathbf{x}_2 \in \mathcal{S}, \qquad \forall\theta \in [0,1], \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}.$$

That is, $\mathcal{S}$ is a convex set if the straight line segment connecting any two points in $\mathcal{S}$ lies entirely in $\mathcal{S}$.

Particularly, if $\mathcal{X}$ and $\mathcal{U}$ are the feasible regions delimited by linear inequality constraints, and we express the cost function and the constraints of (2.14) in terms of $\Delta\mathbf{u}_{N_c}$ (condensed form of the MPC), then problem (2.14) can be written as a Quadratic Program (QP) in $\Delta\mathbf{u}_{N_c} \in \mathbb{R}^{n_u \cdot N_c}$ as follows:

$$\min_{\Delta\mathbf{u}_{N_c}} \frac{1}{2} (\Delta\mathbf{u}_{N_c})^T \mathbf{H} (\Delta\mathbf{u}_{N_c}) + \mathbf{f}_l^T \Delta\mathbf{u}_{N_c} \tag{2.17}$$

subject to

$$\mathbf{A}_{\text{ineq}} \Delta\mathbf{u}_{N_c} \leq \mathbf{b}_{\text{ineq}}$$

where $\mathbf{H} \in \mathbb{R}^{(n_u \cdot N_c)\times(n_u \cdot N_c)} \succ 0$ is the so-called Hessian matrix, $\mathbf{f}_l \in \mathbb{R}^{n_u \cdot N_c}$ is the vector accompanying the linear term, m is the number of linear inequality constraints, $\mathbf{A}_{\text{ineq}} \in \mathbb{R}^{m\times(n_u \cdot N_c)}$ is the matrix of the inequality constraint coefficients and $\mathbf{b}_{\text{ineq}} \in \mathbb{R}^m$ is the right hand side vector of the inequality constraints. See [29] for more information on Quadratic Programming and its history. By ensuring that $\mathbf{Q}$ and $\mathbf{R}$ in (2.14) are positive semi-definite and positive definite respectively, the positive definiteness of $\mathbf{H}$ is guaranteed, and therefore problem (2.17) is strictly convex. In (2.17) only the matrix $\mathbf{H}$ can be computed off-line. In contrast, the vector $\mathbf{f}_l$ has to be calculated at each time instant k, since it depends on the current measured/estimated state of the plant.

2.3.2 Estimation of the states

The MPC control algorithm described in the previous section requires having the current state of the plant for solving the optimization problem (2.17)

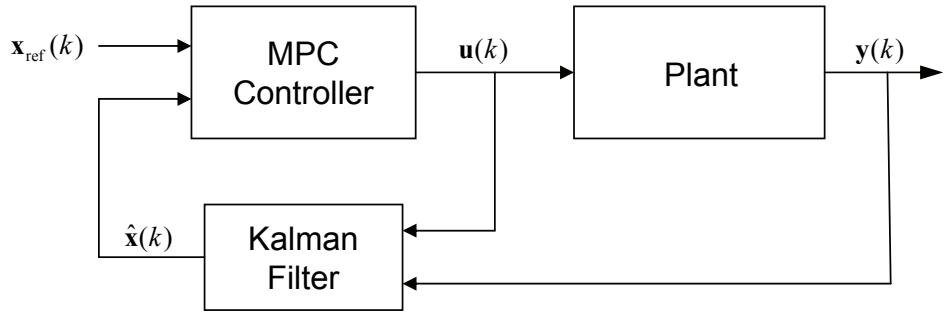


Figure 2.2: MPC control Scheme.

at each time instant k. However, in general, the entire state vector $\mathbf{x}(k)$ is not available. Therefore the use of an observer or soft-sensor is necessary in order to estimate the state vector of the plant from the process input values and the measured process outputs, on the basis of a mathematical model of the system. The estimation of the state vector $\mathbf{x}(k)$ will be denoted by $\hat{\mathbf{x}}(k)$. Figure 2.2 shows a typical MPC control loop. For the design of an observer it is assumed that the discrepancies between the model predictions and the measured process outputs are caused by errors in the initial values of the state variables, disturbances on the process state variables and disturbances on the measured process outputs. The equation of an observer (Luenberger observer [96]) includes the model of the plant (2.12) and an additional term that uses the error between the predicted outputs and the measured outputs for correcting the estimations of the state vector via a feedback gain matrix. This equation is given by

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\left(\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k) \end{aligned} \tag{2.18}$$

where $\mathbf{L} \in \mathbb{R}^{n_x \times n_u}$ is the feedback gain matrix or observer gain. The dynamics of the estimation error $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ is modelled by $\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{LC})\,\mathbf{e}(k)$ with $\mathbf{e}(0) = \mathbf{x}(0) - \hat{\mathbf{x}}(0)$. From this last equation it is clear that the estimation error will converge to zero when k goes to infinity, and that the velocity of this convergence is influenced by the observer gain $\mathbf{L}$. Here it was assumed that the observer gain has been chosen such that the observer is asymptotically stable, that is, the eigenvalues of $\mathbf{A} - \mathbf{LC}$ are inside the unit circle. The observer gain $\mathbf{L}$ can be found by the pole placement method [54] or by using optimal estimation theory. When $\mathbf{L}$ is calculated by means of optimal estimation techniques, the observer is

referred to as Kalman filter. The Kalman filter was introduced in the sixties by R. E Kalman [77] and it will be the observer used in this thesis. The fundamentals of this observer will be presented very briefly in the following lines.

Consider the following discrete-time model of the plant,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{G}\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \tag{2.19}$$

where $\mathbf{G} \in \mathbb{R}^{n_x \times n_w}$ is a weighting matrix and $\mathbf{w}(k) \in \mathbb{R}^{n_w}$ and $\mathbf{v}(k) \in \mathbb{R}^{n_y}$ are random variables that represent the process and measurement noises respectively. The process noise $\mathbf{w}(k)$ is modeled as a Gaussian white noise with zero mean and covariance matrix $\mathbf{R}_{\mathrm{w}} \in \mathbb{R}^{n_w \times n_w}$ and the measurement noise $\mathbf{v}(k)$ is modeled as a Gaussian white noise with zero mean and covariance matrix $\mathbf{R}_{\mathrm{v}} \in \mathbb{R}^{n_y \times n_y}$. These covariance matrices are defined by

$$\mathbf{R}_{\mathrm{w}} = \varepsilon\left\{\mathbf{w}(k)\mathbf{w}(k)^T\right\}, \tag{2.20a}$$

$$\mathbf{R}_{\mathrm{v}} = \varepsilon\left\{\mathbf{v}(k)\mathbf{v}(k)^T\right\}, \tag{2.20b}$$

where $\varepsilon\{\cdot\}$ denotes expectation. Additionally we have that $\varepsilon\left\{\mathbf{v}(k)\mathbf{w}(k)^T\right\} = 0$ and it is assumed that $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are not correlated with $\mathbf{x}(k)$ and $\mathbf{y}(k)$.

We are interested in finding an observer gain $\mathbf{L}$ such that the covariance of the estimation error $(\mathbf{x}(k) - \hat{\mathbf{x}}(k))$ given by

$$J_{\mathrm{obs}} = \frac{1}{2}\varepsilon\left\{\sum_{k=0}^{\infty}\left(\mathbf{x}(k) - \hat{\mathbf{x}}(k)\right)^T\left(\mathbf{x}(k) - \hat{\mathbf{x}}(k)\right)\right\}$$

is minimized. The solution of this optimization problem is given by the so-called Kalman Gain,

$$\mathbf{L} = \mathbf{A}\bar{\mathbf{Q}}\mathbf{C}^T\left(\mathbf{C}\bar{\mathbf{Q}}\mathbf{C}^T + \mathbf{R}_{\mathrm{v}}\right)^{-1} \tag{2.21}$$

where $\bar{\mathbf{Q}}$ is the covariance matrix of the steady-state estimation error that satisfies the so-called Algebraic Riccati Equation (ARE),

$$\bar{\mathbf{Q}} - \mathbf{A}\bar{\mathbf{Q}}\mathbf{A}^T + \mathbf{A}\bar{\mathbf{Q}}\mathbf{C}^T\left(\mathbf{R}_{\mathrm{v}} + \mathbf{C}\bar{\mathbf{Q}}\mathbf{C}^T\right)^{-1}\mathbf{C}\bar{\mathbf{Q}}\mathbf{A}^T - \mathbf{G}\mathbf{R}_{\mathrm{w}}\mathbf{G}^T = 0. \tag{2.22}$$

Readers interested in the derivation of the solution (2.21) are referred to [54, 86].

Finally, we want to stress that for the control of high-dimensional systems, the MPC controller and the Kalman Filter in Figure 2.2 will be designed from the reduced order model of the process obtained by means of the POD and Galerkin projection techniques.

2.4 Example: Temperature control in a one–dimensional bar

For illustration purposes, in this section we present the application of POD and predictive control techniques to the control of the temperature profile of a one-dimensional bar [2]. Initially an MPC controller without a disturbance model is designed. The control objective is to allow the bar to reach a desired temperature distribution in steady state as fast as possible, satisfying at the same time the process constraints. Afterwards, an MPC with a disturbance model is implemented in order to reject the perturbations that affect the bar. Both MPCs are based on the reduced order model of the system found by using POD and Galerkin projection.

2.4.1 Heat transfer in a one-dimensional bar

The system to be controlled is a "perfectly insulated heated bar" (see Figure 2.3), which has 3 inputs ($u_1(t)$,$u_2(t)$ and $u_3(t)$) and 2 measurement points ($y_1(t)$ and $y_2(t)$). It is important to underline that the inputs of the bar correspond to the boundary conditions of the PDE describing the system.

If only temperature variations in the $z-$direction are considered, the dynamics of the temperature $T(z,t)$ of the bar can be modeled by the following parabolic PDE :

$$G\frac{\partial^2 T(z,t)}{\partial z^2} = \frac{\partial T(z,t)}{\partial t} \tag{2.23}$$

with

$$G = \frac{\kappa}{\rho c_{\mathrm{p}}}$$

where ρ is the density of the bar in $[\mathrm{kg}\cdot\mathrm{m}^{-3}]$, κ is the thermal conductivity in $[\mathrm{J}\cdot\mathrm{s}^{-1}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}]$, c_{p} is the heat capacity in $[\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}^{-1}]$ and z is

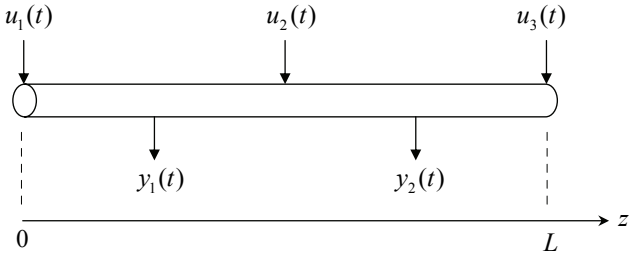


Figure 2.3: Heated bar. The signals $u_1(t)$, $u_2(t)$ and $u_3(t)$ are the boundary conditions at $z = 0$, $z = L/2$ and $z = L$. The measured outputs $y_1(t)$ and $y_2(t)$ are the temperatures of the bar at $z = L/4$ and $z = 3L/4$ respectively.

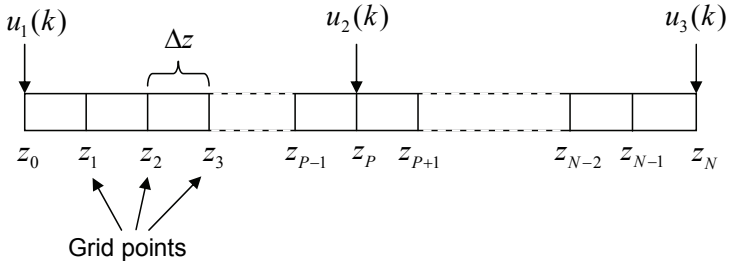


Figure 2.4: Spatial discretization of the bar.

the position in [m]. The initial and boundary conditions (Dirichlet type) of (2.23) are given by,

$$T(z, 0) = T_0(z), \tag{2.24a}$$

$$T(0, t) = u_1(t), \quad T(L/2, t) = u_2(t), \quad T(L, t) = u_3(t). \tag{2.24b}$$

The length of the bar is $L = 0.1$ m and the parameter G is equal to 10^{-5}. The initial temperature distribution is set to $T_0(z) = 0°\text{C}$ and the input signals $u_1(t)$, $u_2(t)$ and $u_3(t)$ must be between 0°C and 150°C (input constraints).

2.4.2 Discretization

For design and simulation purposes, Equation (2.23) is discretized in space (see Figure 2.4) and time by means of the "Implicit Backward Euler method" (a finite difference method), which unlike the "Explicit Forward Euler method", is unconditionally stable [155]. The stability condition of the

explicit forward Euler algorithm turns out to be $G\frac{\Delta t}{\Delta z^2} \leq \frac{1}{2}$. This implies that as we decrease the spatial interval Δz for better accuracy, we must also decrease the time step Δt at the cost of more computations in order not to lose the stability. In the backward Euler method, the second partial derivative with respect to z is replaced by a central difference approximation, and the time derivative by a backward difference approximation as follows:

$$G\left(\frac{T_{i+1}(k) - 2T_i(k) + T_{i-1}(k)}{\Delta z^2}\right) = \frac{T_i(k) - T_i(k-1)}{\Delta t} \tag{2.25}$$

$$\text{for } i = 1, 2, \ldots, P-1, P+1, \ldots, N-1$$

$$\text{for } k = 1, 2, \ldots, M$$

with

$$T_0(k) = u_1(k), \quad T_P(k) = u_2(k), \quad T_N(k) = u_3(k),$$

where N is the number of sections in which the bar is divided, Δz is the length of each section, Δt is the sampling time, $T_i(k) = T(z_i, t_k)$ is the temperature in the grid point $z_i = i\Delta z$ at the time $t_k = k\Delta t$, P is the grid point where $u_2(t)$ is located ($z = L/2$) and M is the number of time steps.

If we define $\mathbf{T}(k) \in \mathbb{R}^{N-2} = [T_1(k), \ldots, T_{P-1}(k), T_{P+1}(k), \ldots, T_{N-1}(k)]^T$ as the vector containing the temperatures of the grid points $z_i, \forall i = 1, 2, \ldots, P-1, P+1, \ldots, N-1$ at each time step k, Equation (2.25) can be cast into a recursive linear system of equations as follows:

$$\mathbf{AT}(k+1) = \mathbf{T}(k) + \mathbf{Bu}(k)$$

$$\mathbf{T}(k+1) = \mathbf{A}^{-1}\mathbf{T}(k) + \mathbf{A}^{-1}\mathbf{Bu}(k) \tag{2.26}$$

where $\mathbf{u}(k) = [u_1(k), u_2(k), u_3(k)]^T$ is the vector of inputs, and $\mathbf{A} \in \mathbb{R}^{(N-2)\times(N-2)}$ and $\mathbf{B} \in \mathbb{R}^{(N-2)\times 3}$ are the matrices describing the system defined as follows,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_s \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} r & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & r & r & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & r \end{bmatrix}^T,$$

with

$$\mathbf{A}_{\mathrm{s}} \in \mathbb{R}^{(P-1)\times(P-1)} = \begin{bmatrix} 1+2r & -r & 0 & \cdots & 0 \\ -r & 1+2r & \ddots & \ddots & \vdots \\ 0 & -r & \ddots & -r & 0 \\ \vdots & \ddots & \ddots & 1+2r & -r \\ 0 & \cdots & 0 & -r & 1+2r \end{bmatrix}, \; r = G\frac{\Delta t}{\Delta z^2} \, .$$

In this example the sampling time is set to 1 s, and the spatial domain is divided into $N = 400$ sections which means that Equation (2.26) has $N - 2 = 398$ states. This large number of states makes the design of feedback controllers for the bar difficult. Therefore, it is necessary to find a reduced order model. Such a model is derived in the following subsection by using POD and Galerkin projection.

2.4.3 Model reduction using POD

For deriving a reduced order model of (2.26), the subsequent steps were followed:

1. **Generation of the Snapshot Matrix**. We have constructed a snapshot matrix $\mathbf{T}_{\mathrm{snap}} \in \mathbb{R}^{398\times500}$ from the system response when Pseudo Random Binary Noise Signals (PRBNS) were applied simultaneously to the inputs $u_1(k)$, $u_2(k)$, and $u_3(k)$ of the discrete model of the bar (2.26),

$$\mathbf{T}_{\mathrm{snap}} = [\mathbf{T}(k=1), \mathbf{T}(k=2), \ldots, \mathbf{T}(k=500)] \, . \tag{2.27}$$

   Along the simulations, a switching probability of 2% and an amplitude of 100°C were set to the PRBNS signals, and 500 samples were collected using a sampling time of 1 s.

2. **Derivation of the POD basis vectors.** As it was explained in Section 2.2.1, the POD basis vectors are found by calculating the SVD of the snapshot matrix $\mathbf{T}_{\mathrm{snap}}$,

$$\mathbf{T}_{\mathrm{snap}} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$$

   where $\mathbf{\Phi} \in \mathbb{R}^{398\times398}$ and $\mathbf{\Psi} \in \mathbb{R}^{500\times500}$ are unitary matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{398\times500}$ is a matrix containing the singular values of $\mathbf{T}_{\mathrm{snap}}$ in a

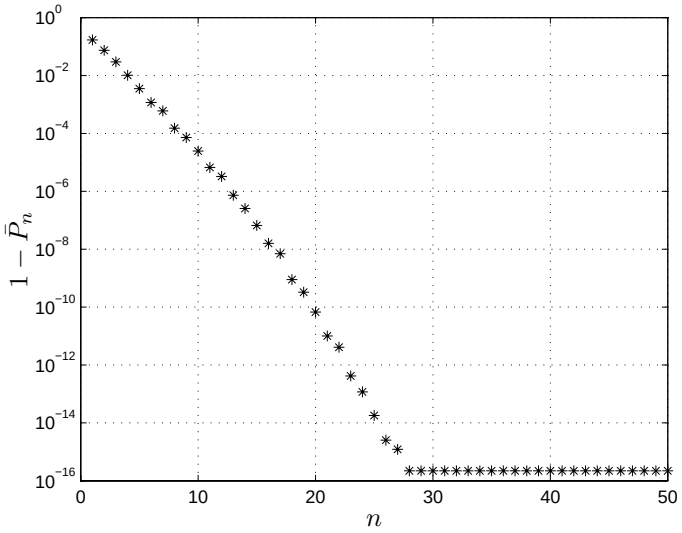


Figure 2.5: The logarithmic plot of $1 - \bar{P}_n$ which is used to determine the truncation degree of the POD basis vectors.

decreasing order on its main diagonal. The left singular vectors, that is, the columns of the matrix $\mathbf{\Phi}$,

$$\mathbf{\Phi} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{398}]$$

are the POD basis vectors.

3. **Selection of the most relevant POD basis vectors.** The n most relevant POD basis vectors are chosen using the energy criterion presented in Section 2.2.1. The plot of $1 - \bar{P}_n$ (see Equation (2.7)) for the first 50 basis vectors is shown in Figure 2.5. In this case, we selected the first $n = 10$ POD basis vectors (they are shown in Figure 2.6) based on their truncation degree $1 - \bar{P}_n = 2.454 \cdot 10^{-5}$. Thus, the 10th order approximation of $\mathbf{T}(k)$ is given by

$$\mathbf{T}_n(k) = \sum_{j=1}^{10} a_j(k)\boldsymbol{\varphi}_j = \mathbf{\Phi}_n \mathbf{a}(k) \tag{2.28}$$

where $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{10}]$ and $\mathbf{a}(k) = [a_1(k), a_2(k), \ldots, a_{10}(k)]^T$.

4. **Construction of the model for the first n=10 POD coefficients.** As it was explained in Section 2.2.2, the dynamic model for the

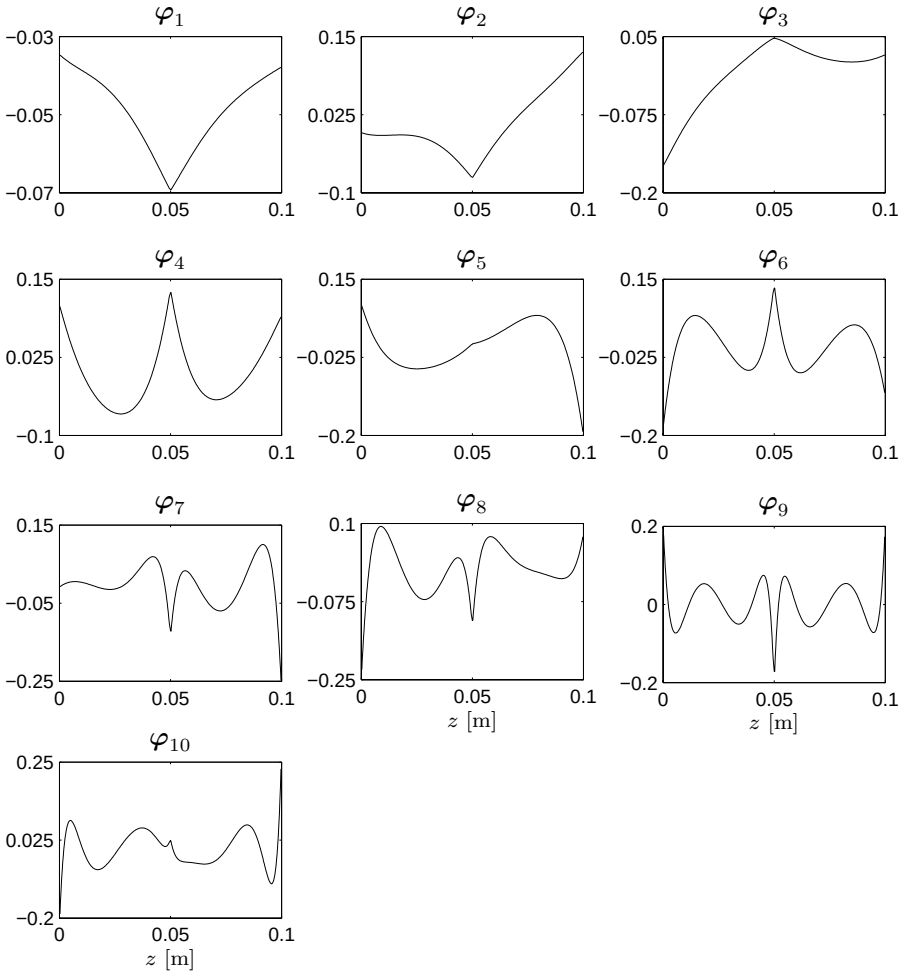


Figure 2.6: Selected POD basis vectors.

POD coefficients can be derived by projecting (Galerkin projection) the model (2.26) on the space spanned by the selected POD basis vectors $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{10}]$. If we replace $\mathbf{T}(k)$ by its nth order approximation $\mathbf{T}_n(k) = \mathbf{\Phi}_n \mathbf{a}(k)$ in Equation (2.26), and we apply the inner product criterion (Galerkin projection) to the resulting equation, we have that

$$\left\langle \mathbf{\Phi}_n \mathbf{a}(k+1), \boldsymbol{\varphi}_j \right\rangle = \left\langle \mathbf{A}^{-1} \mathbf{\Phi}_n \mathbf{a}(k) + \mathbf{A}^{-1} \mathbf{B} \mathbf{u}(k), \boldsymbol{\varphi}_j \right\rangle, \qquad (2.29)$$

$$\forall j = 1, 2, \ldots, n = 10.$$

By evaluating the inner product in (2.29),

$$\mathbf{\Phi}_n^T\mathbf{\Phi}_n\mathbf{a}(k+1) = \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{\Phi}_n\mathbf{a}(k) + \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{B}\mathbf{u}(k)$$

$$\mathbf{a}(k+1) = \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{\Phi}_n\mathbf{a}(k) + \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{B}\mathbf{u}(k) \tag{2.30}$$

we obtain the model for the first n POD coefficients. The reduced order model of the bar with only 10 states is then given by

$$\begin{aligned}\mathbf{a}(k+1) &= \tilde{\mathbf{A}}\mathbf{a}(k) + \tilde{\mathbf{B}}\mathbf{u}(k) \\ \mathbf{T}_n(k) &= \mathbf{\Phi}_n\mathbf{a}(k)\end{aligned} \tag{2.31}$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{10\times 10} = \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{\Phi}_n$ and $\tilde{\mathbf{B}} \in \mathbb{R}^{10\times 3} = \mathbf{\Phi}_n^T\mathbf{A}^{-1}\mathbf{B}$.

For validating the reduced order model, constant inputs $u_1(k) = 0°\text{C}$, $u_2(k) = 100°\text{C}$ and $u_3(k) = 50°\text{C}$ were applied to the full order model and to the reduced order model, and afterwards their outputs were compared. Figure 2.7 shows the temperature profile of the bar at the time steps $k = 1$, $k = 25$, $k = 50$ and $k = 250$ for each model. It is really difficult to observe differences between the responses of the models. Figure 2.8 presents the plot of the average of the absolute error which was calculated by means of the following formula:

$$\mathbf{E}_\text{T} = \frac{1}{N_\text{s}}\sum_{k=1}^{N_\text{s}} |\mathbf{T}(k) - \mathbf{T}_n(k)|$$

where $N_\text{s} = 250$ is the number of simulation time steps. The maximum peak in Figure 2.8 is only 0.198°C, which means that the reduced order model with only 10 states approximates very well the behavior of the full order model (398 states).

2.4.4 MPC control scheme without a disturbance model

The control goal is to allow the bar to reach a desired temperature distribution in steady state as fast as possible. In addition, the control actions must satisfy the input constraints of the process, that is, $0°\text{C} \leq u_1(k), u_2(k), u_3(k) \leq 150°\text{C}$. In the top plot of Figure 2.10, the desired temperature profile $\mathbf{T}_\text{ref} \in \mathbb{R}^{398}$ for the bar can be observed. This temperature profile corresponds to the steady state temperature distribution when the bar is heated from zero temperature by constant inputs $u_1(k) = 30°\text{C}$, $u_2(k) = 60°\text{C}$ and $u_3(k) = 10°\text{C}$. The control of the temperature profile

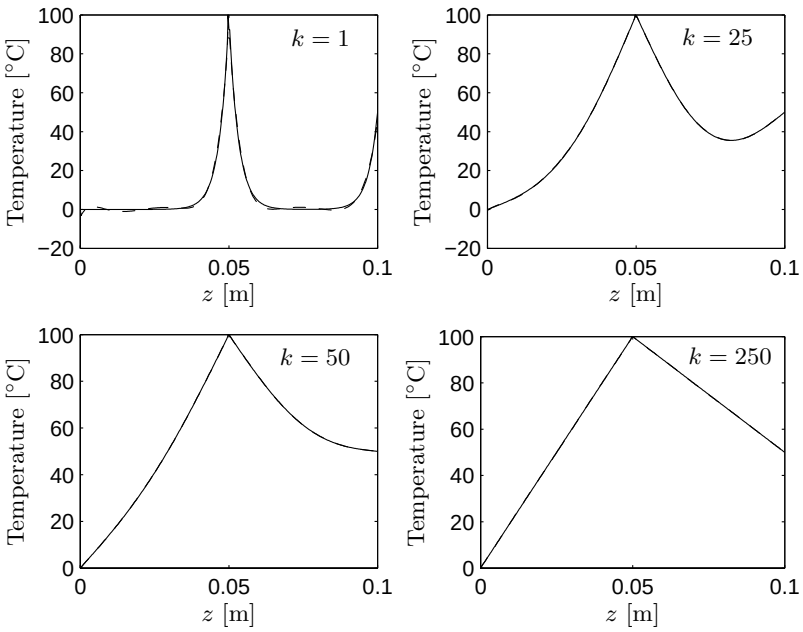


Figure 2.7: Temperature profile at different time steps. Solid line - Full order model (2.26). Dashed line - Reduced order model (2.31).

of the bar is achieved indirectly by controlling the POD coefficients. The references ($\mathbf{a}_{\text{ref}}$) of these POD coefficients can be calculated by

$$\mathbf{a}_{\text{ref}} = \mathbf{\Phi}_n^T \mathbf{T}_{\text{ref}}. \tag{2.32}$$

For controlling the first $n = 10$ POD coefficients and consequently the temperature profile of the bar, we initially implemented the MPC control scheme shown in Figure 2.9. In this scheme the MPC controller uses the reduced order model given by (2.31) to predict the future behavior of the process. An observer, which in this case is a Kalman filter, is used for estimating the state of the reduced order model from the measurements $\mathbf{y}(k) = [y_1(k), y_2(k)]^T$ and the process inputs $\mathbf{u}(k) = [u_1(k), u_2(k), u_3(k)]^T$. The observer equations are given by,

$$\begin{aligned} \hat{\mathbf{a}}(k+1) &= \tilde{\mathbf{A}}\hat{\mathbf{a}}(k) + \tilde{\mathbf{B}}\mathbf{u}(k) + \mathbf{L}\left(\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}_{\text{s}}\hat{\mathbf{T}}_n(k) = \mathbf{C}_{\text{s}}\mathbf{\Phi}_n\hat{\mathbf{a}}(k) \end{aligned} \tag{2.33}$$

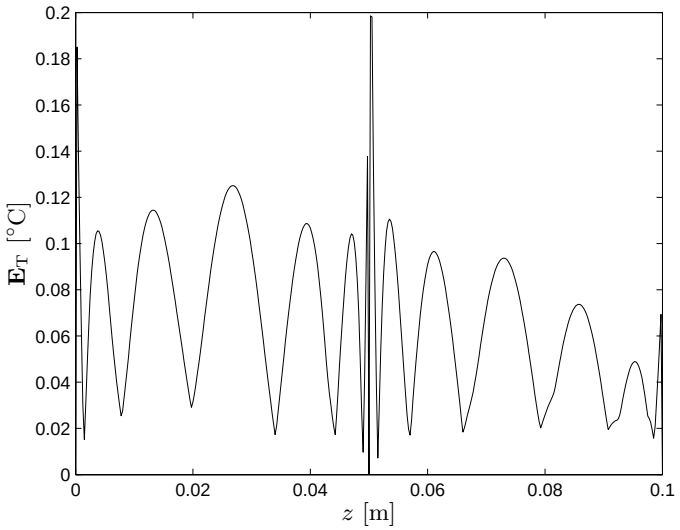


Figure 2.8: Average of the absolute error between the full order model and the reduced order model.

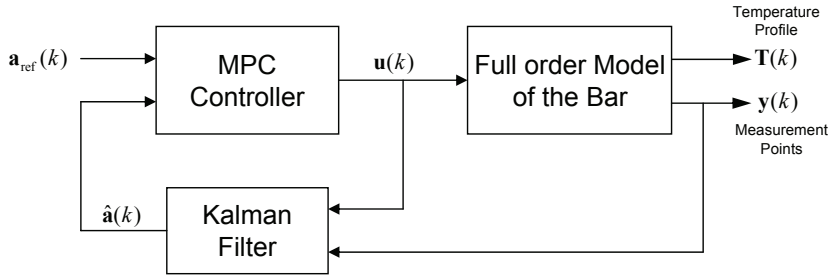


Figure 2.9: MPC control scheme without a disturbance model.

where $\hat{\mathbf{a}}(k)$ is the estimated vector of the POD coefficients, $\hat{\mathbf{y}}(k)$ and $\hat{\mathbf{T}}_n(k)$ are the estimations of the output vector $\mathbf{y}(k)$ and the nth order approximation of the temperature profile $\mathbf{T}_n(k)$ respectively, $\mathbf{L}$ is the observer gain (Kalman gain) and $\mathbf{C}_\mathrm{s}$ is a selection matrix which selects the measured temperatures ($y_1(k)$ and $y_2(k)$) from the vector $\hat{\mathbf{T}}_n(k)$. The Kalman gain was calculated from the following covariance matrices: $\mathbf{R}_\mathrm{v} = 10^{-6} \cdot \mathbf{I}_2$, $\mathbf{R}_\mathrm{w} = \mathbf{I}_{10}$. Here, $\mathbf{R}_\mathrm{v}$ is the covariance matrix of the measurement noise ($\mathbf{v}(k)$) and $\mathbf{R}_\mathrm{w}$ is the covariance matrix of the process noise ($\mathbf{w}(k)$). The diagonal of $\mathbf{R}_\mathrm{v}$ contains the measured noise variance of each output

signal which are assumed to be uncorrelated. For this example these values were assumed to be equal to 10^{-6}. Physically, $\mathbf{R}_{\mathrm{w}}$ tries to explain unknown disturbances, whether they are steps, white noise, or imperfections in the model of the plant. This parameter can be used to trade speed and robustness. In this case, $\mathbf{R}_{\mathrm{w}}$ was chosen to be equal to the identity matrix. The simulations results confirmed that it was an appropriated choice for calculating the observer gain.

The estimated state $\hat{\mathbf{a}}(k)$ is used together with the reference vector $\mathbf{a}_{\mathrm{ref}}$ by the MPC controller to calculate the appropriated control actions to be applied to the bar. The minimization problem that is solved by the MPC controller at each time instant k is defined as:

$$\min_{\mathbf{a}_{N_{\mathrm{p}}},\Delta\mathbf{u}_{N_{\mathrm{c}}}} \sum_{i=1}^{N_{\mathrm{p}}} \|\mathbf{a}_{\mathrm{ref}}(k+i) - \mathbf{a}(k+i)\|^2_{\mathbf{Q}} + \sum_{i=0}^{N_{\mathrm{c}}-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}} \tag{2.34}$$

subject to

$$\begin{aligned}
&\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i), && i = 0,\ldots,N_{\mathrm{p}}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0,\ldots,N_{\mathrm{c}}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_{\mathrm{c}},\ldots,N_{\mathrm{p}}-1,\\
&\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max}, && i = 0,\ldots,N_{\mathrm{c}}-1,
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{a}_{N_{\mathrm{p}}} &= [\mathbf{a}(k+1); \mathbf{a}(k+2); \ldots; \mathbf{a}(k+N_{\mathrm{p}})]\\
\Delta\mathbf{u}_{N_{\mathrm{c}}} &= [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_{\mathrm{c}}-1)]
\end{aligned}$$

where $\|\mathbf{v}\|^2_{\mathbf{Q}}$ denotes $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$, N_{p} is the prediction horizon, N_{c} is the control horizon, $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are weighting matrices, and $\mathbf{u}_{\min} \in \mathbb{R}^3$ and $\mathbf{u}_{\max} \in \mathbb{R}^3$ are the lower and upper bounds of $\mathbf{u}(k)$.

The control horizon N_{c} was set to 7 samples and the prediction horizon N_{p} was selected according to the following criterion: "Prediction Horizon = Control Horizon + Largest Settling Time = 80 samples". $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ were set according to the input constraints of the process, that is, $\mathbf{u}_{\min} = [0,0,0]^T$ and $\mathbf{u}_{\max} = [150,150,150]^T$, and the weighting matrices in this way: $\mathbf{Q} = \mathbf{I}_{10}$, $\mathbf{R} = \mathbf{I}_3$.

The simulation results are shown in Figures 2.10 and 2.11. In Figure 2.10 we can observe the steady state temperature profile of the bar when the MPC

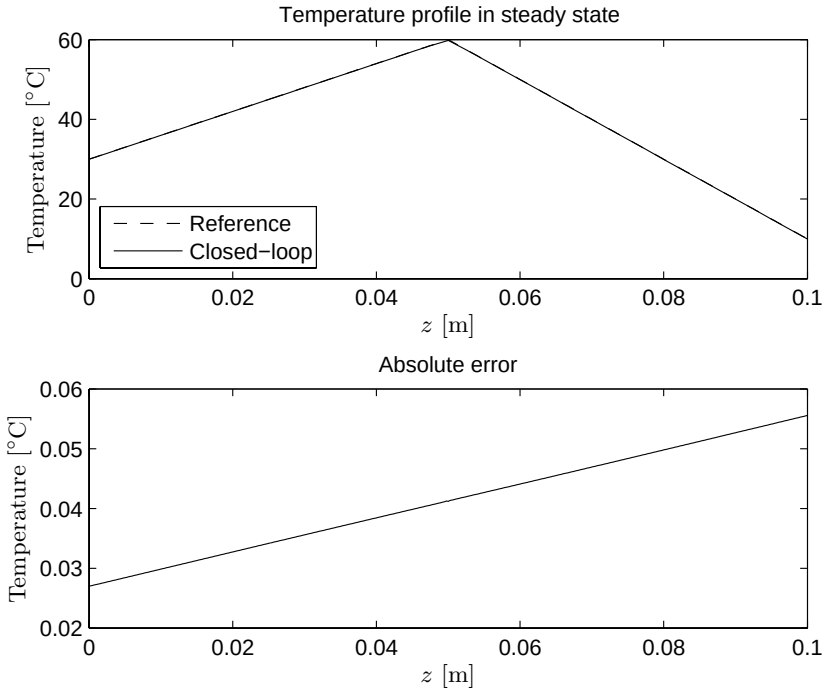


Figure 2.10: Steady-state temperature profile of the bar when the MPC controller (2.34) is used.

controller (2.34) is used. Also the absolute error between the reference and the steady state response of the closed loop system is shown. The maximum deviation in the temperature profile is only 0.055°C. Figure 2.11 presents the evolution in time of the temperature at the measurement points and the control actions of the MPC controller, which satisfy the input constraints of the process. The closed-loop responses reach steady state faster than the open loop responses because the control system pushes the inputs to the limits. In spite of the dramatic reduction of model order, on which the controller is based, the controller performs very well.

Now, we are going to suppose that the left edge of the bar is exposed to the ambient temperature (20°C) from a specific time instant. It implies that temperature at the left edge would be given by the ambient temperature plus $u_1(k)$. Figure 2.12 presents the steady state temperature profile of the bar in disturbed conditions with the MPC controller. It is clear that the behavior of the control system in the presence of input disturbances is not

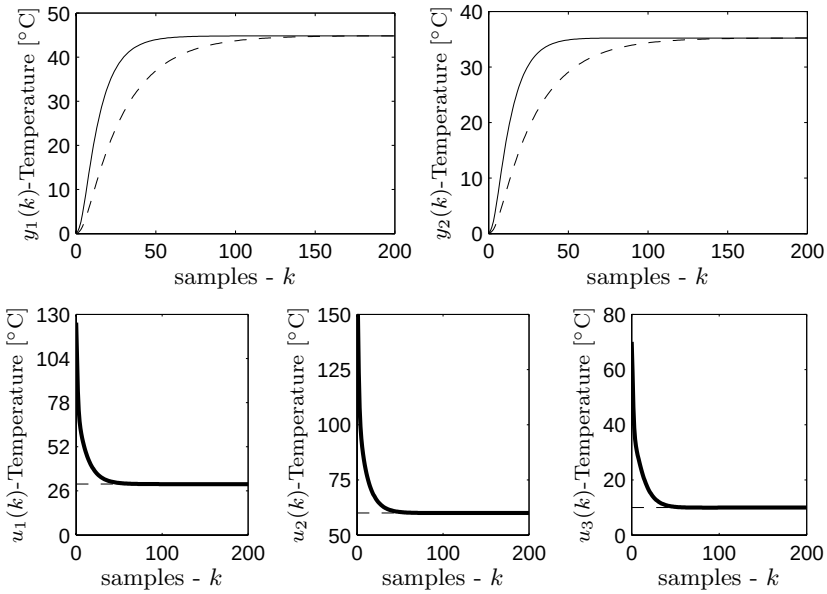


Figure 2.11: Temperature at the measurement points and control actions of the MPC controller (2.34). Solid line - Closed-loop response. Dashed line - Open loop response.

good. At the left edge of the bar there is a large temperature deviation of 14.22°C. This was an expected result, since the control scheme does not include a mechanism for rejecting such kind of disturbances. So, in order to overcome such limitation, an MPC controller with a disturbance model is designed and presented in the next subsection.

2.4.5 MPC control scheme with a disturbance model

The new MPC control scheme can be observed in Figure 2.13. The formulation of the MPC controller with a disturbance model is given by

$$\min_{\mathbf{a}_{N_\mathrm{p}},\Delta\mathbf{u}_{N_\mathrm{c}}} \sum_{i=1}^{N_\mathrm{p}} \|\mathbf{a}_\mathrm{ref}(k+i) - \mathbf{a}(k+i)\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_\mathrm{c}-1} \|\Delta\mathbf{u}(k+i)\|_{\mathbf{R}}^2 \qquad (2.35a)$$

subject to

$$\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \mathbf{F}d(k+i), \quad i = 0, \ldots, N_\mathrm{p} - 1,$$

$$
\begin{aligned}
&d(k+i+1) = d(k+i), && i = 0, \ldots, N_\mathrm{p} - 1, \qquad (2.35b)\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0, \ldots, N_\mathrm{c} - 1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_\mathrm{c}, \ldots, N_\mathrm{p} - 1,\\
&\mathbf{u}_\mathrm{min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_\mathrm{max}, && i = 0, \ldots, N_\mathrm{c} - 1,
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{a}_{N_\mathrm{p}} &= [\mathbf{a}(k+1); \mathbf{a}(k+2); \ldots; \mathbf{a}(k+N_\mathrm{p})]\\
\Delta\mathbf{u}_{N_\mathrm{c}} &= [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_\mathrm{c}-1)]
\end{aligned}
$$

where $\mathbf{F}$ is the disturbance model matrix, $d(k) \in \mathbb{R}$ is the disturbance signal and Equation (2.35b) is the disturbance equation, which in this case corresponds to a step. Since the disturbance $d(k)$ and the state $\mathbf{a}(k)$ are unknown, they must be estimated by means of a new Kalman filter with the following formulation:

$$
\begin{aligned}
\begin{bmatrix} \hat{\mathbf{a}}(k+1) \\ \hat{d}(k+1) \end{bmatrix} &= \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{F} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) + \qquad (2.36)\\
&+ \begin{bmatrix} \mathbf{L}_\mathrm{a} \\ \mathbf{L}_\mathrm{d} \end{bmatrix} (\mathbf{y}(k) - \hat{\mathbf{y}}(k))\\
\hat{\mathbf{y}}(k) &= \mathbf{C}_\mathrm{s} \hat{\mathbf{T}}_n(k) = \mathbf{C}_\mathrm{s} \mathbf{\Phi}_n \hat{\mathbf{a}}(k)
\end{aligned}
$$

where $\hat{\mathbf{a}}(k)$ is the estimated state, $\hat{d}(k)$ is the estimated disturbance vector, $\hat{\mathbf{y}}(k)$ is the estimated output vector, $\left[\mathbf{L}_\mathrm{a}^T, \mathbf{L}_\mathrm{d}^T\right]^T$ is the Kalman gain and $\mathbf{C}_\mathrm{s}$ is a selection matrix which selects the measured temperatures ($y_1(k)$ and $y_2(k)$) from the vector $\hat{\mathbf{T}}_n(k)$.

The number of disturbance terms that can be estimated without losing observability is equal to the number of sensors [114]. This means that in this problem we might estimate at most 2 disturbance signals. In the MPC formulation (2.35) the disturbance model is defined by the matrix $\mathbf{F}$. Since we are interested in rejecting the disturbance associated to the input $u_1(k)$ (left edge of the bar), the matrix $\mathbf{F}$ is chosen in the following way: $\mathbf{F} = \tilde{\mathbf{B}}(:, 1)$, where $\tilde{\mathbf{B}}(:, 1)$ denotes the first column of the matrix $\tilde{\mathbf{B}}$.

The control and prediction horizon were set to 7 samples and 80 samples respectively, $\mathbf{u}_\mathrm{min}$ and $\mathbf{u}_\mathrm{max}$ were selected according to the input constraints of the process and the weighting matrices in this way: $\mathbf{Q} = \mathbf{I}_{10}$, $\mathbf{R} = \mathbf{I}_3$.

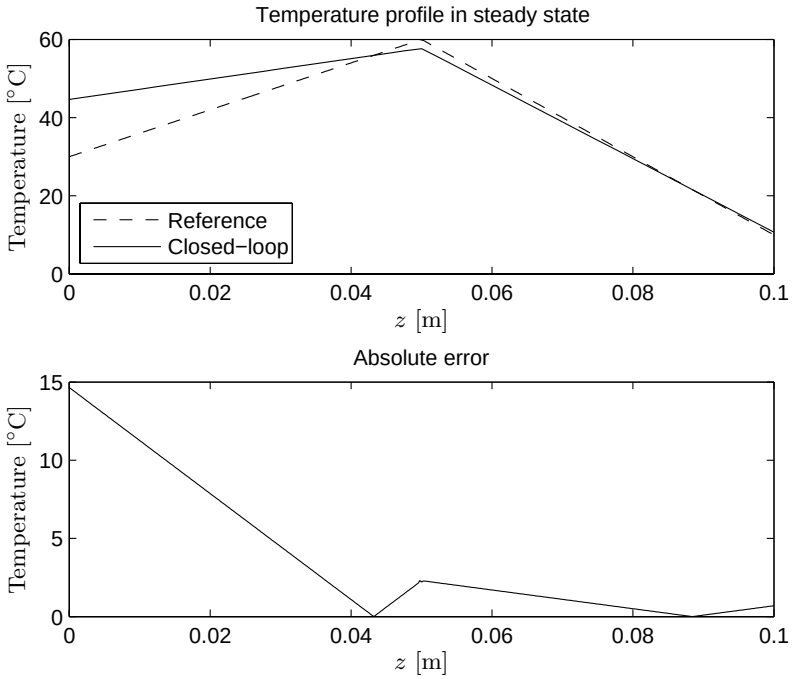


Figure 2.12: Steady-state temperature profile of the bar in disturbed conditions with the MPC controller (2.34).

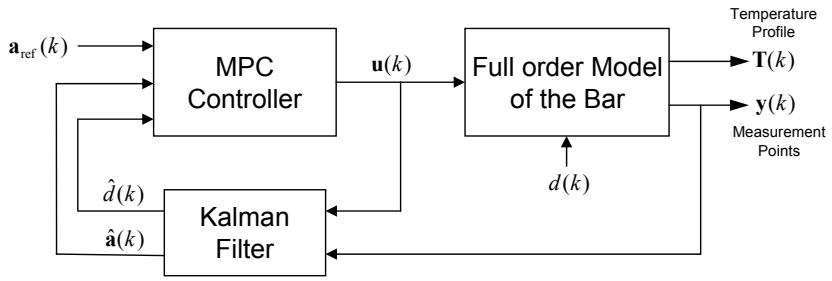


Figure 2.13: MPC control scheme with a disturbance model.

The simulation results of the MPC controller with an input disturbance model are shown in Figures 2.14 and 2.15. The left edge of the bar was exposed to the ambient temperature at the time step $k = 201$. Figures 2.14 and 2.15 show the evolution of the temperature profile before and after the disturbance respectively. The controller works very well and rejects

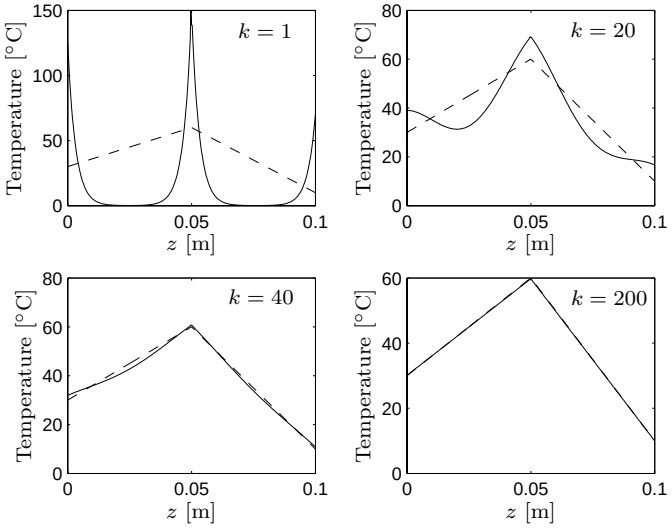


Figure 2.14: Evolution of the temperature profile before the disturbance (MPC with a disturbance model). Solid line - Closed-loop response. Dashed line - Reference.

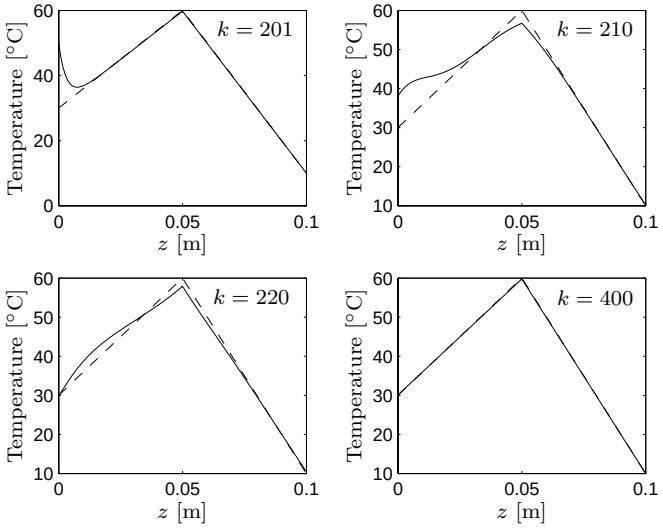


Figure 2.15: Evolution of the temperature profile after the disturbance (MPC with a disturbance model). Solid line - Closed-loop response. Dashed line - Reference.

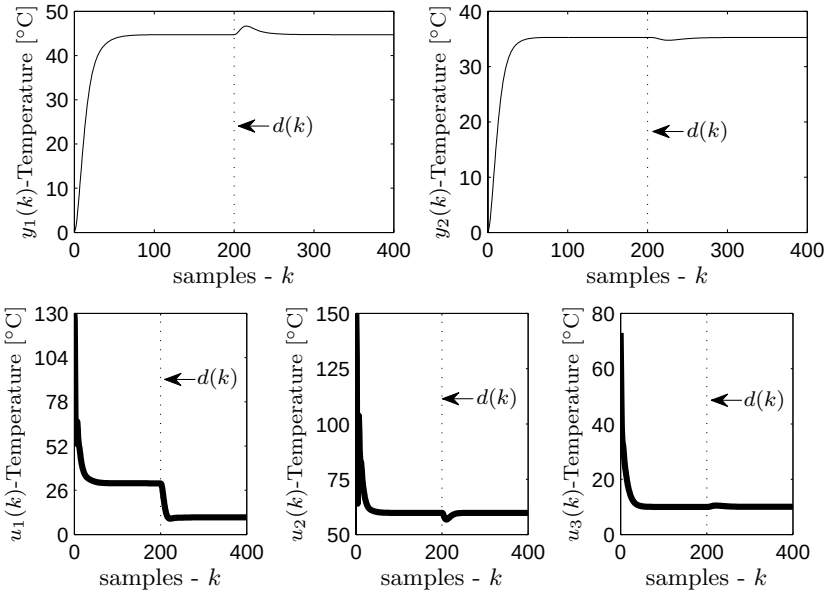


Figure 2.16: Temperature at the measurement points and control actions of the MPC controller (2.35) before and after the disturbance.

the perturbation quickly. In steady state, the maximum deviation in the temperature profile is only 0.21°C which is practically negligible. The control actions of the MPC Controller with a disturbance model as well as the evolution of the temperature at the measurement points can be observed in Figure 2.16. The control actions are all the time within the limits.

2.5 Conclusions

In this chapter we have presented the basics of Proper Orthogonal Decomposition (POD) and Galerkin projection, and we have shown how these two techniques can be used for deriving reduced order models of high-dimensional systems, typically obtained after discretizing in space the PDEs that model many processes. Furthermore we have presented the fundamentals of Model Predictive Control (MPC) and we have reviewed very briefly the basic ideas behind the Kalman filter. In the last section of

this chapter, we have included a very detailed example where the mentioned techniques have been used to control the temperature profile of a one-dimensional bar. In the example, a big model order reduction is achieved thanks to the POD technique that makes possible the control design. Initially, an MPC controller without a disturbance model was developed for controlling the temperature distribution when no input disturbances were applied to the bar. Its performance was very good in spite of the dramatic reduction of model order on which the controller is based. Afterwards, an MPC controller with a disturbance model was designed in order to reject the kind of disturbances that affect the system. This controller worked very well and rejected the perturbations quickly.

Chapter 3

Control of a Non-isothermal Tubular Reactor

3.1 Introduction

In this chapter the control of a non-isothermal tubular chemical reactor is addressed using POD and predictive control techniques. The control goal is to maintain the reactor at a desired operating condition in spite of disturbances in the feed flow, while keeping the maximum temperature inside the reactor low enough to avoid the formation of undesirable byproducts. The operating condition of the reactor is determined by means of an optimization algorithm which provides the optimal temperature and concentration profiles for the process [3]. This algorithm, which takes into account the input and state constraints of the system, is described in detail in this chapter. For controlling the reactor, we propose two MPC control schemes: an MPC where its formulation is in terms of the POD coefficients (similar to the formulation of the MPCs used to control the bar in Section 2.4) [3] and an MPC whose formulation is in terms of physical variables (the temperature of some selected points and the concentration at the reactor outlet) [4]. Unlike the first MPC controller, the second one incorporates the temperature constraint of the reactor in some selected points of the spatial domain. This MPC handles this constraint as a soft-constraint using a slack variable approach with l_∞-norm and time-dependent weights to deal with the infeasibilities that can emerge [68, 69].

Both MPC controllers are based on a linear model derived by means of the POD and Galerkin projection techniques, which reduce the high-dimensionality of the discretized system used to approximate the linear PDEs that model the reactor around the operating profiles. In this chapter several tests are carried out in order to evaluate the performance of the MPC controllers.

This chapter is organized as follows. In Section 3.2 we present a description of the tubular reactor that is used along this dissertation. We provide both the nonlinear and linear mathematical models of the system, and additionally we introduce the optimization algorithm for deriving the operating profiles of the reactor. Section 3.3 shows the derivation of the reduced order model of the process using POD and Galerkin projection. In Section 3.4 the design and implementation of the MPC control schemes is described. Section 3.5 presents some simulation results, and finally Section 3.6 summarizes the main conclusions.

3.2 Tubular chemical reactor

A chemical reactor is basically a vessel where chemical reactions take place. A reactor is usually the heart of an overall chemical or biochemical process. In order to model the behavior of most chemical reactors there are three main basic models that are commonly used, namely, the batch reactor model (batch), the Continuous Stirred-Tank Reactor (CSTR) model and the Plug Flow Reactor (PFR) model [122]. Plug flow reactors are also called Continuous Tubular Reactors (CTRs) or simply tubular reactors and nowadays are widespread in chemical industry. In [95], a tubular chemical reactor is defined and described as a tube in which chemical reactions take place during the continuous axial transport of the reaction mixture from the inlet towards the outlet. Typically, they are operated under time-invariant or steady-state conditions which leads to the production of large amounts of products with a constant and high quality. One big advantage of this kind of reactors is the possibility of large-scale and low cost production related to their continuous operation since there are no down times as there are in batch processes. Furthermore, they are suitable for advanced, automated process control and optimization techniques, and they deliver constant and high product quality due to the tight monitoring and control of the reaction environment. However, they have some disadvantages, the investment costs

are larger than in the other kinds of reactors and they are not suitable to produce a variety of products in small amounts since the switching between products can lead to a considerable amount of off-spec production [95].

For the sake of the generality of the results and conceptual contributions, in this dissertation we will focus our attention on an elementary reaction in an ideal plug-flow reactor model, instead of reactions in specific complex industrial reactors. In the following subsection we will describe in detail the type of tubular reactor for which we will design and implement POD-based MPC control strategies.

3.2.1 Plug flow reactor model

The system to be controlled is a non-isothermal tubular reactor where a single, first order, irreversible, exothermic reaction takes place ($A \rightarrow B$). The reactor is surrounded by 3 cooling/heating jackets as it is shown in Figure 3.1. The temperature of the jackets fluids ($T_{\mathrm{J}1}, T_{\mathrm{J}2}$ and $T_{\mathrm{J}3}$) can be manipulated independently in order to control the concentration and temperature profiles in the reactor. It is assumed that the fluid or mixture flows as a plug through the reactor body in the axial direction. In this dynamic only three phenomena are taken into account, namely, convection (macroscopic movement of the fluid inside the reactor), reaction (transformation of the chemical species present in the reactor, while releasing heat at the same time) and heat transfer (between the reactor and its jackets). In this study we are not considering the diffusion/dispersion phenomena and we are neglecting the heat transfer effects between the jackets fluids and the reactor wall. Under the previous assumptions, the mathematical model of the tubular chemical reactor consists of the following coupled nonlinear PDEs:

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial z} - k_0 C e^{-\frac{E}{RT}} \tag{3.1a}$$

$$\frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial z} + G_{\mathrm{r}} C e^{-\frac{E}{RT}} + H_{\mathrm{r}}(T_{\mathrm{w}} - T) \tag{3.1b}$$

$$G_{\mathrm{r}} = -\frac{\Delta H k_0}{\rho C_{\mathrm{p}}}, \quad H_{\mathrm{r}} = \frac{4h}{2r_{\mathrm{s}}\rho C_{\mathrm{p}}},$$

with the following boundary conditions:

$$C = C_{\mathrm{in}} \text{ at } z = 0 \text{ and } T = T_{\mathrm{in}} \text{ at } z = 0.$$

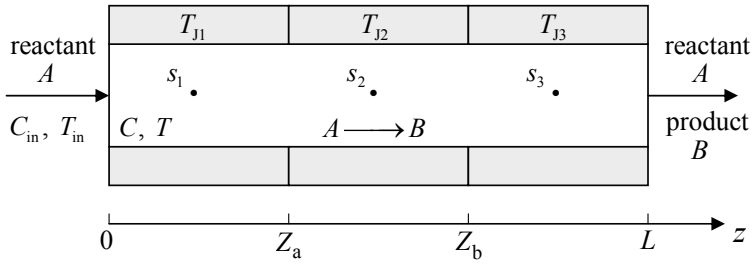


Figure 3.1: Tubular chemical reactor with 3 cooling/heating jackets.

Here $C(z,t)$ is the reactant concentration in [mol/l], $T(z,t)$ is the reactant temperature in [K], v is the fluid superficial velocity in [m/s], ΔH is the heat of the reaction in [cal/mol] ($\Delta H < 0$ for an exothermic reaction), ρ and $C_{\rm p}$ are the density in [kg/l] and the specific heat in [cal/kg/K] of the mix respectively, k_0 is the kinetic constant in [1/s], E is the activation energy in [cal/mol], R is the ideal gas constant in [cal/mol/K], h is the heat transfer coefficient in [cal/s/m^2/K], $r_{\rm s}$ is the reactor radius in [m], L is the reactor length in [m], $C_{\rm in}$ and $T_{\rm in}$ are the concentration in [mol/l] and the temperature in [K] of the feed flow, z is the axial coordinate in [m], t is the time in [s] and $T_{\rm w}(z,t)$ is the reactor wall temperature in [K] defined as follows (see Figure 3.1),

$$T_{\rm w} = \begin{cases} T_{\rm J1}, & 0 \leq z < Z_{\rm a} \\ T_{\rm J2}, & Z_{\rm a} \leq z < Z_{\rm b} \\ T_{\rm J3}, & Z_{\rm b} \leq z \leq L. \end{cases}$$

The parameters values of the reactor model are taken from [138] which were inspired by the values given in [53]. These values are presented in Table 3.1. Additionally, in Equation (3.1) the following assumptions have been made: perfect radial mixing (the reactor diameter is assumed to be small enough such that radial and angular variations are negligible), a constant heat transfer between the jacket and the reactor, a constant volume of the fluid in the reactor, a constant density and heat capacity of the reacting fluid, a constant velocity, and a constant heat of reaction. In Equation (3.1) only the temperature dependence of the reaction rate is incorporated through the Arrhenius law ($k_{\rm r} = k_0 e^{-E/(RT)}$, where $k_{\rm r}$ is the reaction rate constant). This is not the case for the other properties where the temperature dependence is neglected. As it is stated in [95], in spite of the conceptual and highly simplified character of model (3.1), this model can provide valuable insights for tubular reactors in practice, for example,

Table 3.1: Values of the reactor parameters

Parameter	Value
v	0.1 m·s^{-1}
L	1 m
k_0	10^6 s^{-1}
E	11250 cal·mol^{-1}
R	1.986 cal·mol^{-1}·K^{-1}
C_{in}	0.02 mol·l^{-1}
T_{in}	340 K
G_{r}	$4.25 \cdot 10^9$ l·K·mol^{-1}·s^{-1}
H_{r}	0.2 s^{-1}

in the production of low density polyethylene (LDPE) [30], or in oxidation processes in an environment with an excess of oxygen [52].

The temperature of the jacket sections $T_{\text{J}1}$, $T_{\text{J}2}$ and $T_{\text{J}3}$ must be between 280 K and 400 K. In addition, the temperature inside the reactor must be below 400 K in order to avoid the formation of side products. The kind of disturbances that affects the reactor are principally the variations in the temperature and concentration of the feed flow. Typically, such variations are in the range of ±10 K for the temperature and ±5% of the nominal value for the concentration. In this system, only the temperature of the feed flow is measured directly. In addition, the reactor has a temperature sensor at the output and 3 temperature sensors (s_1, s_2 and s_3) distributed in its interior as shown in Figure 3.1.

3.2.2 Operating profiles

The operating profiles (steady-state concentration and temperature profiles) of the reactor are derived by means of an optimization algorithm, which minimizes a cost function subject to the steady-state equations of the reactor described by (3.1), and the input and state constraints defined previously. The steady-state model of the reactor is given by the following Ordinary Differential Equations (ODEs):

$$\begin{aligned} \frac{dC}{dz} &= -\frac{k_0}{v} C e^{-\frac{E}{RT}} \\ \frac{dT}{dz} &= \frac{G_{\text{r}}}{v} C e^{-\frac{E}{RT}} + \frac{H_{\text{r}}}{v}(T_{\text{w}} - T) \end{aligned} \tag{3.2}$$

with $T = T_{\text{in}}$ at $z = 0$ and $C = C_{\text{in}}$ at $z = 0$, and the discrete version of (3.2) can be found by replacing the spatial derivatives by forward difference approximations as follows:

$$\begin{aligned} \bar{C}_{i+1} &= \bar{C}_i - \frac{k_0 \Delta z}{v} \bar{C}_i e^{\frac{-E}{RT_f \bar{T}_i}} \\ \bar{T}_{i+1} &= \bar{T}_i \left(1 - \frac{H_r \Delta z}{v}\right) + \left(\frac{G_r \Delta z C_f}{v T_f}\right) \bar{C}_i e^{\frac{-E}{RT_f \bar{T}_i}} + \left(\frac{H_r \Delta z}{v}\right) \bar{T}_{w,i} \\ &\qquad \text{for } i = 1, 2, \ldots, N = 300, \end{aligned} \tag{3.3}$$

with

$$\bar{T}_{\text{w,i}} = \begin{cases} \bar{T}_{J1} = T_{J1}/T_f, & \forall i = 1, \ldots, z_a \\ \bar{T}_{J2} = T_{J2}/T_f, & \forall i = z_a + 1, \ldots, z_b \\ \bar{T}_{J3} = T_{J3}/T_f, & \forall i = z_b + 1, \ldots, N \end{cases}$$

$$\bar{T}_0 = \frac{T_{\text{in}}}{T_f}$$

$$\bar{C}_0 = \frac{C_{\text{in}}}{C_f}$$

where N is the number of sections in which the reactor is divided, z_a and z_b are the reactor sections defining the ending of the first and second jacket respectively, T_f and C_f are normalization factors, $\bar{C}_i = C_i/C_f$ and $\bar{T}_i = T_i/T_f$ are the normalized concentration and temperature of the ith section of the reactor, $\bar{T}_{w,i} = T_{w,i}/T_f$ is the normalized reactor wall temperature of the ith section, and Δz is the length of each section. The variables are normalized in order to avoid possible numerical problems. The optimization problem that is solved for deriving the operating profiles is defined as:

$$\min_{\bar{T}_{J1}, \bar{T}_{J2}, \bar{T}_{J3}} w(\bar{C}_r - \bar{C}_N)^2 + (1 - w)\frac{1}{N} \sum_{i=1}^{N} \left(\bar{T}_{r,i} - \bar{T}_i\right)^2 \tag{3.4}$$

subject to

$$\begin{aligned} &\text{steady} - \text{state model given by (3.3)} \\ &\frac{T_{J_{\min}}}{T_f} \leq \bar{T}_{J1}, \bar{T}_{J2}, \bar{T}_{J3} \leq \frac{T_{J_{\max}}}{T_f} \\ &\bar{T}_i \leq \frac{T_{\max}}{T_f}, \quad \text{for } i = 1, 2, \ldots, N = 300, \end{aligned}$$

where $\bar{C}_r$ is the normalized desired concentration at the reactor output, $\bar{T}_{r,i}$ is the normalized desired temperature inside the reactor of the ith section,

$\bar{C}_N$ is the normalized concentration at the reactor output, w is a trade-off parameter, $T_{\mathrm{J}_{\min}}$ and $T_{\mathrm{J}_{\max}}$ are the limits of the jackets temperatures, and $T_{\max}$ is the maximum allowed temperature inside the tubular reactor.

The first term of the cost function corresponds to the squared error of the normalized concentration at the reactor output (terminal cost), and the second term is related to the mean squared error of the normalized temperature along the reactor (integral cost). In this problem $\bar{C}_{\mathrm{r}}$ was set to 0, and $\bar{T}_{\mathrm{r},i}$ was selected equal to the normalized temperature of the feed flow ($\bar{T}_{\mathrm{in}} = T_{\mathrm{in}}/T_{\mathrm{f}}$) for $i = 1, 2, \ldots, N$. The trade-off parameter w can take values from 0 to 1. When w goes to 1, the reduction of the reactant concentration at the reactor output becomes more important than the temperature deviations. On the other hand when w goes to 0, the temperature deviations become more important than the concentration at the reactor output and the risk of the formation of hot spots is reduced.

In order to solve the optimization problem described by (3.4) the following algorithm (a sort of Sequential Quadratic Programming - SQP) is proposed:

1. Choose the initial values of the jackets temperatures $\bar{\mathbf{T}}^*_{\mathrm{J}} = \left[\bar{T}^*_{\mathrm{J}1}, \bar{T}^*_{\mathrm{J}2}, \bar{T}^*_{\mathrm{J}3}\right]^T$ in such a way that the constraints are satisfied.

2. Using $\bar{\mathbf{T}}^*_{\mathrm{J}}$, simulate (3.3) in order to obtain the temperature ($\bar{\mathbf{T}}^* \in \mathbb{R}^N$) and concentration ($\bar{\mathbf{C}}^* \in \mathbb{R}^N$) profiles of the reactor in steady state.

3. Linearize the nonlinear model given by (3.3) around $\bar{\mathbf{T}}^*$,$\bar{\mathbf{C}}^*$, and $\bar{\mathbf{T}}^*_{\mathrm{J}}$ by means of the Taylor series. The resulting linear model would have the following structure:

$$\begin{aligned} \begin{bmatrix} \bar{C}^{\Delta}_{i+1} \\ \bar{T}^{\Delta}_{i+1} \end{bmatrix} &= \mathbf{A}_{\mathrm{ss}}(i) \begin{bmatrix} \bar{C}^{\Delta}_{i} \\ \bar{T}^{\Delta}_{i} \end{bmatrix} + \mathbf{B}_{\mathrm{ss}} \bar{T}^{\Delta}_{\mathrm{w},i} \\ \begin{bmatrix} \bar{C}_i \\ \bar{T}_i \end{bmatrix} &= \begin{bmatrix} \bar{C}^{\Delta}_{i} \\ \bar{T}^{\Delta}_{i} \end{bmatrix} + \begin{bmatrix} \bar{C}^{*}_{i} \\ \bar{T}^{*}_{i} \end{bmatrix} \end{aligned} \tag{3.5}$$

$$\text{for } i = 1, 2, \ldots, N,$$

with

$$\mathbf{A}_{\mathrm{ss}}(i) = \begin{bmatrix} 1 - \frac{k_0 \Delta z}{v} e^{-\frac{E}{RT^*_i}} & -\frac{k_0 T_{\mathrm{f}} E \Delta z}{v C_{\mathrm{f}} R} \frac{C^*_i}{T^{*2}_i} e^{-\frac{E}{RT^*_i}} \\ \frac{C_{\mathrm{f}} G_{\mathrm{r}} \Delta z}{v T_{\mathrm{f}}} e^{-\frac{E}{RT^*_i}} & 1 - \frac{H_{\mathrm{r}} \Delta z}{v} + \frac{G_{\mathrm{r}} E \Delta z}{vR} \frac{C^*_i}{T^{*2}_i} e^{-\frac{E}{RT^*_i}} \end{bmatrix},$$

$$\mathbf{B}_{\text{ss}} = \begin{bmatrix} 0 \\ \frac{H_{\text{r}}\Delta z}{v} \end{bmatrix},$$

where $\mathbf{A}_{\text{ss}}(i)$ and $\mathbf{B}_{ss}$ are the matrices describing the system dynamics, $\bar{C}_i^*$, $\bar{T}_i^*$ and $\bar{T}_{\text{w},i}^*$ are the normalized operating points of the concentration, temperature and reactor wall temperature of the ith section, $\bar{C}_i^{\Delta}$, $\bar{T}_i^{\Delta}$ and $\bar{T}_{\text{w},i}^{\Delta}$ are the normalized deviation variables of the concentration, temperature and reactor wall temperature respectively.

4. Solve the following Quadratic Problem (QP):

$$\min_{\bar{T}_{\text{J1}}^{\Delta},\bar{T}_{\text{J2}}^{\Delta},\bar{T}_{\text{J3}}^{\Delta}} w(\bar{C}_{\text{r}} - \bar{C}_N)^2 + (1-w)\frac{1}{N}\sum_{i=1}^{N}\left(\bar{T}_{\text{r},i} - \bar{T}_i\right)^2$$

subject to

$$\begin{aligned}
&\text{steady} - \text{state linear model given by (3.5)} \\
&\frac{T_{\text{Jmin}}}{T_{\text{f}}} \leq \bar{T}_{\text{J1}}, \bar{T}_{\text{J2}}, \bar{T}_{\text{J3}} \leq \frac{T_{\text{Jmax}}}{T_{\text{f}}} \\
&\frac{-T_{\text{J}}^{\Delta_{\max}}}{T_{\text{f}}} \leq \bar{T}_{\text{J1}}^{\Delta}, \bar{T}_{\text{J2}}^{\Delta}, \bar{T}_{\text{J3}}^{\Delta} \leq \frac{T_{\text{J}}^{\Delta_{\max}}}{T_{\text{f}}} \\
&\bar{T}_i \leq \frac{T_{\max}}{T_{\text{f}}}, \quad \text{for } i = 1, 2, \ldots, N = 300,
\end{aligned}$$

where $\bar{T}_{\text{J1}}^{\Delta}, \bar{T}_{\text{J2}}^{\Delta}, \bar{T}_{\text{J3}}^{\Delta}$ are the normalized deviation variables of the jackets temperatures, and $T_{\text{J}}^{\Delta_{\max}}$ is a "local input constraint" which limits the range of the jackets temperatures in such a way that the linear model (3.5) is still a good approximation of the nonlinear model (3.3). If this is not the case, then we would have convergence problems.

5. Calculate the new jackets temperatures $\bar{\mathbf{T}}_{\text{J}}^{\text{op}} \in \mathbb{R}^3$ as follows:

$$\bar{\mathbf{T}}_{\text{J}}^{\text{op}} = \bar{\mathbf{T}}_{\text{J}}^{\Delta,\text{op}} + \bar{\mathbf{T}}_{\text{J}}^{*}$$

where $\bar{\mathbf{T}}_{\text{J}}^{\Delta,\text{op}} = \left[\bar{T}_{\text{J1}}^{\Delta,\text{op}}, \bar{T}_{\text{J2}}^{\Delta,\text{op}}, \bar{T}_{\text{J3}}^{\Delta,\text{op}}\right]^T$ is the solution of the QP problem stated in the previous step.

6. Using $\bar{\mathbf{T}}_{\text{J}}^{\text{op}}$, simulate (3.3) in order to obtain the new temperature ($\bar{\mathbf{T}}^{\text{op}} \in \mathbb{R}^N$) and concentration ($\bar{\mathbf{C}}^{\text{op}} \in \mathbb{R}^N$) profiles of the reactor in steady state.

7. If $\max\left(\left|\bar{\mathbf{T}}_{\mathrm{J}}^{\mathrm{op}} - \bar{\mathbf{T}}_{\mathrm{J}}^{*}\right|\right) \leq \mathrm{Tol}$ then stop, else make $\bar{\mathbf{T}}_{\mathrm{J}}^{*} = \bar{\mathbf{T}}_{\mathrm{J}}^{\mathrm{op}}$, $\bar{\mathbf{C}}^{*} = \bar{\mathbf{C}}^{\mathrm{op}}$, $\bar{\mathbf{T}}^{*} = \bar{\mathbf{T}}^{\mathrm{op}}$ and go to step 3.

The proposed algorithm was executed with the following parameters : $N = 300$, $T_{\mathrm{f}} = 340$ K, $C_{\mathrm{f}} = 0.02$ mole/l, $\Delta z = 1/300$ m, $T_{\mathrm{Jmin}} = 280$ K, $T_{\mathrm{Jmax}} = 400$ K, $T_{\max} = 390$ K, $\mathrm{Tol} = 10^{-4}$, $w = 0.3$ and $T_{\mathrm{J}}^{\Delta_{\max}} = 20$ K.

The maximum allowed temperature ($T_{\max}$) inside the reactor was chosen 10 degrees below the actual limit (400 K) in order to give to the feedback controller enough room of maneuverability. The trade-off coefficient w was found by trial and error and the local input constraint $T_{\mathrm{J}}^{\Delta_{\max}}$ was selected in such a way that the differences between the nonlinear and linear model are small.

The algorithm was executed using different initial conditions and some of the results obtained are presented in Table 3.2. Along the experiments, three local minima were found, however it does not mean that these are the only ones. Figures 3.2, 3.3 and 3.4 show the reactor profiles associated to these local minima. This result was expected due to the non-convex nature of the optimization problem. These local minima might be unified by modifying the cost function, for instance, by adding a term that penalizes the temperature deviation at the reactor output.

The selection of the optimal temperature and concentration profiles was done by checking the value of the cost function and the deviation of the temperature at the reactor output with respect to the temperature of the feed flow. In many practical situations, a large deviation is undesirable since it increases the heat loss [137].

From the three local minima, the second one ($T_{\mathrm{J1}} = 374.6$ K, $T_{\mathrm{J2}} = 310.1$ K and $T_{\mathrm{J3}} = 325.2$ K) was chosen since it has the smallest cost function value and a small temperature deviation at the reactor output. The optimal concentration and temperature profiles can be observed in Figure 3.4. The concentration at the reactor output is $1.5737 \cdot 10^{-3}$ mol/l which is 12.7 times smaller than the concentration of the feed flow (0.02 mol/l). In addition, the temperature of the hot spot is 390 K.

3.2.3 Linear model

The linear model of the tubular chemical reactor is obtained by linearizing (3.1) around the jackets temperatures and the operating profiles presented

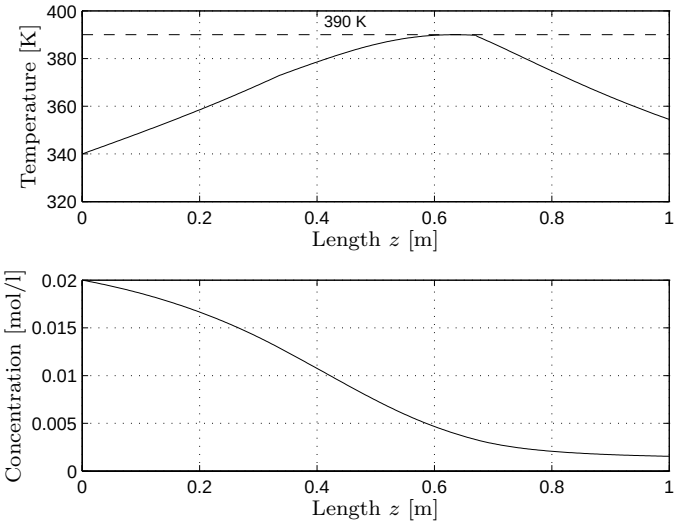


Figure 3.2: Steady-state concentration and temperature profiles when $T_{\mathrm{J}1} = 359.6$ K, $T_{\mathrm{J}2} = 348.5$ K and $T_{\mathrm{J}3} = 291.5$ K.

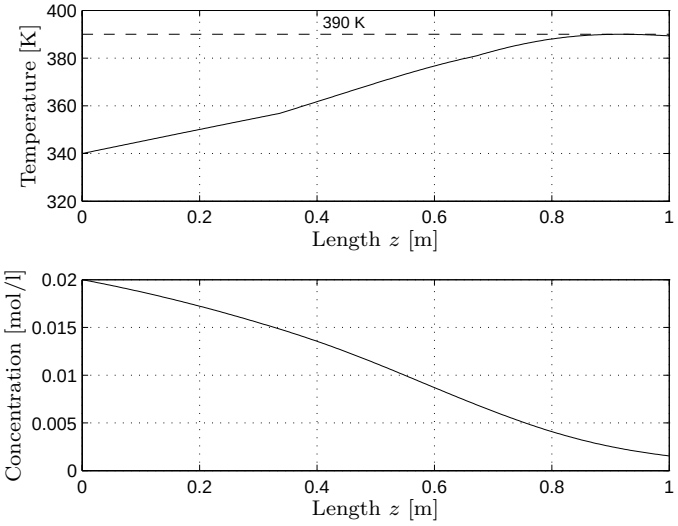


Figure 3.3: Steady-state concentration and temperature profiles when $T_{\mathrm{J}1} = 340.7$ K, $T_{\mathrm{J}2} = 355.1$ K and $T_{\mathrm{J}3} = 365.7$ K.

Table 3.2: Some results obtained with the optimization algorithm

Initial Jackets Temperatures			Final Jackets Temperatures			Cost Function	C_{in}/C_N	No. Iter [a]
T_{J1}	T_{J2}	T_{J3}	T_{J1}	T_{J2}	T_{J3}			
- First Minimum - (see Figure 3.2)								
300	300	300						66
350	350	350	359.6	348.5	291.5	$8.356 \cdot 10^{-3}$	12.6	73
280	400	280						91
- Second Minimum - (see Figure 3.4)								
350	280	350	374.6	310.1	325.2	$8.337 \cdot 10^{-3}$	12.7	38
370	290	370						39
- Third Minimum - (see Figure 3.3)								
280	280	400	340.7	355.1	365.7	$8.416 \cdot 10^{-3}$	12.7	29
380	340	400						24

[a]Number of iterations

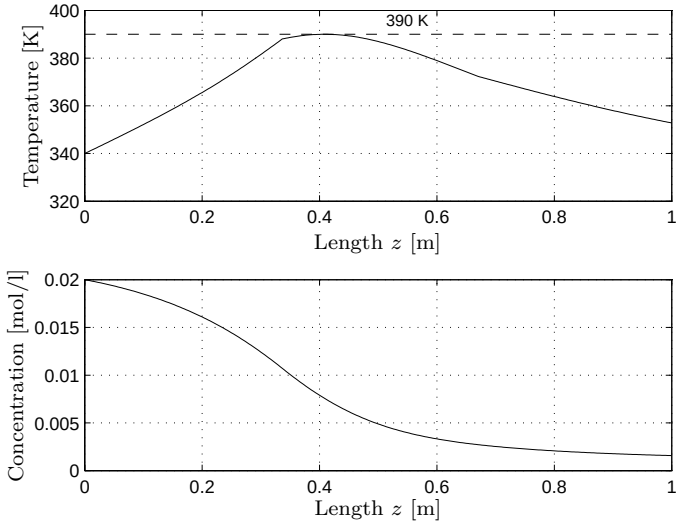


Figure 3.4: Selected operating profiles. Steady-state concentration and temperature profiles when $T_{\text{J1}} = 374.6$ K, $T_{\text{J2}} = 310.1$ K and $T_{\text{J3}} = 325.2$ K.

in Figure 3.4. This linear model is given by,

$$\begin{aligned}\frac{\partial C^{\Delta}}{\partial t} &= -v\frac{\partial C^{\Delta}}{\partial z} - \alpha_{\mathrm{A}}(z)C^{\Delta} - \alpha_{\mathrm{B}}(z)T^{\Delta} \\ \frac{\partial T^{\Delta}}{\partial t} &= -v\frac{\partial T^{\Delta}}{\partial z} - \alpha_{\mathrm{C}}(z)C^{\Delta} - \alpha_{\mathrm{D}}(z)T^{\Delta} + H_{\mathrm{r}}T_{\mathrm{w}}^{\Delta}\end{aligned} \tag{3.6}$$

with

$$\begin{aligned}\alpha_{\mathrm{A}}(z) &= k_0 e^{-\frac{E}{RT^*}} \\ \alpha_{\mathrm{B}}(z) &= k_0 C^* \frac{E}{RT^{*2}} e^{-\frac{E}{RT^*}} \\ \alpha_{\mathrm{C}}(z) &= -G_{\mathrm{r}} e^{-\frac{E}{RT^*}} \\ \alpha_{\mathrm{D}}(z) &= -G_{\mathrm{r}} C^* \frac{E}{RT^{*2}} e^{-\frac{E}{RT^*}} + H_{\mathrm{r}}\end{aligned}$$

and with these boundary conditions

$$C^{\Delta} = C_{\mathrm{in}} - C_{\mathrm{in}}^*, \quad T^{\Delta} = T_{\mathrm{in}} - T_{\mathrm{in}}^* \text{ at } z = 0.$$

Here C^*, T^* and T_{w}^* are the steady state profiles (operating profiles) of the concentration, temperature and reactor wall temperature respectively, C_{in}^* and T_{in}^* are the concentration and temperature in steady state of the feed flow, $C^{\Delta} = C - C^*$, $T^{\Delta} = T - T^*$ and $T_{\mathrm{w}}^{\Delta} = T_{\mathrm{w}} - T_{\mathrm{w}}^*$ are the deviations from steady state of the concentration, temperature and reactor wall temperature.

In order to reduce the infinite dimensionality of (3.6), the partial derivatives with respect to space are replaced by backward difference approximations leading to the following system of ODEs:

$$\begin{aligned}\frac{d\bar{C}_i^{\Delta}}{dt} &= -\frac{v}{\Delta z}\left(\bar{C}_i^{\Delta} - \bar{C}_{i-1}^{\Delta}\right) - \alpha_{\mathrm{A},i}\bar{C}_i^{\Delta} - \alpha_{\mathrm{B},i}\frac{T_{\mathrm{f}}}{C_{\mathrm{f}}}\bar{T}_i^{\Delta} \\ \frac{d\bar{T}_i^{\Delta}}{dt} &= -\frac{v}{\Delta z}\left(\bar{T}_i^{\Delta} - \bar{T}_{i-1}^{\Delta}\right) - \alpha_{\mathrm{C},i}\frac{C_{\mathrm{f}}}{T_{\mathrm{f}}}\bar{C}_i^{\Delta} - \alpha_{\mathrm{D},i}\bar{T}_i^{\Delta} + H_{\mathrm{r}}\bar{T}_{\mathrm{w},i}^{\Delta}\end{aligned} \tag{3.7}$$

$$\text{for } i = 1, 2, \ldots, N,$$

with

$$\bar{C}_0^{\Delta} = \bar{C}_{\mathrm{in}}^{\Delta},\ \bar{T}_0^{\Delta} = \bar{T}_{\mathrm{in}}^{\Delta},$$

where $\alpha_{\mathrm{A},i} = \alpha_{\mathrm{A}}(z_i)$, $\alpha_{\mathrm{B},i} = \alpha_{\mathrm{B}}(z_i)$, $\alpha_{\mathrm{C},i} = \alpha_{\mathrm{C}}(z_i)$, $\alpha_{\mathrm{D},i} = \alpha_{\mathrm{D}}(z_i)$, $z_i = i\Delta z$, C_i, T_i, $T_{\mathrm{w},i}$ are the concentration, temperature and reactor wall temperature of the ith section, T_{f} and C_{f} are normalization factors, C_i^*, T_i^*, $T_{\mathrm{w},i}^*$ are the steady state concentration, temperature and reactor wall temperature of the ith section, C_{in}^* and T_{in}^* are the the steady state concentration and temperature of the feed flow, $\bar{C}_i^{\Delta} = (C_i - C_i^*)/C_{\mathrm{f}}$, $\bar{T}_i^{\Delta} = (T_i - T_i^*)/T_{\mathrm{f}}$, $\bar{T}_{\mathrm{w},i}^{\Delta} = (T_{\mathrm{w},i} - T_{\mathrm{w},i}^*)/T_{\mathrm{f}}$ are the normalized deviations from steady state of the concentration, temperature and reactor wall temperature of the ith section, $\bar{C}_{\mathrm{in}}^{\Delta} = (C_{\mathrm{in}} - C_{\mathrm{in}}^*)/C_{\mathrm{f}}$ and $\bar{T}_{\mathrm{in}}^{\Delta} = (T_{\mathrm{in}} - T_{\mathrm{in}}^*)/T_{\mathrm{f}}$ are the normalized deviations from steady state of the concentration and temperature of the feed flow, N is the number of sections in which the reactor is divided, and Δz is the length of each section.

If we define the following vectors,

$$\begin{aligned}
\mathbf{x}(t) &= \left[\bar{C}_1^{\Delta}, \bar{C}_2^{\Delta}, \ldots, \bar{C}_N^{\Delta}, \bar{T}_1^{\Delta}, \bar{T}_2^{\Delta}, \ldots, \bar{T}_N^{\Delta}\right]^T \\
\mathbf{d}(t) &= \left[\bar{C}_{\mathrm{in}}^{\Delta}, \bar{T}_{\mathrm{in}}^{\Delta}\right]^T \\
\mathbf{u}(t) &= \left[\bar{T}_{\mathrm{J1}}^{\Delta}, \bar{T}_{\mathrm{J2}}^{\Delta}, \bar{T}_{\mathrm{J3}}^{\Delta}\right]^T
\end{aligned}$$

then (3.7) can be cast as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{d}(t) \tag{3.8}$$

with

$$\mathbf{A} \in \mathbb{R}^{2N\times 2N} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix},\ \mathbf{B} \in \mathbb{R}^{2N\times 3} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_1 \end{bmatrix},\ \mathbf{F} \in \mathbb{R}^{2N\times 2} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_1 \end{bmatrix},$$

$$\mathbf{A}_1 \in \mathbb{R}^{N\times N} = \begin{bmatrix} -\left(\alpha_{\mathrm{A},1} + \frac{v}{\Delta z}\right) & 0 & \cdots & \cdots & 0 \\ \frac{v}{\Delta z} & -\left(\alpha_{\mathrm{A},2} + \frac{v}{\Delta z}\right) & \ddots & & \vdots \\ 0 & \frac{v}{\Delta z} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{v}{\Delta z} & -\left(\alpha_{\mathrm{A},N} + \frac{v}{\Delta z}\right) \end{bmatrix}$$

$$\mathbf{A}_2 \in \mathbb{R}^{N\times N} = \begin{bmatrix} -\alpha_{\mathrm{B},1}\frac{T_{\mathrm{f}}}{C_{\mathrm{f}}} & 0 & \cdots & 0 \\ 0 & -\alpha_{\mathrm{B},2}\frac{T_{\mathrm{f}}}{C_{\mathrm{f}}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\alpha_{\mathrm{B},N}\frac{T_{\mathrm{f}}}{C_{\mathrm{f}}} \end{bmatrix}$$

$$\mathbf{A}_3 \in \mathbb{R}^{N\times N} = \begin{bmatrix} -\alpha_{\mathrm{C},1}\frac{C_\mathrm{f}}{T_\mathrm{f}} & 0 & \cdots & 0 \\ 0 & -\alpha_{\mathrm{C},2}\frac{C_\mathrm{f}}{T_\mathrm{f}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\alpha_{\mathrm{C},N}\frac{C_\mathrm{f}}{T_\mathrm{f}} \end{bmatrix}$$

$$\mathbf{A}_4 \in \mathbb{R}^{N\times N} = \begin{bmatrix} -\left(\alpha_{\mathrm{D},1}+\frac{v}{\Delta z}\right) & 0 & \cdots & \cdots & 0 \\ \frac{v}{\Delta z} & -\left(\alpha_{\mathrm{D},2}+\frac{v}{\Delta z}\right) & \ddots & & \vdots \\ 0 & \frac{v}{\Delta z} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{v}{\Delta z} & -\left(\alpha_{\mathrm{D},N}+\frac{v}{\Delta z}\right) \end{bmatrix}$$

$$\mathbf{B}_1 \in \mathbb{R}^{N\times 3} = \begin{bmatrix} H_\mathrm{r} & \cdots & H_\mathrm{r} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & H_\mathrm{r} & \cdots & H_\mathrm{r} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & H_\mathrm{r} & \cdots & H_\mathrm{r} \end{bmatrix}^T$$

$$\mathbf{f}_1 \in \mathbb{R}^N = \begin{bmatrix} \frac{v}{\Delta z} & 0 & \cdots & 0 \end{bmatrix}^T$$

where $T^*_{\mathrm{J}1}$, $T^*_{\mathrm{J}2}$, $T^*_{\mathrm{J}3}$, are the steady state jacket temperatures, $\bar{T}^{\Delta}_{\mathrm{J}1} = (T_{\mathrm{J}1} - T^*_{\mathrm{J}1})/T_\mathrm{f}$, $\bar{T}^{\Delta}_{\mathrm{J}2} = (T_{\mathrm{J}2} - T^*_{\mathrm{J}2})/T_\mathrm{f}$ and $\bar{T}^{\Delta}_{\mathrm{J}3} = (T_{\mathrm{J}3} - T^*_{\mathrm{J}3})/T_\mathrm{f}$ are the normalized deviations of the jacket temperatures, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{F}$ are the matrices describing the system, $\mathbf{x}(t) \in \mathbb{R}^{2N}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^3$ is the vector of the inputs and $\mathbf{d}(t) \in \mathbb{R}^2$ is the vector of the disturbances.

Since the spatial domain of the reactor is divided into $N = 300$ sections, the number of states of (3.8) is equal to 600. Given that such large number of states makes the design and implementation of feedback controllers for the reactor difficult, in the next section a reduced order model will be derived using POD and Galerkin projection.

3.3 Model reduction using POD

In a similar fashion as it was done in Section 2.4.3, the reduced order model of (3.8) is found by following the subsequent steps:

1. **Generation of the Snapshot Matrix**. We have created a snapshot matrix $\mathbf{X}_{\mathrm{snap}} \in \mathbb{R}^{600\times 1500}$ from the system response when independent step changes were made in the input $\mathbf{u}(t)$ and perturbation $\mathbf{d}(t)$ signals

of the linear model (3.8),

$$\mathbf{X}_{\text{snap}} = [\mathbf{x}(t = \Delta t), \mathbf{x}(t = 2\Delta t), \ldots, \mathbf{x}(t = 1500\Delta t)]\,. \tag{3.9}$$

Along the simulations 1500 samples were collected using a sampling time Δt of 0.05 s. The amplitude of the step changes was chosen in such a way as to produce changes of similar magnitude in the temperature and concentration profiles. This avoids a possible bias in the resulting model.

2. **Derivation of the POD basis vectors.** The POD basis vectors are obtained by computing the SVD of the snapshot matrix $\mathbf{X}_{\text{snap}}$,

$$\mathbf{X}_{\text{snap}} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$$

where $\mathbf{\Phi} \in \mathbb{R}^{600\times600}$ and $\mathbf{\Psi} \in \mathbb{R}^{1500\times1500}$ are unitary matrices, and $\mathbf{\Sigma} \in \Re^{600\times1500}$ is a matrix that contains the singular values of $\mathbf{X}_{\text{snap}}$ in a decreasing order on its main diagonal. The left singular vectors, i.e., the columns of $\mathbf{\Phi}$,

$$\mathbf{\Phi} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{600}]$$

are the POD basis vectors.

3. **Selection of the most relevant POD basis vectors.** The n most relevant POD basis vectors are chosen using the energy criterion presented in Section 2.2.1. The plot of $1 - \bar{P}_n$ (see Equation (2.7)) for the first 160 basis vectors is shown in Figure 3.5. In this problem, we chose the first $n = 20$ POD basis vectors based on their truncation degree $1-\bar{P}_n = 3.3\cdot10^{-4}$ ($\bar{P}_n = 0.9996$). In Figures 3.6 and 3.7 we can observe the basis vectors associated to the 20 largest singular vectors. Observe that the first half of each basis vector is associated to the normalized deviations of the concentration profile whereas the second half is related to the normalized deviations of the temperature profile.

   The 20th order approximation of $\mathbf{x}(t)$ is given by the following truncated sequence:

$$\mathbf{x}_n(t) = \sum_{j=1}^{20} a_j(t)\boldsymbol{\varphi}_j = \mathbf{\Phi}_n\mathbf{a}(t) \tag{3.10}$$

   where $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{20}]$ and $\mathbf{a}(t) = [a_1(t), a_2(t), \ldots, a_{20}(t)]^T$.

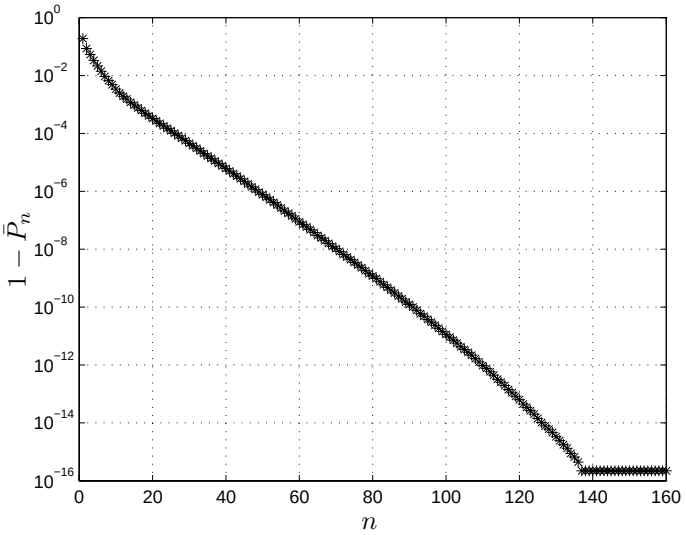


Figure 3.5: Logarithmic plot of $1-\bar{P}_n$ for determining the truncation degree of the POD basis vectors in the reactor case.

4. **Construction of the model for the first n=20 POD coefficients.** As it was explained in Section 2.2.2, the dynamic model for the POD coefficients can be derived by projecting (Galerkin projection) the model (3.8) on the space spanned by the selected POD basis vectors $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{20}]$. If we replace $\mathbf{x}(t)$ by its nth order approximation $\mathbf{x}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t)$ in Equation (3.8), and we apply the inner product criterion (Galerkin projection) to the resulting equation, we have that

$$\left\langle \mathbf{\Phi}_n \dot{\mathbf{a}}(t), \boldsymbol{\varphi}_j \right\rangle = \left\langle \mathbf{A}\mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{d}(t), \boldsymbol{\varphi}_j \right\rangle \tag{3.11}$$

$$\forall j = 1, 2, \ldots, n = 20.$$

By evaluating the inner product in (3.11),

$$\mathbf{\Phi}_n^T \mathbf{\Phi}_n \dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T \mathbf{A}\mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{\Phi}_n^T \mathbf{B}\mathbf{u}(t) + \mathbf{\Phi}_n^T \mathbf{F}\mathbf{d}(t)$$

$$\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T \mathbf{A}\mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{\Phi}_n^T \mathbf{B}\mathbf{u}(t) + \mathbf{\Phi}_n^T \mathbf{F}\mathbf{d}(t) \tag{3.12}$$

we obtain the model for the first n POD coefficients. The reduced order model of the reactor with only 20 states is then given by

$$\dot{\mathbf{a}}(t) = \mathbf{A}_{\mathrm{r}} \mathbf{a}(t) + \mathbf{B}_{\mathrm{r}} \mathbf{u}(t) + \mathbf{F}_{\mathrm{r}} \mathbf{d}(t) \tag{3.13}$$

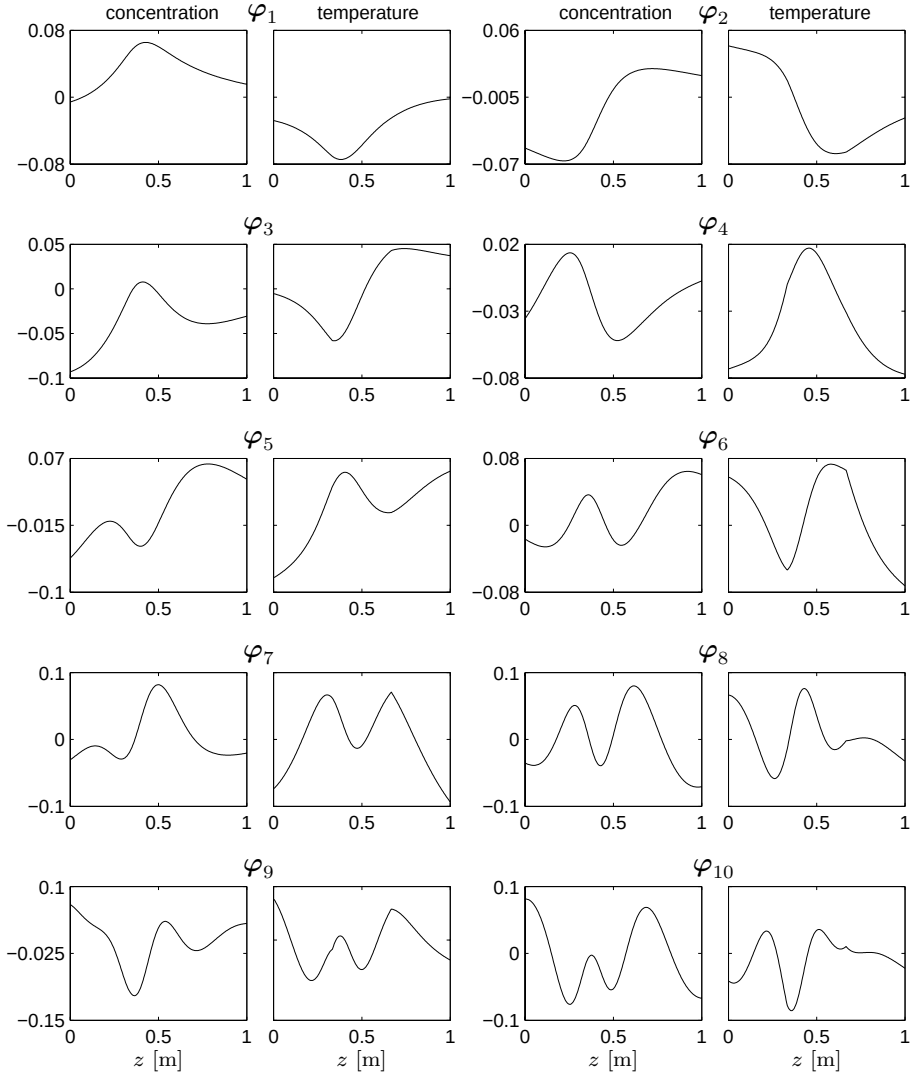


Figure 3.6: POD basis vectors : $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{10}$.

$$\mathbf{x}_n(t) = \boldsymbol{\Phi}_n \mathbf{a}(t)$$

where $\mathbf{A}_\mathrm{r} = \boldsymbol{\Phi}_n^T \mathbf{A} \boldsymbol{\Phi}_n$, $\mathbf{B}_\mathrm{r} = \boldsymbol{\Phi}_n^T \mathbf{B}$ and $\mathbf{F}_\mathrm{r} = \boldsymbol{\Phi}_n^T \mathbf{F}$. The initial condition for $\mathbf{a}(t)$ reads as $\mathbf{a}(0) = \mathbf{0}$.

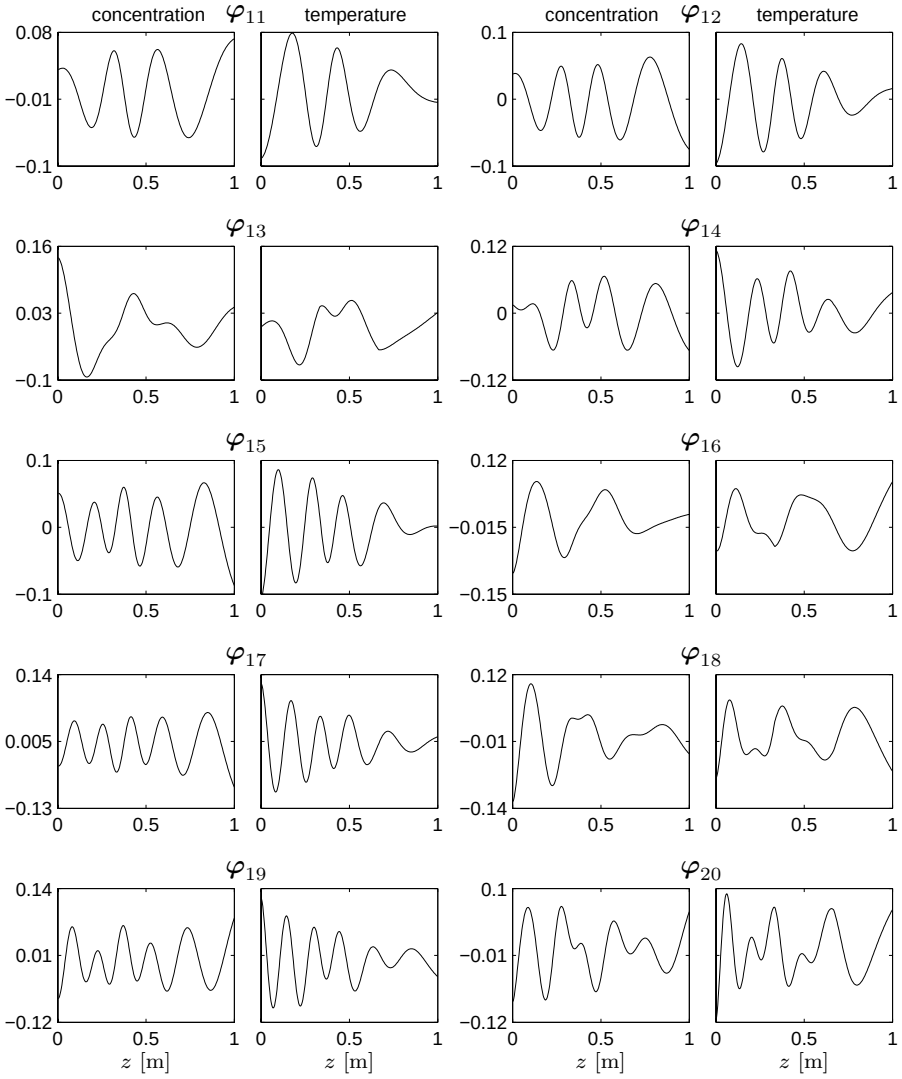


Figure 3.7: POD basis vectors : $\varphi_{11}, \varphi_{12}, \ldots, \varphi_{20}$.

For validating the reduced order model of the reactor, we applied constant input signals $\bar{T}^{\Delta}_{\mathrm{J1}}(t) = 0.125$ ($T^{\Delta}_{\mathrm{J1}}(t) = 10$ K), $\bar{T}^{\Delta}_{\mathrm{J2}}(t) = -0.25$ ($T^{\Delta}_{\mathrm{J2}}(t) = -20$ K) and $\bar{T}^{\Delta}_{\mathrm{J3}}(t) = 0.25$ ($T^{\Delta}_{\mathrm{J3}}(t) = 20$ K) and constant perturbation signals $\bar{C}^{\Delta}_{\mathrm{in}}(t) = 0.05$ ($C^{\Delta}_{\mathrm{in}}(t) = 10^{-3}$ mol/l) and $\bar{T}^{\Delta}_{\mathrm{in}}(t) = 0.0625$ ($T^{\Delta}_{\mathrm{in}}(t) = 5$ K) to both the full order model (3.8) and the reduced order model (3.13), and afterwards we compared their responses. Figures 3.8 and 3.9 show the

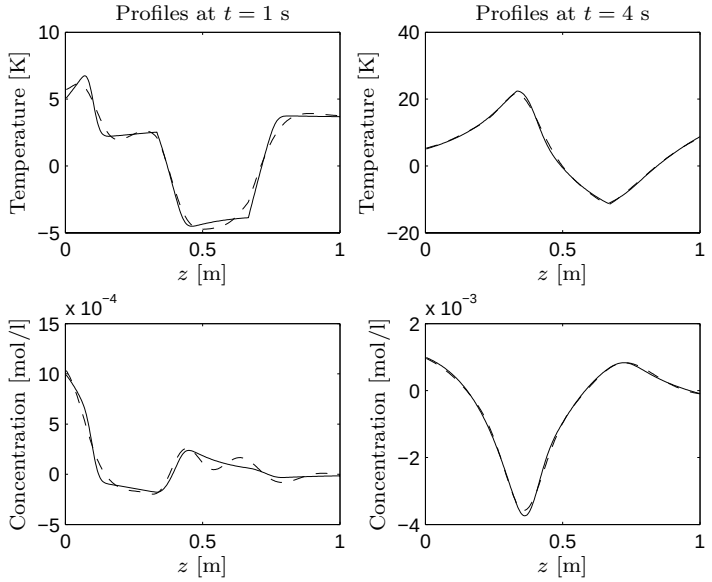


Figure 3.8: Temperature and concentration deviation profiles at $t = 1$ s and $t = 4$ s. Solid line - Full order model. Dashed line - Reduced order model.

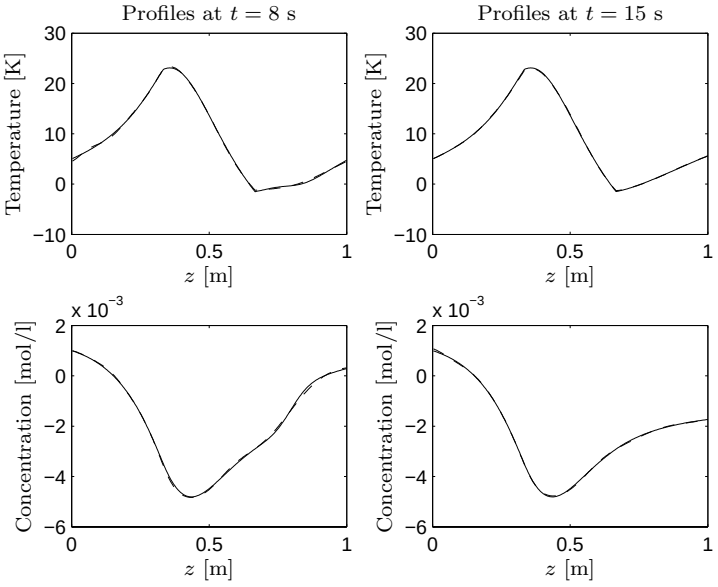


Figure 3.9: Temperature and concentration deviation profiles at $t = 8$ s and $t = 15$ s. Solid line - Full order model. Dashed line - Reduced order model.

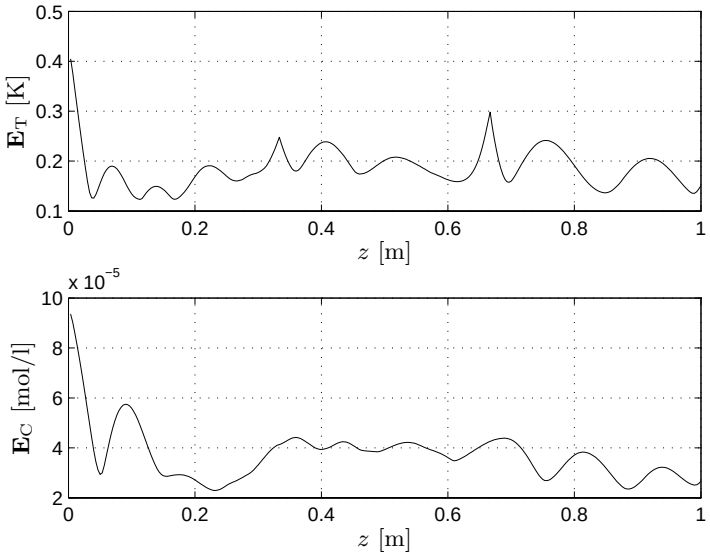


Figure 3.10: Average of the absolute error between the full order model (3.8) and the reduced order model (3.13).

temperature and concentration deviation profiles of the reactor at different time instants for each model. During the first seconds we can see some differences between the models, but as the simulation advances, these differences become negligible and really difficult to observe. In order to measure the quality of the reduced order model the averages of the absolute error for the temperature ($\mathbf{E}_\mathrm{T}$) and concentration ($\mathbf{E}_\mathrm{C}$) were calculated by means of the following formulas:

$$\mathbf{E}_\mathrm{T} = \frac{T_\mathrm{f}}{N_\mathrm{s}} \sum_{k=1}^{N_s} \left| \bar{\mathbf{T}}^{\Delta}(k\Delta t) - \bar{\mathbf{T}}_n^{\Delta}(k\Delta t) \right|$$

$$\mathbf{E}_\mathrm{C} = \frac{C_\mathrm{f}}{N_\mathrm{s}} \sum_{k=1}^{N_s} \left| \bar{\mathbf{C}}^{\Delta}(k\Delta t) - \bar{\mathbf{C}}_n^{\Delta}(k\Delta t) \right|$$

where $N_\mathrm{s} = 300$ is the number of time steps and $\Delta t = 0.05$ s. The plots of $\mathbf{E}_\mathrm{T}$ and $\mathbf{E}_\mathrm{C}$ with respect to the spatial coordinate z are shown in Figure 3.10. The maximum values for the errors $\mathbf{E}_\mathrm{T}$ and $\mathbf{E}_\mathrm{C}$ are 0.405 Kelvin and $9.35 \cdot 10^{-5}$ mol/l respectively. From the previous results we can conclude that the reduced order model with only 20 states provides an acceptable approximation of the full order model.

The discrete-time version of (3.13) that will be used by the predictive controllers is obtained using the bilinear transformation [116] with a sampling time of 0.2 s,

$$\mathbf{a}(k+1) = \tilde{\mathbf{A}}\mathbf{a}(k) + \tilde{\mathbf{B}}\mathbf{u}(k) + \tilde{\mathbf{F}}\mathbf{d}(k) \tag{3.14}$$

$$\mathbf{x}_n(k) = \mathbf{\Phi}_n\mathbf{a}(k),$$

where $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$ are the matrices describing the new system. The sampling time was chosen by dividing the smallest time constant of the system (3.13) by 10.

3.4 Predictive control schemes

The control objective is to reject the disturbances that affect the reactor, that is the changes in the temperature and concentration of the feed flow. In addition, the control actions must satisfy the input constraints of the process (280 K$\leq T_{\mathrm{J}1}(t), T_{\mathrm{J}2}(t), T_{\mathrm{J}3}(t) \leq$ 400 K), and the control system should keep the temperature inside the reactor below 400 K. In the following subsections we are going to present two POD-based MPC control schemes: a scheme where the MPC is formulated in terms of the POD coefficients (MPC-NTC), and a scheme where the formulation of the MPC controller is in terms of physical variables (MPC-PV).

3.4.1 First MPC control scheme (MPC-NTC) - Formulation in terms of the POD coefficients

In this scheme, the control of the temperature and concentration profiles is achieved indirectly by controlling the POD coefficients. The references ($\mathbf{a}_{\mathrm{ref}}$) of these POD coefficients can be calculated by

$$\mathbf{a}_{\mathrm{ref}} = \mathbf{\Phi}_n^T\mathbf{x}_{\mathrm{ref}} \tag{3.15}$$

where $\mathbf{x}_{\mathrm{ref}}$ is the reference of the vector $\mathbf{x}(t)$ and is equal to $\mathbf{0}$ since the control system has to keep the reactor operating around the profiles shown in Figure 3.4. The MPC controller, which uses model (3.14) to predict the future behavior of the reactor, is formulated as follows:

$$\min_{\mathbf{a}_{N_\mathrm{p}},\Delta\mathbf{u}_{N_\mathrm{c}},\mathbf{d}_{N_\mathrm{p}}} \sum_{i=1}^{N_\mathrm{p}} \|\mathbf{a}_\mathrm{ref}(k+i)-\mathbf{a}(k+i)\|^2_{\mathbf{Q}} + \sum_{i=0}^{N_\mathrm{c}-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}} \quad (3.16a)$$

subject to

$$\begin{aligned}
&\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \tilde{\mathbf{F}}\mathbf{d}(k+i), && i=0,\ldots,N_\mathrm{p}-1,\\
&\mathbf{d}(k+i+1) = \mathbf{d}(k+i), && i=0,\ldots,N_\mathrm{p}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i=0,\ldots,N_\mathrm{c}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i=N_\mathrm{c},\ldots,N_\mathrm{p}-1,\\
&\mathbf{u}_\mathrm{min} \le \mathbf{u}(k+i) \le \mathbf{u}_\mathrm{max}, && i=0,\ldots,N_\mathrm{c}-1,
\end{aligned} \quad (3.16b)$$

with

$$\begin{aligned}
\mathbf{a}_{N_\mathrm{p}} &= [\mathbf{a}(k+1);\mathbf{a}(k+2);\ldots;\mathbf{a}(k+N_\mathrm{p})]\\
\mathbf{d}_{N_\mathrm{p}} &= [\mathbf{d}(k+1);\mathbf{d}(k+2);\ldots;\mathbf{d}(k+N_\mathrm{p})]\\
\Delta\mathbf{u}_{N_\mathrm{c}} &= [\Delta\mathbf{u}(k);\Delta\mathbf{u}(k+1);\ldots;\Delta\mathbf{u}(k+N_\mathrm{c}-1)]
\end{aligned}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices ($\mathbf{Q}\succeq 0, \mathbf{R}\succ 0$), $\|\mathbf{v}\|^2_{\mathbf{Q}}$ denotes $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, N_p is the prediction horizon, N_c is the control horizon, $\mathbf{u}_\mathrm{min}$ and $\mathbf{u}_\mathrm{max}$ are the lower and upper bounds (these hard constraints are necessary due to physical limitations of the actuators) of $\mathbf{u}(k)$ and Equation (3.16b) describes the dynamics of the disturbance vector which in this case is assumed to be a step. Observe that the temperature constraint $T(z,t) \le 400$ K of the reactor has not been included in this MPC formulation.

Since the state vector $\mathbf{a}(k)$ is unknown and the changes in the concentration of the feed flow ($d_1(k) = \bar{C}^{\Delta}_{\mathrm{in}}(k)$) are not measured directly, they are estimated by means of an observer (in this case a Kalman filter) with the following formulation:

$$\begin{bmatrix} \hat{\mathbf{a}}(k+1) \\ \hat{d}_1(k+1) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{F}}_\mathrm{C} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}(k) \\ \hat{d}_1(k) \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) +$$

$$+ \begin{bmatrix} \tilde{\mathbf{F}}_\mathrm{T} \\ 0 \end{bmatrix} d_2(k) + \begin{bmatrix} \mathbf{L}_\mathrm{a} \\ \mathbf{L}_\mathrm{d} \end{bmatrix} (\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \quad (3.17a)$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}_\mathrm{s}\hat{\mathbf{x}}_n(k) = \mathbf{C}_\mathrm{s}\mathbf{\Phi}_n\hat{\mathbf{a}}(k) \quad (3.17b)$$

where $\hat{\mathbf{a}}(k)$ is the estimated vector of the POD coefficients, $\hat{d}_1(k)$ is the estimation of $\bar{C}_{\text{in}}^{\Delta}(k)$, $d_2(k)$ is the normalized temperature deviation of the feed flow $\bar{T}_{\text{in}}^{\Delta}(k)$, $\mathbf{y}(k) \in \mathbb{R}^4$ is a vector which contains the four temperature measurements (normalized deviations) along the reactor, $\hat{\mathbf{y}}(k)$ is the estimate of $\mathbf{y}(k)$, $\mathbf{L}_{\text{a}}$ and $\mathbf{L}_{\text{d}}$ are the submatrices of the observer gain (Kalman gain), $\tilde{\mathbf{F}}_{\text{C}}$ and $\tilde{\mathbf{F}}_{\text{T}}$ are the column vectors of $\tilde{\mathbf{F}} = \left[\tilde{\mathbf{F}}_{\text{C}}, \tilde{\mathbf{F}}_{\text{T}}\right]$ and $\mathbf{C}_{\text{s}}$ is a selection matrix which selects the measured temperatures from the vector $\mathbf{x}_n(k)$.

The block diagram of this control system can be observed in Figure 3.11. The control horizon N_{c} was set to 10 samples and the prediction horizon N_{p} was selected according to the following criterion: "Prediction Horizon = Control Horizon + Largest Settling Time = 80 samples". $\mathbf{u}_{\min} = [-1.1825, -0.3759, -0.565]^T$ and $\mathbf{u}_{\max} = [0.3175, 1.1241, 0.935]^T$ were selected according to the input constraints of the process and the operating temperatures of the jackets, and the weighting matrices in this way: $\mathbf{Q} = \mathbf{I}_{20}$ and $\mathbf{R} = 110 \cdot \mathbf{I}_3$. The Kalman gain matrix was computed from the following covariance matrices: $\mathbf{R}_{\text{w}} = \mathbf{I}_{21}$, $\mathbf{R}_{\text{v}} = 10^{-6} \cdot \mathbf{I}_3$. As it was mentioned in the previous chapter, the diagonal of the covariance matrix of the measurement noise $\mathbf{R}_{\text{v}}$ contains the measured noise variance of each process output. In this case we assumed a variance value of 10^{-6} for each process output, and we used the covariance matrix of the process noise $\mathbf{R}_{\text{w}}$ to trade speed and robustness.

3.4.2 Second MPC control scheme (MPC-PV) - Formulation in terms of physical variables

Unlike the previous control system, in this scheme the formulation of the MPC controller is in terms of physical variables. These variables are the temperature of some selected points along the reactor and the concentration at the reactor outlet. This formulation makes the tuning procedure more intuitive and allows us to have more flexibility in the definition of the control objectives; for example, we can give more importance to the concentration at the reactor outlet than to the temperature of the selected points or vice versa. In this MPC formulation, the temperature constraint ($T(z,t) \leq$ 400 K) of the system is imposed in the selected points. It has been observed through simulations that only imposing the temperature constraint in some points, is sufficient for guaranteeing the satisfaction of the temperature constraint in the remaining points of the spatial domain.

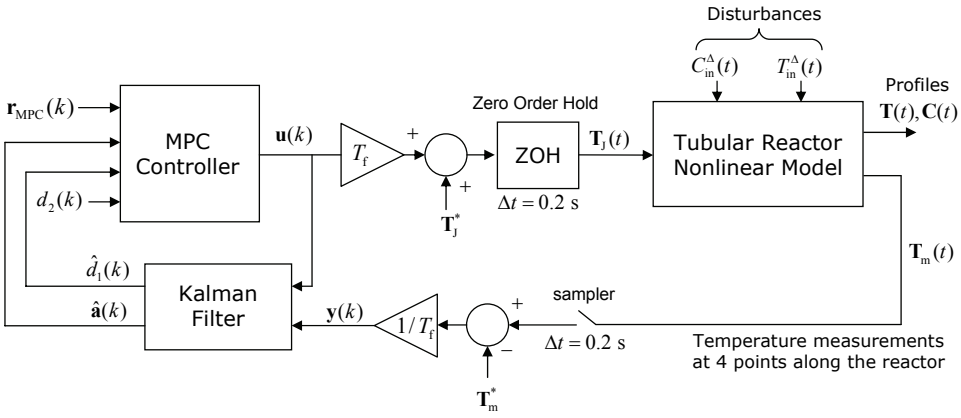


Figure 3.11: Block diagram of the MPC control systems. For the MPC-NTC case $\mathbf{r}_{\text{MPC}}(k) = \mathbf{a}_{\text{ref}}(k)$, and for the MPC-PV case $\mathbf{r}_{\text{MPC}}(k) = [\bar{C}_{N,\text{ref}}^{\Delta}(k), \bar{\mathbf{T}}_{\text{p,ref}}^{\Delta}(k)]^T$. In the diagram, $\mathbf{T}_{\text{J}}(t) = [T_{\text{J1}}(t), T_{\text{J2}}(t), T_{\text{J3}}(t)]^T$, $\mathbf{T}_{\text{J}}^{*} = [T_{\text{J1}}^{*}, T_{\text{J2}}^{*}, T_{\text{J3}}^{*}]^T$, $\mathbf{T}_{\text{m}}(t) \in \mathbb{R}^4$ contains the temperature measurements of the reactor and $\mathbf{T}_{\text{m}}^{*} \in \mathbb{R}^4$ contains their steady state values.

Since in this MPC formulation the temperature constraint along the reactor is taken into account, it is necessary to define a mechanism for handling the infeasibilities that can emerge due to the differences between the process and the model used by the MPC, the magnitude of the disturbances, the saturation of the actuators, etc. A way to deal with these infeasibilities is by softening the temperature constraint using a slack variables approach. A soft constraint formulation avoids infeasibilities problems by allowing violations in the temperature constraint, but at the same time it tries to minimize such violations by penalizing them in the objective function. In this MPC control system a slack variable approach with ℓ_∞-norm and time-dependent weights is used [68, 69]. The MPC controller is then formulated as follows:

$$\min_{\mathbf{w}_{N_\text{p}}, \Delta\mathbf{u}_{N_\text{c}}, \mathbf{d}_{N_\text{p}}, \boldsymbol{\xi}} \quad \sum_{i=1}^{N_\text{p}} \left(\left\| \bar{\mathbf{T}}_{\text{p,ref}}^{\Delta}(k+i) - \bar{\mathbf{T}}_{\text{p}}^{\Delta}(k+i) \right\|_{\mathbf{Q}_\text{T}}^2 + \right. \tag{3.18a}$$

$$\left. + \left\| \bar{C}_{N,\text{ref}}^{\Delta}(k+i) - \bar{C}_{N}^{\Delta}(k+i) \right\|_{Q_\text{C}}^2 \right) + \sum_{i=0}^{N_\text{c}-1} \|\Delta\mathbf{u}(k+i)\|_{\mathbf{R}}^2 + \boldsymbol{\xi}^T \mathbf{P}_\text{Q} \boldsymbol{\xi} + \mathbf{P}_\text{L}^T \boldsymbol{\xi}$$

subject to

$$\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \tilde{\mathbf{F}}\mathbf{d}(k+i), \;\; i = 0, \ldots, N_\text{p} - 1,$$

$$
\begin{aligned}
&\mathbf{d}(k+i+1) = \mathbf{d}(k+i), && i = 0, \ldots, N_\mathrm{p}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0, \ldots, N_\mathrm{c}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_\mathrm{c}, \ldots, N_\mathrm{p}-1,\\
&\bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k+i) = \mathbf{C}_{\mathrm{s}_1}\boldsymbol{\Phi}_n\mathbf{a}(k+i), && i = 1, \ldots, N_\mathrm{p},\\
&\bar{C}_N^{\Delta}(k+i) = \mathbf{C}_{\mathrm{s}_2}\boldsymbol{\Phi}_n\mathbf{a}(k+i), && i = 1, \ldots, N_\mathrm{p},\\
&\mathbf{u}_\mathrm{min} \le \mathbf{u}(k+i) \le \mathbf{u}_\mathrm{max}, && i = 0, \ldots, N_\mathrm{c}-1,\\
&\bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k+i) \le \bar{\mathbf{T}}_\mathrm{p}^{\Delta_\mathrm{max}} + \eta(i)\boldsymbol{\xi}, && i = 1, \ldots, N_\mathrm{p}, \quad (3.18b)\\
&\boldsymbol{\xi} \ge \mathbf{0}, && (3.18c)
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{w}(k) &= \left[\bar{C}_\mathrm{N}^{\Delta}(k); \bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k)\right]\\
\mathbf{w}_{N_\mathrm{p}} &= [\mathbf{w}(k+1); \mathbf{w}(k+2); \ldots; \mathbf{w}(k+N_\mathrm{p})]\\
\mathbf{d}_{N_\mathrm{p}} &= [\mathbf{d}(k+1); \mathbf{d}(k+2); \ldots; \mathbf{d}(k+N_\mathrm{p})]\\
\Delta\mathbf{u}_{N_\mathrm{c}} &= [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_\mathrm{c}-1)]
\end{aligned}
$$

where $\mathbf{Q}_\mathrm{T} \succeq 0$ and $\mathbf{R} \succ 0$ are weighting matrices, $Q_\mathrm{C} > 0$ is a weighting factor, $\|\mathbf{v}\|_{\mathbf{Q}}^2$ denotes $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, N_p is the prediction horizon, N_c is the control horizon, $\mathbf{u}_\mathrm{min}$ and $\mathbf{u}_\mathrm{max}$ are the lower and upper bounds of $\mathbf{u}(k)$, $\bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k)$ is a vector which contains the normalized deviations of the temperature of the selected points, $\bar{C}_N^{\Delta}(k)$ is the normalized deviation of the concentration at the reactor outlet, $\bar{\mathbf{T}}_{\mathrm{p,ref}}^{\Delta}(k)$ and $\bar{C}_{N,\mathrm{ref}}^{\Delta}(k)$ are the references for $\bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k)$ and $\bar{C}_N^{\Delta}(k)$ respectively, $\bar{\mathbf{T}}_\mathrm{p}^{\Delta_\mathrm{max}}$ is a vector which contains the maximum allowed temperatures for the selected points, the inequality (3.18b) is the temperature constraint, $\mathbf{C}_{\mathrm{s}_1}$ and $\mathbf{C}_{\mathrm{s}_2}$ are matrices for extracting $\bar{\mathbf{T}}_\mathrm{p}^{\Delta}(k)$ and $\bar{C}_N^{\Delta}(k)$ from $\mathbf{x}_n(k) = \boldsymbol{\Phi}_n\mathbf{a}(k)$, $\mathbf{P}_\mathrm{L}$ is a weighting vector and $\mathbf{P}_\mathrm{Q}$ is a weighting matrix, $\boldsymbol{\xi} \in \mathbb{R}^7$ is the vector of the slack variables, and $\eta(i) = 1/r_\mathrm{c}^{i-1}$, is a time-dependent weight ($r_\mathrm{c} > 1$).

In this formulation, we are penalizing the maximum violation of the temperature constraint along the prediction horizon for each selected point by means of the term $\boldsymbol{\xi}^T\mathbf{P}_\mathrm{Q}\boldsymbol{\xi} + \mathbf{P}_\mathrm{L}^T\boldsymbol{\xi}$. A sufficiently large $\mathbf{P}_\mathrm{L}$ will ensure that the constraints are enforced as exact soft constraints, that is, that constraint violations will only occur when there is no feasible solution of the original problem [68]. The quadratic term $\boldsymbol{\xi}^T\mathbf{P}_\mathrm{Q}\boldsymbol{\xi}$ is used as an additional tuning parameter and it also leads to a well-posed quadratic program (positive definite Hessian) [131].

The time-dependent weight $\eta(i)$ penalizes future predicted constraint violations increasingly, avoiding long-lasting constraint violations [68].

For estimating the state vector $\mathbf{a}(k)$ and the changes in the concentration of the feed flow ($d_1(k) = \bar{C}^{\Delta}_{\text{in}}(k)$), this control scheme employs the same observer (see Equation (3.17)) used by the previous control system. In fact, the block diagrams of both control schemes are practically the same (see Figure (3.11)), the only difference is the reference vector and of course the internal formulation of the MPC controller.

The points where the temperature is controlled were found by trial and error. Seven points were used and they correspond to the following sections of the reactor ($N = 300$ sections): $\mathcal{P} = \{100, 122, 150, 200, 250, 270, 300\}$.

The parameters of the MPC controller were set as follows: $N_{\text{c}} = 10$ samples, $N_{\text{p}} = 80$ samples, $r_{\text{c}} = 1.2$, $\mathbf{P}_{\text{L}} \in \mathbb{R}^7 = 10^4 \cdot [1, 1, \ldots, 1]^T$, $\mathbf{P}_{\text{Q}} \in \mathbb{R}^{7\times 7} = \mathbf{I}_7$, $\mathbf{Q}_{\text{T}} = \mathbf{I}_7$, $Q_{\text{C}} = 1000$, $\mathbf{R} = 110 \cdot \mathbf{I}_3$. The references for the variables were selected in this way: $\bar{\mathbf{T}}^{\Delta}_{\text{p,ref}} = \mathbf{0}$, and $\bar{C}^{\Delta}_{N,\text{ref}} = -5 \cdot 10^{-3}$ (this corresponds to a reduction of 6.5% in the concentration at the reactor outlet). Notice that in this tuning, it is considered more important to bring the concentration at the reactor outlet to its desired value than to bring the temperatures of the selected points to their references.

3.5 Simulation results

In order to perform the closed-loop simulations of the control systems described in the previous sections, the nonlinear model of the process given in (3.1) was discretized in space by replacing the partial derivatives with respect to space by backward difference approximations [41,138], leading to the following set of nonlinear ODEs:

$$\begin{aligned} \frac{d\bar{C}_i}{dt} &= -\frac{v}{\Delta z}\left(\bar{C}_i - \bar{C}_{i-1}\right) - k_0 \bar{C}_i e^{-\frac{E}{R T_{\text{f}} \bar{T}_i}} \\ \frac{d\bar{T}_i}{dt} &= -\frac{v}{\Delta z}\left(\bar{T}_i - \bar{T}_{i-1}\right) + G_{\text{r}} \frac{C_{\text{f}}}{T_{\text{f}}} \bar{C}_i e^{-\frac{E}{R T_{\text{f}} \bar{T}_i}} + H_{\text{r}}\left(\bar{T}_{\text{w},i} - \bar{T}_i\right) \end{aligned} \tag{3.19}$$

$$\text{for } i = 1, 2, \ldots, N$$

with

$$\bar{T}_{\mathrm{w,i}} = \begin{cases} \bar{T}_{\mathrm{J1}} = T_{\mathrm{J1}}/T_{\mathrm{f}}, & \forall i = 1, \ldots, z_{\mathrm{a}} \\ \bar{T}_{\mathrm{J2}} = T_{\mathrm{J2}}/T_{\mathrm{f}}, & \forall i = z_{\mathrm{a}}+1, \ldots, z_{\mathrm{b}} \\ \bar{T}_{\mathrm{J3}} = T_{\mathrm{J3}}/T_{\mathrm{f}}, & \forall i = z_{\mathrm{b}}+1, \ldots, N \end{cases}$$

$$\bar{T}_0 = \frac{T_{\mathrm{in}}}{T_{\mathrm{f}}}$$

$$\bar{C}_0 = \frac{C_{\mathrm{in}}}{C_{\mathrm{f}}}$$

where $N = 300$ is the number of sections in which the reactor is divided, z_{a} and z_{b} are the reactor sections defining the ending of the first and second jacket respectively, T_{f} and C_{f} are normalization factors, $\bar{C}_i = C_i/C_{\mathrm{f}}$ and $\bar{T}_i = T_i/T_{\mathrm{f}}$ are the normalized concentration and temperature of the ith section of the reactor, $\bar{T}_{\mathrm{J1}}, \bar{T}_{\mathrm{J2}}, \bar{T}_{\mathrm{J3}}$ are the normalized jackets temperatures, and Δz is the length of each section. As it was mentioned before, the variables are normalized in order to avoid possible numerical problems.

On one hand, the use of low order approximations for the spatial derivatives is known to produce excessive smoothing of the profiles due to *numerical diffusion*, and on the other hand, high-order approximations lead to excessive non-physical oscillations due to *numerical dispersion* [95]. Notice that both numerical diffusion and dispersion are two kinds of computational errors that occur as a result of the discretization process, and therefore they should not be confused with their physical counterparts. One way of decreasing these undesirable effects is by increasing the grid density (finer grid), but this measure leads to an increment of the computational burden. So, a trade-off between computational time and accuracy must be found. At the beginning, we divided the reactor into $N = 100$, $N = 300$, $N = 500$ and $N = 1000$ sections, and we found that a partition of 300 sections provides a good trade-off. Alternatively, nonlinear methods like *slope or flux limiters* and *adaptive grids* (see [95,144,147] for details about these two nonlinear methods) might be used in order to mitigate the numerical diffusion and dispersion.

Initially, in order to compare and evaluate the performance of the control systems, the following tests were carried out:

- Test 1: the temperature and concentration of the feed flow are increased by 10 K and 10^{-3} mol/l respectively.
- Test 2: the temperature and concentration of the feed flow are decreased by 10 K and 10^{-3} mol/L respectively.

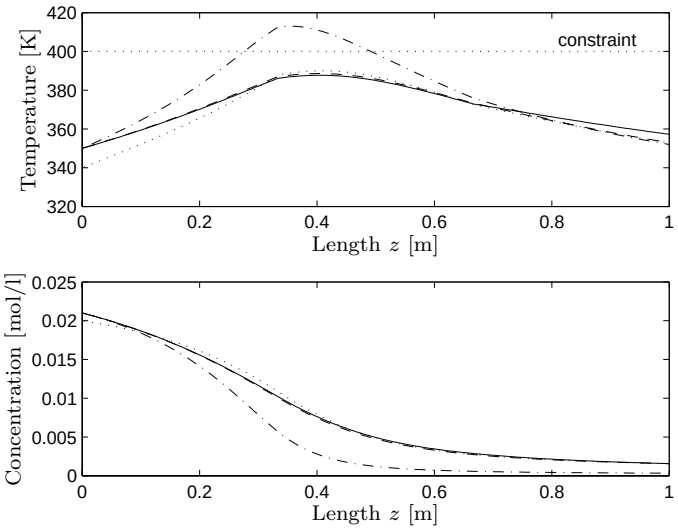


Figure 3.12: Steady-state temperature and concentration profiles of the reactor for Test 1. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

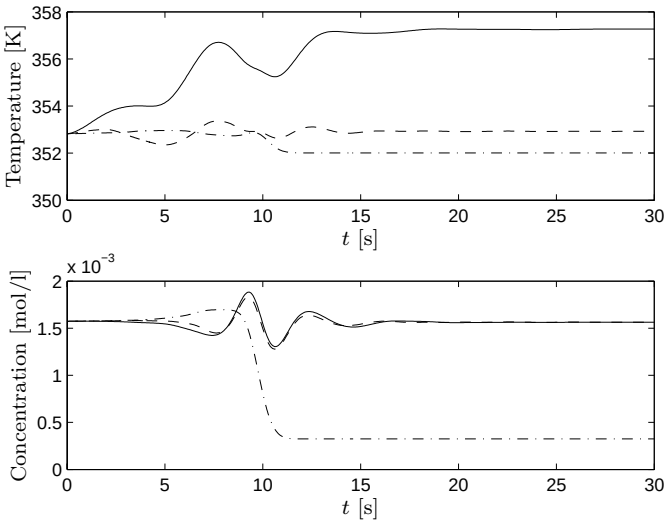


Figure 3.13: Temperature and concentration at the reactor outlet during Test 1. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

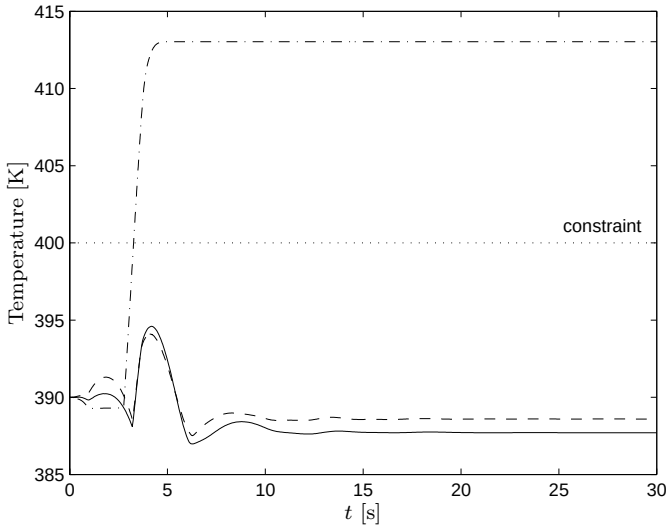


Figure 3.14: Maximal peak of the temperature profile during Test 1. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

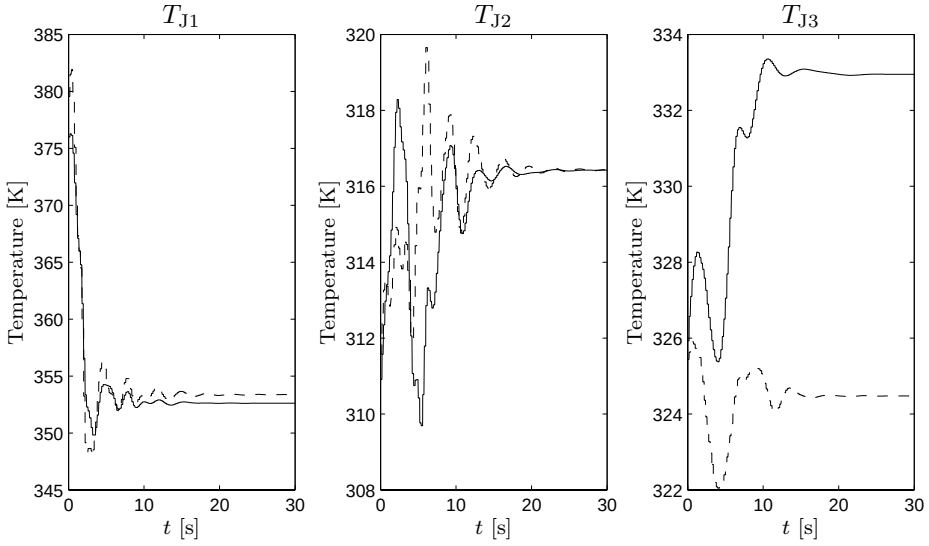


Figure 3.15: Control actions (jackets temperatures) of the MPC controllers along Test 1. Dashed line - MPC-NTC. Solid line - MPC-PV.

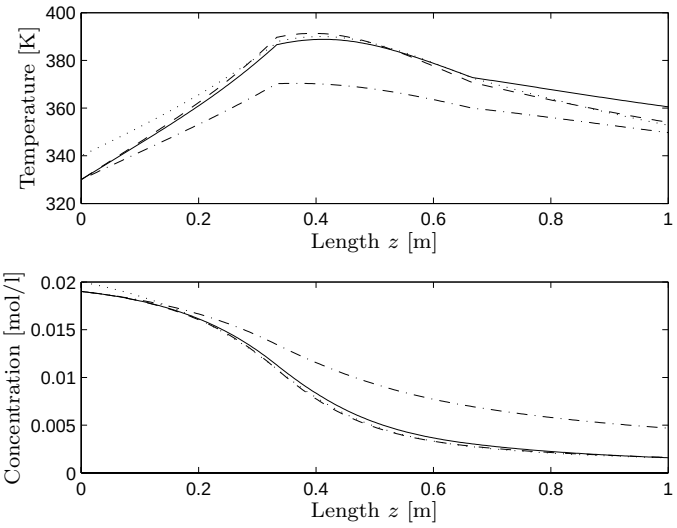


Figure 3.16: Steady-state temperature and concentration profiles of the reactor for Test 2. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

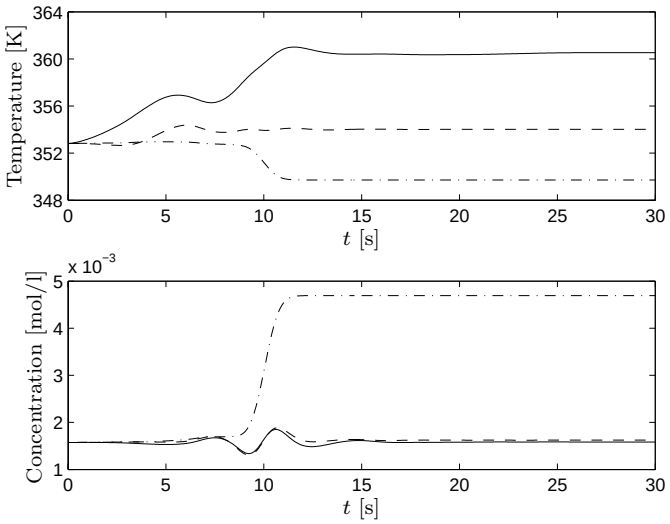


Figure 3.17: Temperature and concentration at the reactor outlet during Test 2. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

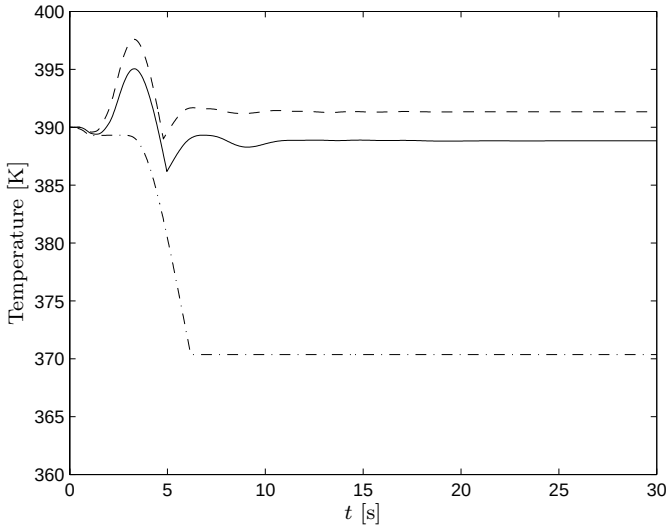


Figure 3.18: Maximal peak of the temperature profile during Test 2. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

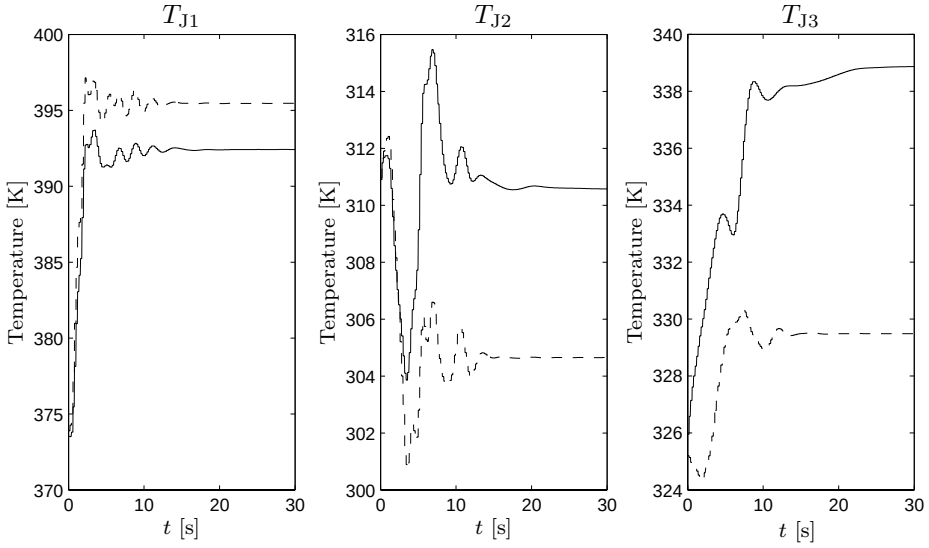


Figure 3.19: Control actions (jackets temperatures) of the MPC controllers along Test 2. Dashed line - MPC-NTC. Solid line - MPC-PV.

For carrying out the simulations, we have used as ODE solver the function *ODE45* of Matlab. This function is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair [43]. It is a one step solver that has been configured with a variable integration step and with a relative tolerance of 10^{-5}. In order to solve the optimization problem of the MPC controllers, the solver *Quadprog* has been used. *Quadprog* is part of the Optimization Toolbox of Matlab [101], and it uses an active set method similar to that described in [57].

The simulation results of Test 1 are presented in Figures 3.12, 3.13, 3.14 and 3.15. Figures 3.16, 3.17, 3.18 and 3.19 show the simulation results of Test 2. Furthermore in Table 3.3, some numbers for quantifying the performance of the control systems are shown. In this table, $T_{\max}$ is the maximum temperature reached inside the reactor during the test. ΔC_{out} is the percentage of change of the concentration in steady state at the reactor output with respect to its nominal value. That is,

$$\Delta C_{\text{out}}(\%) = \frac{C_N - C_N^*}{C_N^*} \times 100 \tag{3.20}$$

where C_N^* is the nominal value ($1.5737 \cdot 10^{-3}$ mol/l) and C_N is the concentration at the reactor output in steady state after the test.

In Test 1, there is a permanent violation of the temperature constraint (see Figures 3.12 and 3.14) for the open loop case, which leads to undesirable side reactions. The maximum temperature value registered is 413.03 K, which is far from the maximum allowed (400 K). However this temperature increment in steady state conduces to a reduction of 79.39 % in the concentration at the reactor output. For the case of the MPC control schemes, the temperature inside the reactor is kept below 400 K, and the concentration at reactor outlet is reduced a little bit. The control efforts of the MPC controllers can be observed in Figure 3.15, where it is clear that their control actions are all the time within the boundaries ($280 \text{ K} \leq T_{\text{J1}}, T_{\text{J2}}, T_{\text{J3}} \leq 400 \text{ K}$).

In Test 2, a significant increment of the concentration at the reactor outlet is observed when the system is operating in an open loop configuration (see Figures 3.16 and 3.17). We have an increment of 198.25% in steady state, which means that the performance of the tubular chemical reactor has been reduced dramatically. The MPC controllers overcome this situation by manipulating the temperatures of the jackets in such a way that the temperature inside the reactor is increased in order to compensate the effect of the input disturbances. We have an increment in the concentration of only

Table 3.3: Performance parameters of the control systems

Test	Open-loop		MPC-NTC		MPC-PV	
	ΔC_{out}[%]	$T_{\max}$[K]	ΔC_{out}[%]	$T_{\max}$[K]	ΔC_{out}[%]	$T_{\max}$[K]
1	-79.39	413.03	-0.69	394.09	-0.7	394.59
2	198.25	390	3.19	397.59	0.63	395.05
3	-99.46	440.46	52.38	405.08	-6.89	396.62
4	493.7	390	134	399.67	-54.12	399.34

$T_{\max}$ [K] = Maximum temperature reached inside the reactor during the test.
ΔC_{out} [%] = Percentage of change of the concentration in steady state at the reactor output with respect to its nominal value (see Equation (3.20)).

3.19% and 0.63% for the MPC-NTC and MPC-PV controllers respectively. It is clear that the performance of the reactor practically has not been affected. Additionally as it is shown in Figure 3.19, the control actions of both controllers are within the limits.

In general, the control schemes showed a good behavior for rejecting the disturbances (typical magnitudes: $C_{\text{in}} = \pm 10^{-3}$ mol/l and $T_{\text{in}} = \pm 10$ K) and both presented a similar performance.

Under the previous tests, the control systems were not operating close to the temperature constraints, and therefore during the tests, these constraints are not active in the MPC-PV controller. So, in order to evaluate the ability of this control scheme of dealing with the temperature constraint along the reactor, the following tests were designed:

- Test 3: the temperature and concentration of the feed flow are increased by 24 K and $3 \cdot 10^{-3}$ mol/l respectively.
- Test 4: the temperature and concentration of the feed flow are decreased by 30 K and $4 \cdot 10^{-3}$ mol/l respectively.

Notice that under these new tests (where the disturbances are too large in comparison with the typical ones) the tubular reactor operates far from the operating profiles shown in Figure 3.4, and therefore the differences between the nonlinear model of the process and the linear POD model used by the controllers are considerable.

The simulation results of Test 3 are given in Figures 3.20, 3.21, 3.22 and

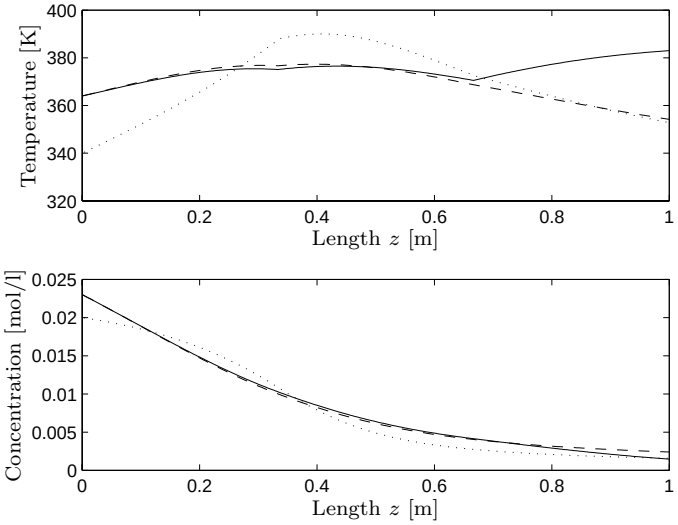


Figure 3.20: Steady-state temperature and concentration profiles of the reactor for Test 3. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV.

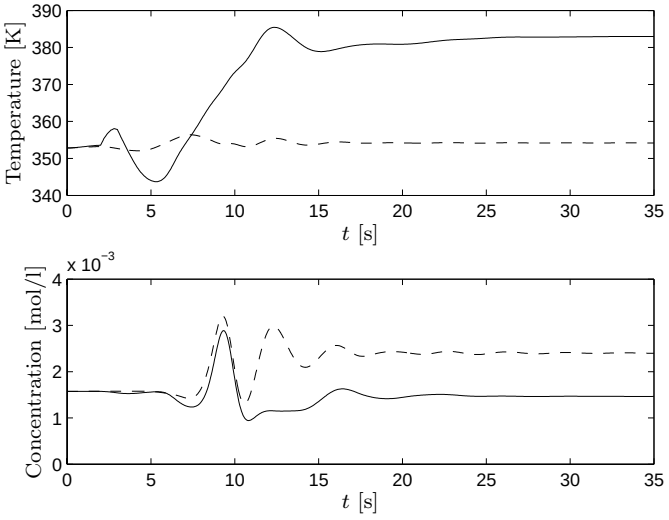


Figure 3.21: Temperature and concentration at the reactor outlet during Test 3. Dashed line - MPC-NTC. Solid line - MPC-PV.

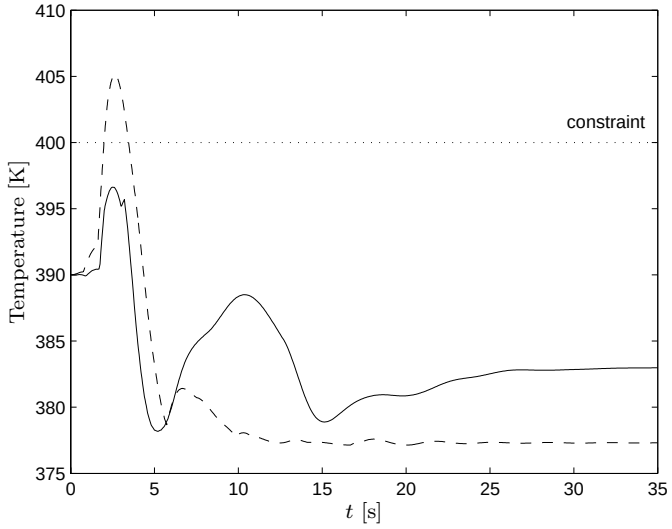


Figure 3.22: Maximal peak of the temperature profile during Test 3. Dashed line - MPC-NTC. Solid line - MPC-PV.

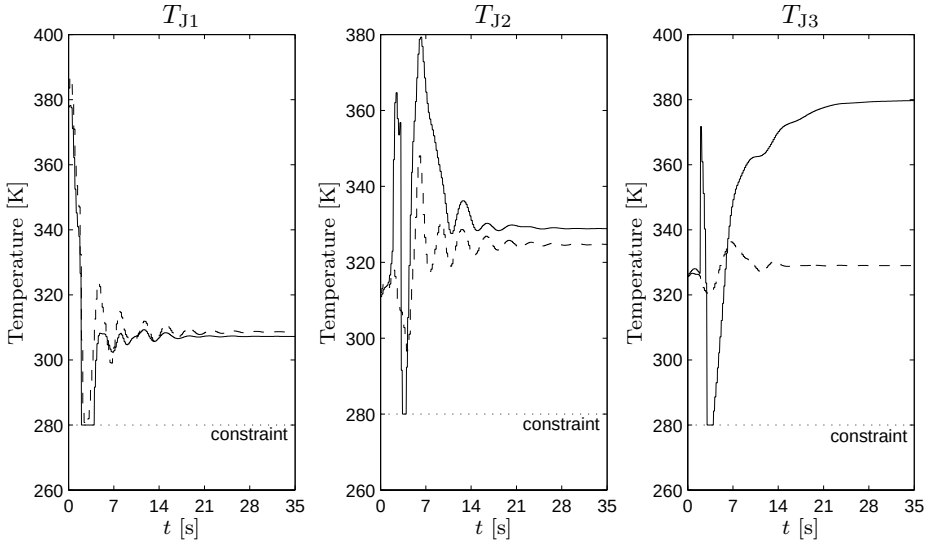


Figure 3.23: Control actions (jackets temperatures) of the MPC controllers along Test 3. Dashed line - MPC-NTC. Solid line - MPC-PV.

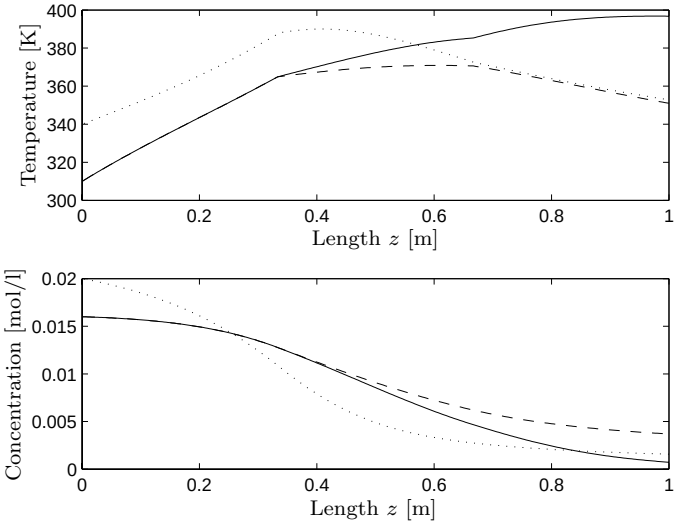


Figure 3.24: Steady-state temperature and concentration profiles of the reactor for Test 4. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV.

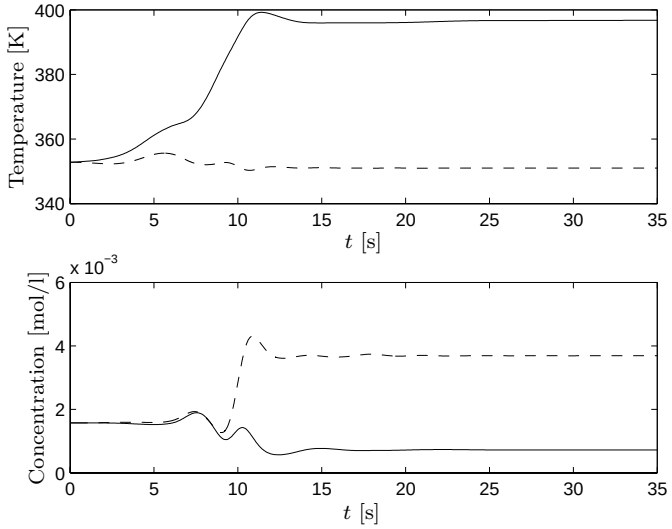


Figure 3.25: Temperature and concentration at the reactor outlet during Test 4. Dashed line - MPC-NTC. Solid line - MPC-PV.

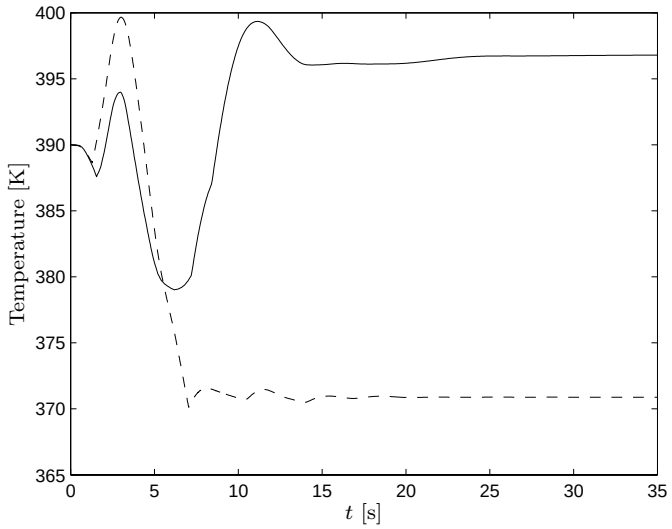


Figure 3.26: Maximal peak of the temperature profile during Test 4. Dashed line - MPC-NTC. Solid line - MPC-PV.

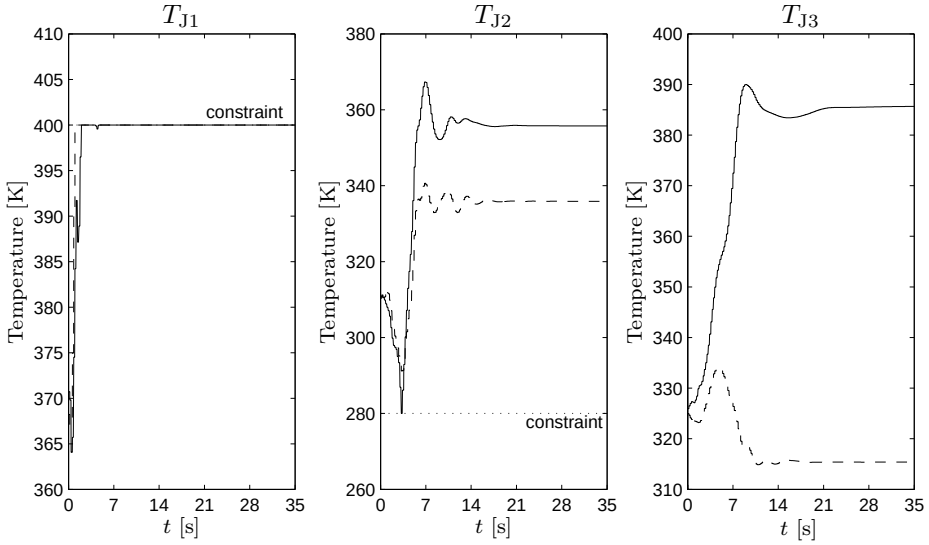


Figure 3.27: Control actions (jackets temperatures) of the MPC controllers along Test 4. Dashed line - MPC-NTC. Solid Line - MPC-PV.

3.23, and Figures 3.24, 3.25, 3.26 and 3.27 present the simulation results of Test 4. In addition, Table 3.3 shows the performance parameters of the control systems.

In Test 3, the MPC-NTC controller violates temporarily the temperature constraint as can be observed from Figure 3.22. The constraint is violated during 1.46 s and the maximum peak registered is 405.08 K. On the other hand the MPC-PV controller keeps all the time the temperature profile below 400 K. It is important to keep in mind that MPC-NTC does not take into account the temperature constraint and the results obtained with it were expected. From Table 3.3, it is clear that the performance of the reactor is degraded in a considerable way (the concentration at the reactor outlet is increased by 52.38%) for the MPC-NTC case, whereas for the MPC-PV case the performance of the reactor is increased. Notice in Figure 3.20, how the last part (particularly when $z > 0.667$ m) of the temperature profiles is different. The temperature profile of the MPC-PV control scheme is over the profile of the MPC-NTC control system allowing a good reduction of the concentration at the reactor outlet (see also Figure 3.21) for the MPC-PV case. This behavior is the consequence of giving more weight to the concentration ($Q_{\mathrm{C}} = 1000$) than to the temperature of the selected points ($\mathbf{Q}_{\mathrm{T}} = \mathbf{I}_7$) in the formulation of the MPC-PV controller. The control actions of both control systems are displayed in Figure 3.23. There are some temporary saturations of the actuators as a consequence of the large disturbances, but the control efforts are all the time between 280 K and 400 K.

In Test 4, there is no violation of the temperature constraint (see Figure 3.26) when both MPC control systems are used. It is remarkable how MPC-PV reduces the concentration at the reactor outlet by 54.12%. This is not the case for the MPC-NTC controller where the concentration at the reactor output is increased by 134%. In order to compensate the considerable decrement in the temperature of the feed flow, both controllers try to increase the temperature of the first jacket as much as possible, in this case until 400 K (actuator upper limit). This can be observed in Figure 3.27.

From the previous two tests, it is evident that MPC-PV notably outperforms MPC-NTC when large disturbances are applied to the reactor. At this point it is important to remark that the MPC-PV controller was somehow favored by the differences between the nonlinear model of the process and the linear POD model on which it is based. This observation can be corroborated by the fact that the controller was set to reduce the concentration at the reactor

outlet by 6.5% and not by 54.12% as happened in Test 4 for instance. It is remarkable how the MPC-PV controller can deal with the temperature constraint of the system in spite of using a linear POD model of the tubular reactor.

3.5.1 Tests on a reactor with axial dispersion

In order to include the axial dispersion phenomena in the plug flow reactor model (3.1), second-order dispersion terms have to be added as follows:

$$\frac{\partial C}{\partial t} = D_{\mathrm{C}}\frac{\partial^2 C}{\partial z^2} - v\frac{\partial C}{\partial z} - k_0 C e^{-\frac{E}{RT}} \tag{3.21a}$$

$$\frac{\partial T}{\partial t} = D_{\mathrm{T}}\frac{\partial^2 C}{\partial z^2} - v\frac{\partial T}{\partial z} + G_{\mathrm{r}} C e^{-\frac{E}{RT}} + H_{\mathrm{r}}(T_{\mathrm{w}} - T) \tag{3.21b}$$

where D_{C} and D_{T} are the mass and energy dispersion coefficients in $[\mathrm{m}^2/\mathrm{s}]$. Note, however, that in practice the dimensionless mass and energy Peclet numbers, i.e., $P_{e_{\mathrm{C}}} = vL/D_{\mathrm{C}}$, and $P_{e_{\mathrm{T}}} = vL/D_{\mathrm{T}}$, are mostly used for indicating the level of dispersion. The boundary conditions of the previous PDEs are the classical Danckwerts boundary conditions [39] given by

$$D_{\mathrm{C}}\frac{\partial C}{\partial z} = v\,(C - C_{\mathrm{in}}) \qquad \text{at } z = 0 \tag{3.22a}$$

$$D_{\mathrm{T}}\frac{\partial T}{\partial z} = v\,(T - T_{\mathrm{in}}) \qquad \text{at } z = 0 \tag{3.22b}$$

$$\frac{\partial C}{\partial z} = 0 \qquad \text{at } z = L \tag{3.22c}$$

$$\frac{\partial T}{\partial z} = 0 \qquad \text{at } z = L. \tag{3.22d}$$

For carrying out the closed-loop simulations of the MPC-NTC and MPC-PV controllers with the dispersive plug flow reactor model, the nonlinear model (3.21) was discretized in space by using a finite difference method. So, the second and first partial derivatives with respect to z were replaced by central difference approximations and backward difference approximations respectively, leading to the following set of nonlinear ODEs:

$$\frac{d\bar{C}_i}{dt} = \frac{D_{\mathrm{C}}}{\Delta z^2}\left(\bar{C}_{i+1} + \bar{C}_{i-1} - 2\bar{C}_i\right) - \frac{v}{\Delta z}\left(\bar{C}_i - \bar{C}_{i-1}\right) - k_0 \bar{C}_i e^{-\frac{E}{RT_{\mathrm{f}}\bar{T}_i}} \tag{3.23}$$

$$\frac{d\bar{T}_i}{dt} = \frac{D_{\mathrm{T}}}{\Delta z^2}\left(\bar{T}_{i+1} + \bar{T}_{i-1} - 2\bar{T}_i\right) - \frac{v}{\Delta z}\left(\bar{T}_i - \bar{T}_{i-1}\right) + \\ + G_{\mathrm{r}}\frac{C_{\mathrm{f}}}{T_{\mathrm{f}}}\bar{C}_i e^{-\frac{E}{RT_{\mathrm{f}}\bar{T}_i}} + H_{\mathrm{r}}\left(\bar{T}_{\mathrm{w},i} - \bar{T}_i\right)$$

$$\text{for } i = 1, 2, \ldots, N$$

with

$$\bar{T}_{\mathrm{w,i}} = \begin{cases} \bar{T}_{\mathrm{J1}} = T_{\mathrm{J1}}/T_{\mathrm{f}}, & \forall i = 1, \ldots, z_{\mathrm{a}} \\ \bar{T}_{\mathrm{J2}} = T_{\mathrm{J2}}/T_{\mathrm{f}}, & \forall i = z_{\mathrm{a}}+1, \ldots, z_{\mathrm{b}} \\ \bar{T}_{\mathrm{J3}} = T_{\mathrm{J3}}/T_{\mathrm{f}}, & \forall i = z_{\mathrm{b}}+1, \ldots, N \end{cases}$$

$$\bar{T}_0 = \frac{1}{D_{\mathrm{T}} + v\Delta z}\left(v\Delta z \frac{T_{\mathrm{in}}}{T_{\mathrm{f}}} + D_{\mathrm{T}}\bar{T}_1\right)$$

$$\bar{C}_0 = \frac{1}{D_{\mathrm{C}} + v\Delta z}\left(v\Delta z \frac{C_{\mathrm{in}}}{C_{\mathrm{f}}} + D_{\mathrm{C}}\bar{C}_1\right)$$

$$\bar{T}_{N+1} = \bar{T}_N$$

$$\bar{C}_{N+1} = \bar{C}_N$$

where $N = 300$ is the number of sections in which the reactor is divided, z_{a} and z_{b} are the reactor sections defining the ending of the first and second jacket respectively, T_{f} and C_{f} are normalization factors, $\bar{C}_i = C_i/C_{\mathrm{f}}$ and $\bar{T}_i = T_i/T_{\mathrm{f}}$ are the normalized concentration and temperature of the ith section of the reactor, $\bar{T}_{\mathrm{J1}}, \bar{T}_{\mathrm{J2}}, \bar{T}_{\mathrm{J3}}$ are the normalized jackets temperatures, and Δz is the length of each section.

For the sake of simplicity, in this study it is assumed that the mass and energy dispersion coefficients are equal, that is, $D_{\mathrm{C}} = D_{\mathrm{T}} = D$ ($P_{e_{\mathrm{C}}} = P_{e_{\mathrm{T}}} = P_e$). In Figure 3.28 we can observe the effect of increasing the dispersion coefficient D on the steady-state profiles of the reactor derived by means of the optimization algorithm presented in Section 3.2.2 (in Figure 3.28, case when $D = 0$ or equivalently $P_e = \infty$). It is clear that as the dispersion coefficient increases, the dispersion induces lower temperatures inside the reactor, which decreases the magnitude of the hot spot but inevitably degrades the reactor performance. For $D = 0.001$ m^2/s ($P_e = 100$) and $D = 0.005$ m^2/s ($P_e = 20$), the concentration at the reactor outlet is increased by 22.48% and 79.8% respectively (these increments are measured with respect to the concentration at the reactor outlet when $D = 0$). Notice also, that as the dispersion coefficient D grows, the difference between the feed values ($C_{\mathrm{in}} = 0.02$ mol/l and $T_{\mathrm{in}} = 340$ K) and the inlet values ($C(z = 0)$ and $T(z = 0)$) increases according to the Danckwerts boundary conditions.

Tests 1 and 2 were carried out again, but this time the predictive controllers were attached to the dispersive plug flow reactor model (3.21). The fact of using a dispersive tubular reactor constitutes a way of evaluating the

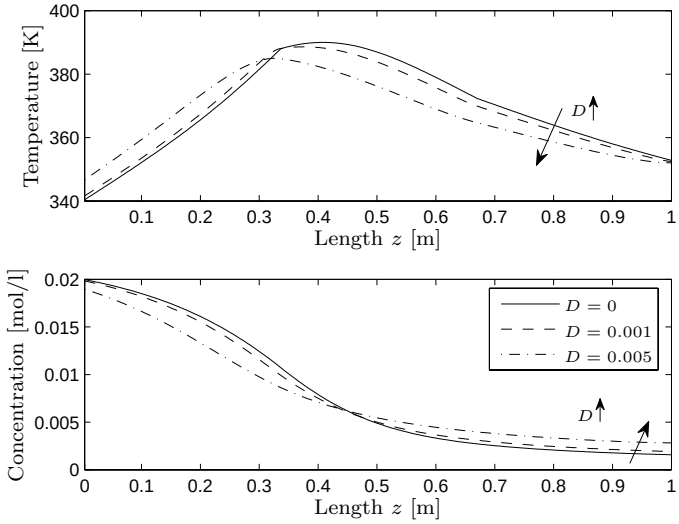


Figure 3.28: Effect of increasing the dispersion coefficient D [m^2/s] on the steady-state profiles of the reactor when $T_{J1} = 374.6$ K, $T_{J2} = 310.1$ K and $T_{J3} = 325.2$ K.

robustness of the MPC-NTC and MPC-PV controllers, since the model on which they are based does not consider the dispersion phenomena. Two levels of dispersion were used along the tests, $D = 0.001$ m^2/s and $D = 0.005$ m^2/s.

The simulation results of Test 1 for $D = 0.001$ m^2/s and $D = 0.005$ m^2/s are presented in Figures 3.29, 3.30, 3.31 and 3.32. Figures 3.33, 3.34, 3.35 and 3.36, show the simulation results of Test 2 for $D = 0.001$ m^2/s and $D = 0.005$ m^2/s. In addition, Table 3.4 presents some numerical values that quantify the performance of the control systems. The initial conditions of the dispersive reactor model are given by the profiles shown in Figure 3.28 for $D > 0$.

In Test 1, once more, we can observe a permanent violation of the temperature constraint for the open loop case (see Figures 3.29, 3.30, 3.31 and 3.32), which leads to undesirable side reactions. Notice, however, that the magnitude of this violation becomes smaller as the dispersion coefficient increases. Additionally, the increment of the level of dispersion increases the concentration at the reactor outlet. For the case of the MPC control

Table 3.4: Performance parameters of the control systems

Test	D	Open-loop		MPC-NTC		MPC-PV	
	[m^2/s]	ΔC_{out}	$T_{\max}$	ΔC_{out}	$T_{\max}$	ΔC_{out}	$T_{\max}$
1	0	-79.39	413.03	-0.69	394.09	-0.7	394.59
	0.001	-71.35	411.6	24.21	390.3	10.04	389.5
	0.005	-35.18	404.3	97.82	385.6	11.41	390.44
2	0	198.25	390	3.19	397.59	0.63	395.05
	0.001	211.4	388.6	38.5	390.48	8.43	389.13
	0.005	261.7	384.98	153.1	384.99	1.87	399.42

$T_{\max}$ [K] = Maximum temperature reached inside the reactor during the test.
ΔC_{out} [%] = Percentage of change of the concentration in steady state at the reactor outlet with respect to $1.5737 \cdot 10^{-3}$ mol/l (see Equation (3.20)).

schemes, the temperature inside the reactor is kept below 400 K. From Table 3.4, it is clear that the MPC-NTC controller is not as good as the MPC-PV controller for keeping the concentration at the reactor outlet close to its nominal value.

In Test 2, the reactor conversion is notably reduced when the system is operating in an open-loop configuration. This situation becomes more and more critical as the level of dispersion increases (see Figures 3.33 and 3.34 and Table 3.4). In order to counteract the effect of the input disturbances, the predictive controllers manipulate the jackets temperatures in such a way that the temperature inside the reactor is increased. From the results obtained, it is clear that the MPC-PV controller does a much better job than the MPC-NTC controller, which loses performance when the dispersion in the reactor is increased.

In general, the presence of dispersion in the reactor has had a much bigger impact on the performance of the MPC-NTC controller than on the performance of the MPC-PV controller. As the dispersion increases, the MPC-NTC controller loses its ability of keeping the concentration at the reactor outlet close to its nominal value. So, it is clear that for improving the behavior of this controller, it would be necessary to include the dispersion phenomena in the POD model on which this controller is based. Furthermore, the nonlinearities of the reactor contribute to the poor performance exhibited by the MPC-NTC controller, since under dispersive conditions the reactor can operate far away from the nominal operating

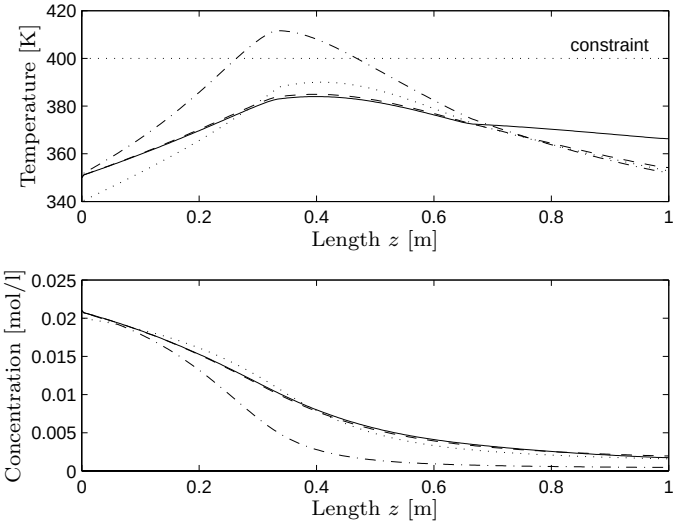


Figure 3.29: Steady-state profiles of the reactor with $D = 0.001$ m^2/s for Test 1. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

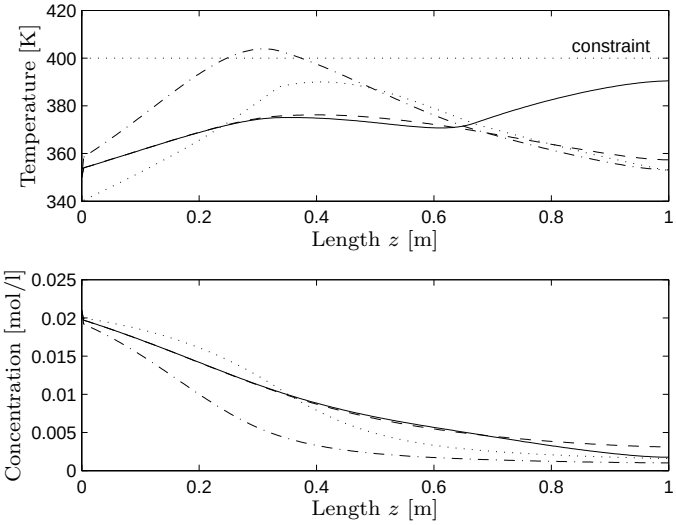


Figure 3.30: Steady-state profiles of the reactor with $D = 0.005$ m^2/s for Test 1. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

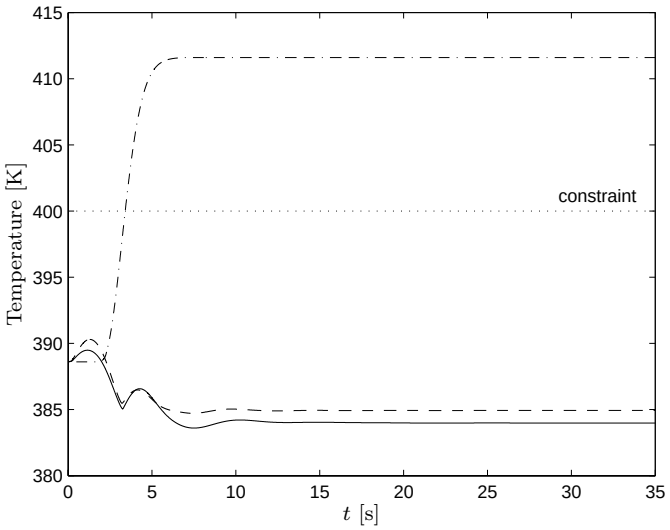


Figure 3.31: Maximal peak of the temperature profile during Test 1 when $D = 0.001$ m^2/s. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

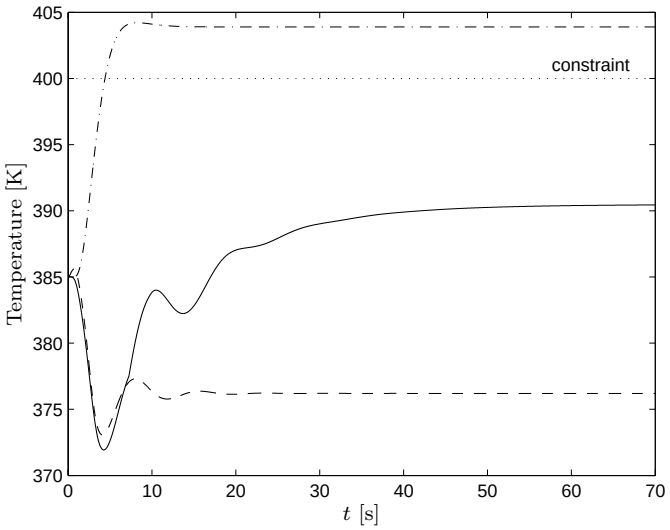


Figure 3.32: Maximal peak of the temperature profile during Test 1 when $D = 0.005$ m^2/s. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

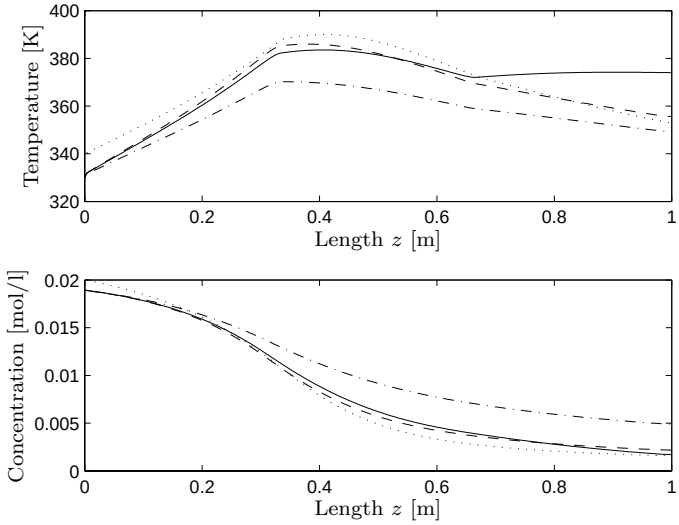


Figure 3.33: Steady-state profiles of the reactor with $D = 0.001$ m^2/s for Test 2. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

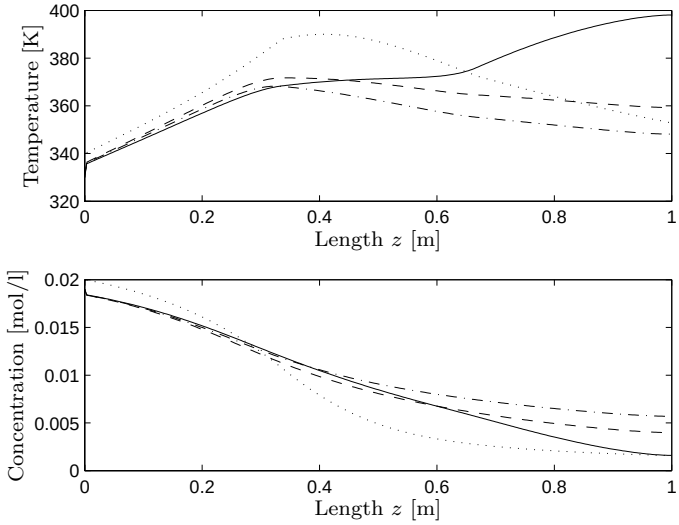


Figure 3.34: Steady-state profiles of the reactor with $D = 0.005$ m^2/s for Test 2. Dotted line - Nominal profile (reference). Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

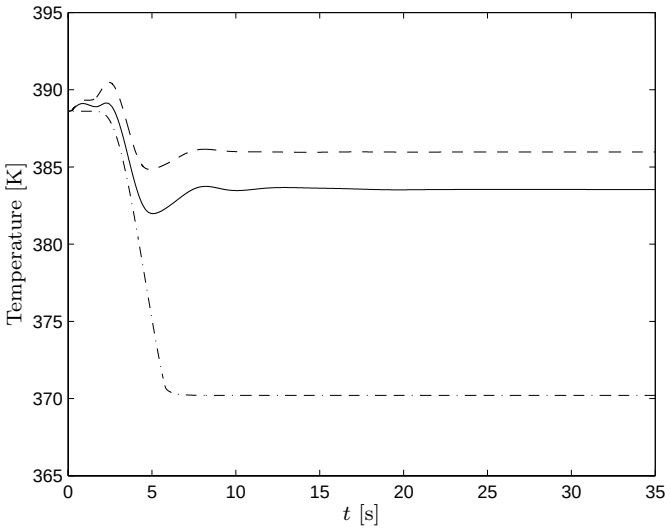


Figure 3.35: Maximal peak of the temperature profile during Test 2 when $D = 0.001$ m^2/s. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

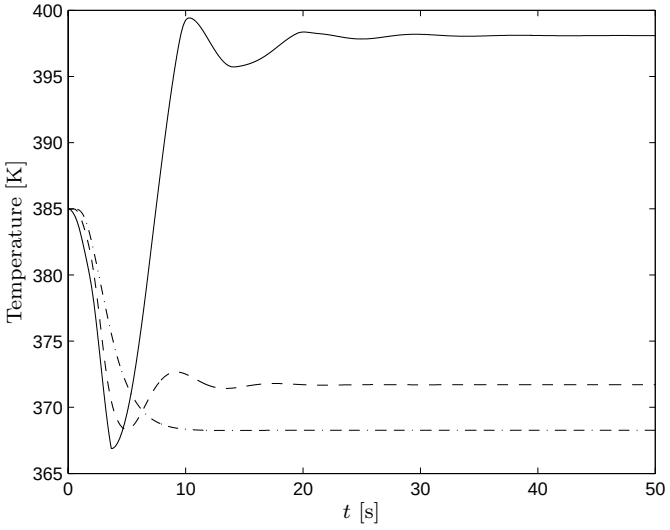


Figure 3.36: Maximal peak of the temperature profile during Test 2 when $D = 0.005$ m^2/s. Dashed line - MPC-NTC. Solid line - MPC-PV. Dash-dotted line - Open loop response.

profiles (see Figure 3.28, case when $D = 0$ m^2/s) where the linear POD model provides a suitable approximation of the nonlinear model of the system.

Observe in Figures 3.29, 3.30, 3.33 and 3.34, how the MPC-PV controller increases the temperature of the last sections of the reactor in order to guarantee a good performance of the system. As it was explained before, this behavior is the consequence of giving more importance to the reduction of the concentration at the rector outlet than to the reduction of the temperature deviations in the formulation of the predictive controller. It is remarkable how the MPC-PV controller is able to maintain the concentration at the reactor outlet close to its nominal value, in spite of using a POD model that does not incorporate both the dispersion effects and the nonlinearities of the system.

Although it is not shown in this section, the control efforts of the predictive controllers were all the time within the boundaries (280 K $\leq T_{J1}, T_{J2}, T_{J3} \leq$ 400 K).

3.6 Conclusions

In this chapter we have presented the results of applying POD and predictive control techniques to the control of the temperature and concentration profiles of a non-isothermal tubular reactor. In addition, we have introduced an optimization algorithm for deriving the operating profiles of the process. Thanks to the POD and Galerkin projection techniques, the high-dimensionality of the linearized model of the reactor has been significantly reduced making possible the control design. Two POD-based MPC control schemes have been proposed: a scheme where the formulation of the predictive controller is in terms of the POD coefficients (MPC-NTC) and a scheme where the MPC is formulated in terms of physical variables (MPC-PV). In the first case, the control of the reactor profiles is achieved indirectly by controlling the POD coefficients which have no physical meaning. This makes the tuning of the controller little intuitive and the definition of the control goals little flexible. This is not the case for the second MPC controller where its formulation is in terms of the temperature of some selected points along the reactor and the concentration at the reactor output. Taking advantage of the flexibility that this scheme provides, we gave more importance to the reduction of the concentration at

the reactor outlet than to the reduction of the temperature deviations. As a consequence of this setting, we observed in the simulation results how this control system outperforms the first one when large disturbances are applied to the plant. Notice however that under typical disturbances (Test 1 and 2), both control systems practically exhibit a similar performance. In tests with a dispersive tubular reactor, the second control scheme outperformed once more the MPC formulated in terms of the POD coefficients.

Unlike the first control scheme, the MPC formulated in terms of physical variables imposes the temperature constraint of the reactor ($T(z,t) \leq 400$ K) in some selected points of the spatial domain. It was argued that imposing the temperature constraint in some specific points might indirectly enforce this constraint in the remaining points given the smoothness of the temperature profile and the limited number of jackets. The simulation results showed that the controller was able to reject large disturbances without letting this constraint be violated. Nevertheless, the most difficult part in the tuning of this controller is the selection of the points where the temperature constraint is imposed. In this study, such selection was done by trial and error and therefore we had to carry out a lot of simulations in order to find a suitable set of points. This is of course not practical since a lot of time and computational resources are needed. In Section 4.4 of the next chapter, we present a way of finding a suitable set of points by means of a greedy selection algorithm. The reader is referred to the end of this section where we also discuss an improved formulation of the MPC-PV controller.

Given that the MPC-NTC controller does not take into account the temperature constraint of the reactor, a temporary violation of this constraint was observed during Test 3. In the next chapter, we propose an extension of this controller which incorporates the temperature constraint in its formulation. We will see that this new MPC will have to deal with a very large number of linear inequality constraints (24 000). Since such a large number of constraints demands a considerable amount of memory and computational time for solving the optimization problem of the MPC controller, in the next chapter we propose two approaches for reducing or approximating this large number of inequality constraints.

Finally, it is important to remark that the tuning of the controllers have been done in a conservative way in order to let them deal with large input disturbances. We have to keep in mind that under large disturbances in the feed flow (Tests 3 and 4), the reactor operates far from its nominal operating profiles and therefore the differences between the linear POD model used by

the controllers and the nonlinear model of the reactor are considerable. It should be clear that if a tight tuning is required, it would be necessary to incorporate the nonlinear characteristics of the reactor into the POD model.

Chapter 4

Constraint Handling

4.1 Introduction

In the previous chapter, two POD-based predictive controllers were proposed for controlling the temperature and concentration profiles of a non-isothermal tubular chemical reactor. The control goal was to reject the disturbances that affect the process, that is, the changes in the temperature and concentration of the feed flow. One important constraint of the system is that the temperature inside the reactor must be below a given value in order to prevent undesirable side reactions. Under typical disturbances, the controller formulated in terms of the POD coefficients, which is referred to as MPC-NTC (described by (3.16)), performs very well, and the temperature constraint is not violated despite the fact that the predictive controller does not incorporate this constraint in its formulation. However, if larger disturbances are applied, temporary violations of this constraint are observed.

In this chapter we start by presenting an extension of the MPC-NTC controller proposed in Section 3.4.1. This new POD-based controller takes into account the temperature constraint of the reactor and uses a slack variable approach with ℓ_∞-norm and time-dependent weights for handling the infeasibilities that can arise [68]. Given that POD only reduces the number of states and not the number of temperature constraints which usually is very large, the optimization problem posed by this MPC, demands a considerable amount of memory and requires more computational time.

In this chapter we present two techniques for reducing the number of temperature constraints. In the first technique, we approximate the feasible region delimited by the large set of temperature constraints (linear inequalities) by means of the theory of positive polynomials [1]. This approximation leads to a reduction in the number of constraints by replacing many linear inequalities by a few Linear Matrix Inequalities (LMIs) and a small number of linear equalities. In this method the MPC optimization problem is written as a Semidefinite Program (SDP). In the second approach, we propose a greedy algorithm that exploits the similarities between the coefficients of consecutive constraints for selecting a reduced set of them [5], and unlike the first approach the MPC optimization problem is written as a Quadratic Program (QP). These approaches are some of the main contributions of this dissertation.

Based on the polynomial approximation of the temperature constraints, and based on the reduced set of constraints found by the greedy algorithm, two new predictive controllers are devised. These controllers also incorporate the mechanism for handling infeasibilities mentioned before.

This chapter is structured as follows. Section 4.2 presents the extension of the MPC-NTC controller, that is, a POD-based MPC controller that deals with a very large number of temperature constraints. In Section 4.3, our method for approximating the temperature constraints by using the theory of positive polynomials is described, and a new MPC is presented. Furthermore, the basics of this theory are discussed and some simulation results are shown. Section 4.4 presents our greedy selection algorithm for reducing the number of temperature constraints. Additionally, we show some simulation results. Finally in Section 4.5, we summarize and conclude this chapter.

4.2 POD-based MPC controller with temperature constraints

In this section we present an extension of the control scheme described in Section 3.4.1, which takes into account the temperature constraint of the reactor and incorporates a mechanism for dealing with infeasibilities. As it was explained in Section 3.4.1, the control of the temperature and concentration profiles is achieved indirectly by controlling the POD

coefficients, whose references $\mathbf{a}_{\text{ref}}$ are given by

$$\mathbf{a}_{\text{ref}} = \mathbf{\Phi}_n^T \mathbf{x}_{\text{ref}}$$

where $\mathbf{x}_{\text{ref}}$ is the reference of the state vector $\mathbf{x}(t)$ and is equal to $\mathbf{0}$ since the control system has to keep the reactor operating around the profiles shown in Figure 3.4.

As it was stated in Section 3.4.2, the fact of taking into account the temperature constraint of the reactor, creates the necessity of establishing a mechanism for handling the infeasibilities that can come out due to the differences between the process and the model used by the MPC, the magnitude of the disturbances, the saturation of the actuators, and so on. In this MPC formulation, we treat the temperature constraint as a soft constraint by using a *slack variable* approach with ℓ_∞-norm and time-dependent weights [68,69]. This MPC, which will be referred to as MPC-QP, is formulated as follows:

$$\min_{\mathbf{a}_{N_\text{p}}, \Delta\mathbf{u}_{N_\text{c}}, \mathbf{d}_{N_\text{p}}, \xi} \sum_{i=1}^{N_\text{p}} \|\mathbf{a}_{\text{ref}}(k+i) - \mathbf{a}(k+i)\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_\text{c}-1} \|\Delta\mathbf{u}(k+i)\|_{\mathbf{R}}^2 + \quad (4.1a)$$

$$+P_\text{Q}\xi^2 + P_\text{L}\xi$$

subject to

$$\begin{aligned}
&\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \tilde{\mathbf{F}}\mathbf{d}(k+i), && i = 0, \ldots, N_\text{p} - 1,\\
&\mathbf{d}(k+i+1) = \mathbf{d}(k+i), && i = 0, \ldots, N_\text{p} - 1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0, \ldots, N_\text{c} - 1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_\text{c}, \ldots, N_\text{p} - 1,\\
&\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max}, && i = 0, \ldots, N_\text{c} - 1,\\
&\bar{\mathbf{T}}^{\Delta}(k+i) = \mathbf{\Phi}_\text{T}\mathbf{a}(k+i) \leq \bar{\mathbf{T}}^{\Delta_{\max}} + \mathbf{1}\cdot\eta(i)\xi, && i = 1, \ldots, N_\text{p}, \quad (4.1b)\\
&\xi \geq 0, && (4.1c)
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{a}_{N_\text{p}} &= [\mathbf{a}(k+1); \mathbf{a}(k+2); \ldots; \mathbf{a}(k+N_\text{p})]\\
\mathbf{d}_{N_\text{p}} &= [\mathbf{d}(k+1); \mathbf{d}(k+2); \ldots; \mathbf{d}(k+N_\text{p})]\\
\Delta\mathbf{u}_{N_\text{c}} &= [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_\text{c}-1)]
\end{aligned}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices ($\mathbf{Q} \succeq 0, \mathbf{R} \succ 0$), $\|\mathbf{v}\|_{\mathbf{Q}}^2$ denotes $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, N_p is the prediction horizon, N_c is the control horizon, $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ are the lower and upper bounds (these hard constraints are necessary due to physical limitations of the actuators) of $\mathbf{u}(k)$, $\mathbf{\Phi}_\mathrm{T}$ is the lower part (the last $N = 300$ rows) of the matrix $\mathbf{\Phi}_n = \left[\mathbf{\Phi}_\mathrm{C}^T, \mathbf{\Phi}_\mathrm{T}^T\right]^T$ that is associated to the temperature profile, $\bar{\mathbf{T}}^{\Delta}(k) = \mathbf{\Phi}_\mathrm{T}\mathbf{a}(k)$ is a vector which contains the normalized deviations of the temperature profile, $\bar{\mathbf{T}}^{\Delta_{\max}} = \left(400\,\mathrm{K}\cdot\mathbf{1} - \mathbf{T}^*\right)/T_\mathrm{f}$ is a vector that contains the maximal allowed temperature for each point of the reactor, ξ is the slack variable (a scalar quantity), P_Q and P_L are weighting factors ($P_\mathrm{Q} > 0$, $P_\mathrm{L} > 0$), $\mathbf{1} \in \mathbb{R}^{300}$ is a vector of 1's and $\eta(i) = 1/r_\mathrm{c}^{i-1}$, is a time-dependent weight ($r_\mathrm{c} > 1$).

In this formulation the maximum violation of the temperature constraint along the reactor and the prediction horizon is penalized by means of the term $P_\mathrm{Q}\xi^2 + P_\mathrm{L}\xi$. A sufficiently large P_L will ensure that the constraints are enforced as exact soft constraints, that is, that constraint violations will only occur when there is no feasible solution of the original problem [68]. The quadratic term $P_\mathrm{Q}\xi^2$ is used as an additional tuning parameter and it also leads to a well-posed quadratic program (positive definite Hessian) [131].

The time-dependent weight $\eta(i)$ is used for penalizing future predicted constraint violations increasingly, avoiding long-lasting constraint violations [68].

Similarly to the MPC controllers presented in the previous chapter, this controller is embedded in the same block diagram depicted in Figure 3.11 (case MPC-NTC), and it uses the Kalman filter described by (3.17) for estimating the state vector $\mathbf{a}(k)$ of the POD model (3.14) and the changes in the concentration of the feed flow ($d_1(k) = \bar{C}_\mathrm{in}^{\Delta}(k)$).

The control horizon N_c was set to 10 samples and the prediction horizon N_p was selected according to the following criterion: "Prediction Horizon = Control Horizon + Largest Settling Time = 80 samples". $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ were selected according to the input constraints of the process and the operating temperatures of the jackets. The other parameters were selected as follows: $r_\mathrm{c} = 1.2$, $P_\mathrm{L} = 10^4$, $P_\mathrm{Q} = 10^4$, $\mathbf{Q} = \mathbf{I}_{20}$ and $\mathbf{R} = 110 \cdot \mathbf{I}_3$.

The optimization problem (4.1) that is solved by the MPC controller is a quadratic programming (QP) problem which has $N \times N_\mathrm{p} = 300 \times 80 = 24\,000$ temperature constraints. This amount of constraints demands a considerable amount of memory and computing power. Although the POD

technique has reduced the number of state variables of the high-dimensional model (3.8), the number of temperature constraints is still very large.

In the next sections, we present two approaches for tackling this problem. Initially, we introduce an interesting approach which uses the positive polynomials theory for replacing the large number of inequalities by a few linear matrix inequalities while maintaining a control of the temperature at every point at the reactor (infinity number of points). Afterwards, we present a greedy selection algorithm which reduces the number of constraints by choosing properly only some of them. Note however that unlike the positive polynomial approach, in this method we do not have any command on the temperature between the discretization points.

4.3 Positive polynomial approach

4.3.1 Fundamentals

Many specific problems, particularly in systems and control, can be reduced to the verification of the global nonnegativity of a polynomial function [28]. In [117] some examples are provided, namely, Lyapunov function computation, output feedback stabilization, multidimensional system stability, etc.

The Tarski-Seidenberg decision procedure described in [27,28,106], provides an explicit algorithm for deciding if a polynomial is nonnegative, so it is clear that this problem is decidable. In addition, there are some alternative approaches based on decision algebra like the ones discussed in [18,28].

In general, the problem of testing the global nonnegativity of a multivariate polynomial function is in fact Nondeterministic Polynomial-time hard (NP-hard), when its degree is larger or equal than four [117]. So, in order to avoid the complexity problems of the previous methods which provide an exact solution to this problem, we have to look for relaxed conditions that can be efficiently tested while guaranteeing the global nonnegativity of the polynomial.

An obvious necessary condition for the nonnegativity of a polynomial is that its degree has to be even. Now, a simple sufficient condition for a real-valued polynomial $G(x_1, \ldots, x_l) = G(\mathbf{x}) : \mathbb{R}^l \to \mathbb{R}$ to be nonnegative everywhere is given by the existence of its Sum Of Squares (SOS) decomposition defined

as follows,

$$G(\mathbf{x}) = \sum_{j=1}^{m} g_j^2(\mathbf{x}) \tag{4.2}$$

where $g_j(\mathbf{x}) = g_j(x_1, \ldots, x_l)\ \forall j = 1, \ldots, m$ are real-valued polynomials in $x_1, \ldots, x_l$. Notice however that in general, SOS is not equivalent to the nonnegativity of a polynomial. Only in the following special cases they are equivalent: quadratic polynomials, univariate polynomials and quartic polynomials in two variables.

Further, the SOS representability of a polynomial can be expressed as a semidefinite feasibility problem [111,117], as the following proposition states.

Proposition 4.1. *A multivariate polynomial $G(\mathbf{x})$ of degree $2d$ is SOS if and only if there exists a positive semidefinite matrix $\mathbf{W}$ and a vector of monomials $\mathbf{h}(\mathbf{x})$ containing monomials in $x_1, \ldots, x_l$ of degree $\leq d$ such that*

$$G(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{W}\, \mathbf{h}(\mathbf{x}). \tag{4.3}$$

The number of components of $\mathbf{h}(\mathbf{x})$ is given by $\binom{d+l}{d}$ and the number of squares m in (4.2) is equal to the rank of $\mathbf{W}$, which in general is not unique.

Example 4.1 ([115,117])**.** *In this example, we are interested in finding out whether or not the quartic $(2d = 4)$ polynomial in two variables $G(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$ is SOS.*

First of all, let us define $\mathbf{h}(x_1, x_2)$ as follows: $\mathbf{h}(x_1, x_2) = \left[x_1^2, x_2^2, x_1x_2\right]^T$. We can write $G(x_1, x_2)$ in the following quadratic form:

$$\begin{aligned} G(x_1, x_2) &= 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \\ &= \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{12} & w_{22} & w_{23} \\ w_{13} & w_{23} & w_{33} \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix} \\ &= w_{11}x_1^4 + w_{22}x_2^4 + (2w_{12} + w_{33})x_1^2x_2^2 + 2w_{13}x_1^3x_2 + 2w_{23}x_1x_2^3. \end{aligned}$$

By equating the coefficients of $G(x_1, x_2)$ to the coefficients of the monomials obtained from expanding $\mathbf{h}(x_1, x_2)^T\mathbf{W}\ \mathbf{h}(x_1, x_2)$, we obtain the following set of affine relations:

$$w_{11} = 2,$$

$$
\begin{aligned}
w_{22} &= 5, \\
w_{13} &= 1, \\
w_{23} &= 0, \\
2w_{12} + w_{33} &= -1.
\end{aligned}
$$

A positive semidefinite matrix $\mathbf{W} \succeq 0$ that satisfies the previous linear equalities can then be found by using semidefinite programming. After solving numerically, we obtained the following particular solution:

$$
\begin{aligned}
w_{12} &= -3, \\
w_{33} &= 5, \\
\mathbf{W} &= \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} \succeq 0.
\end{aligned}
$$

This is a certificate that $G(x_1, x_2)$ is nonnegative everywhere since $G(x_1, x_2)$ is SOS. Given that the rank of $\mathbf{W}$ is 2, we have only 2 squares in the SOS decomposition. If we want to compute the SOS decomposition of $G(x_1, x_2)$ we proceed as follows. First we calculate the Cholesky factorization of $\mathbf{W}$,

$$
\mathbf{W} = \mathbf{U}_{\mathrm{c}}^T \mathbf{U}_{\mathrm{c}}, \text{ where } \mathbf{U}_{\mathrm{c}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix},
$$

and finally, we substitute the previous factorization into the quadratic form of $G(x_1, x_2)$,

$$
\begin{aligned}
G(x_1, x_2) &= \mathbf{h}(x_1, x_2)^T \mathbf{U}_{\mathrm{c}}^T \mathbf{U}_{\mathrm{c}}\, \mathbf{h}(x_1, x_2) \\
&= \|\mathbf{U}_{\mathrm{c}}\, \mathbf{h}(x_1, x_2)\|_2^2 \\
&= \left\| \frac{1}{\sqrt{2}} \begin{bmatrix} 2x_1^2 - 3x_2^2 + x_1 x_2 \\ x_2^2 + 3x_1 x_2 \end{bmatrix} \right\|_2^2 \\
&= \frac{1}{2} \left(2x_1^2 - 3x_2^2 + x_1 x_2\right)^2 + \frac{1}{2} \left(x_2^2 + 3x_1 x_2\right)^2.
\end{aligned}
$$

As it was mentioned at the beginning of this section, checking the nonnegativity of a polynomial $G(\mathbf{x})$ is an NP-hard problem when the degree of $G(\mathbf{x})$ is at least 4 [108], whereas checking whether $G(\mathbf{x})$ can be written as an SOS is computationally tractable, since it can be formulated as an SDP problem which in the worst case has a polynomial time complexity. Although we are not going into details, there are several results that suggest that this

relaxation is not too conservative [124]. Observe that when we increase the number of variables or the degree of $G(\mathbf{x})$, the computational complexity for testing whether or not $G(\mathbf{x})$ is SOS also increases. Even so, the complexity overload is still a polynomial function of these parameters [115].

4.3.2 Approximation of the temperature constraints by means of positive polynomials

In spite of the fact that the POD model of the reactor has 20 states, in this chapter a 2nd order POD model is used for visualizing the feasible regions delimited by the temperature constraints and explain the main ideas of both this positive polynomial approach and the technique based on the greedy selection algorithm.

Figure 4.1 shows in dashed line the feasible region delimited by the temperature constraints

$$\mathbf{\Phi}_{\mathrm{T}}\mathbf{a}(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}} \tag{4.4}$$

of a 2nd order POD model.

As it was mentioned before, $\mathbf{\Phi}_{\mathrm{T}} \in \mathbb{R}^{300\times 20}$ is the lower part of the matrix $\mathbf{\Phi}_n = \left[\mathbf{\Phi}_{\mathrm{C}}^T, \mathbf{\Phi}_{\mathrm{T}}^T\right]^T$, therefore each column of $\mathbf{\Phi}_{\mathrm{T}}$ corresponds to the part of the basis vectors $\boldsymbol{\varphi}_j$ $\forall j = 1, 2, \ldots, n = 20$ that is associated to the temperature profile. That is,

$$\mathbf{\Phi}_{\mathrm{T}} = \left[\tilde{\boldsymbol{\varphi}}_1, \tilde{\boldsymbol{\varphi}}_2, \ldots, \tilde{\boldsymbol{\varphi}}_j, \ldots, \tilde{\boldsymbol{\varphi}}_{20}\right],$$

where $\tilde{\boldsymbol{\varphi}}_j \in \mathbb{R}^{300}$ for $j = 1, 2, \ldots, n = 20$.

In this thesis we use the term "smoothness" to indicate that a function does not change abruptly and/or that it does not oscillate too much (low frequency content). By taking advantage of the smoothness of the most relevant columns of $\mathbf{\Phi}_{\mathrm{T}}$, we can find polynomial approximations $P_j(z)$ of the vectors $\tilde{\boldsymbol{\varphi}}_j$ $\forall j = 1, 2, \ldots, 20$ by means of a least squares regression. These approximations would satisfy:

$$P_j(z_i) \approx \tilde{\boldsymbol{\varphi}}_{ij} \quad \forall j = 1, 2, \ldots, n = 20, \tag{4.5a}$$

$$P_{\max}(z_i) \approx \bar{\mathbf{T}}_i^{\Delta_{\max}} \tag{4.5b}$$

$$\text{for } i = 1, 2, \ldots, N = 300,$$

where $\tilde{\boldsymbol{\varphi}}_{ij}$ is the ith element of $\tilde{\boldsymbol{\varphi}}_j$ associated to the ith grid point, $\bar{\mathbf{T}}_i^{\Delta_{\max}}$ is the ith element of $\bar{\mathbf{T}}^{\Delta_{\max}}$ associated to the ith grid point, $P_j(z)$ and $P_{\max}(z)$

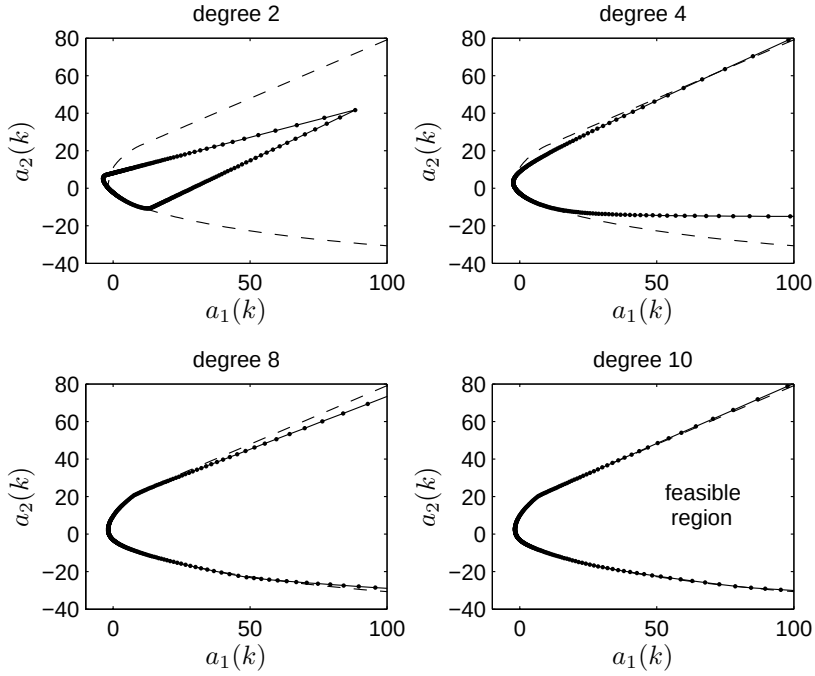


Figure 4.1: Feasible regions delimited by the temperature constraints of a 2nd order POD model. Dashed line - Original temperature constraints. Solid line - Polynomial approximations of different degree given by (4.6). Solid line with dots - Polynomial approximations of different degree given by (4.9).

are univariate real polynomials of degree d that approximate the vectors $\tilde{\boldsymbol{\varphi}}_j$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$ respectively, z is the spatial coordinate and $z_i = i\Delta z$ is the value of the spatial coordinate at the ith grid point.

By using Equations (4.5a) and (4.5b) we can approximate Equation (4.4) by

$$\hat{\boldsymbol{\Phi}}_{\mathrm{T}}\mathbf{a}(k) \leq \hat{\mathbf{T}}^{\Delta_{\max}} \tag{4.6}$$

with

$$\hat{\boldsymbol{\Phi}}_{\mathrm{T}} = \begin{bmatrix} P_1(z_1) & P_2(z_1) & \cdots & P_n(z_1) \\ P_1(z_2) & P_2(z_2) & \cdots & P_n(z_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_1(z_N) & P_2(z_N) & \cdots & P_n(z_N) \end{bmatrix}$$

$$\hat{\mathbf{T}}^{\Delta_{\max}} = \begin{bmatrix} P_{\max}(z_1) \\ P_{\max}(z_2) \\ \vdots \\ P_{\max}(z_N) \end{bmatrix}$$

where $\hat{\mathbf{\Phi}}_{\mathrm{T}}$ and $\hat{\mathbf{T}}^{\Delta_{\max}}$ are the approximations of $\mathbf{\Phi}_{\mathrm{T}}$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$ respectively, $N = 300$ is the number of sections into which the reactor is divided (number of grid points) and $n = 20$ is the number of the selected POD basis vectors.

Figure 4.1 shows the feasible regions (in solid line) delimited by Equation (4.6) for a 2nd order POD model when polynomials of different degree are used. From this Figure, it is clear that polynomials of degree 10 are accurate enough for approximating $\tilde{\boldsymbol{\varphi}}_j \ \forall j = 1, 2$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$.

Equation (4.6) imposes the temperature constraint only on the grid points of the interval $[0, 1]$. However, we can impose the condition (4.6) on every point of the interval $[0, 1]$, giving

$$\sum_{j=1}^{n} a_j(k) P_j(z) \leq P_{\max}(z), \quad \forall z \in [0, 1]$$

which can be rewritten by defining

$$P^{(k)}(z) = P_{\max}(z) - \sum_{j=1}^{n} a_j(k) P_j(z)$$

as follows:

$$P^{(k)}(z) \geq 0, \quad \forall z \in [0, 1]. \tag{4.7}$$

The resulting polynomial $P^{(k)}(z)$ of degree d has to be nonnegative, at least in the interval $z \in [0, 1]$.

Even though we have now seemingly complicated the problem by replacing many by infinitely many inequalities, this new formulation can efficiently be handled by positive polynomials techniques.

4.3.3 Semidefinite representability of positive polynomials on an interval

As it was stated in Section 4.3.1, a sufficient condition for a multivariate real polynomial to be nonnegative everywhere is whether it can be written

as a sum of squared polynomials. We denoted this property, as common, by the acronym SOS, for Sum Of Squares. In general, SOS is not equivalent to the nonnegativity of a polynomial. Nevertheless, as a direct consequence of the Fundamental Theorem of Algebra, univariate real polynomials are nonnegative everywhere if and only if they are SOS. In Section 4.3.1 we showed through Proposition 4.1 that the SOS representability of a polynomial can be expressed as a semidefinite feasibility problem [111, 117]. An adaptation of this proposition to the specific case of univariate real polynomials is given as follows.

Proposition 4.2 (see [111])**.** *A univariate polynomial $P(z)$ of degree $2d$ is SOS if and only if there exists a $(d+1)\times(d+1)$ positive semidefinite matrix* $\mathbf{W}$ *such that*

$$P(z) = \mathbf{f}(z)^T \mathbf{W} \mathbf{f}(z), \tag{4.8}$$

where $\mathbf{f}(z) = \left[1, z, z^2, \ldots, z^d\right]^T$.

As the SOS representation of a polynomial is generically not unique, the matrix $\mathbf{W}$ can not be uniquely defined.

It is possible to relate the positivity of a real univariate polynomial on a compact interval $[a, b]$ to the positivity of some other polynomial on the whole real line by the following transformation.

Proposition 4.3 (see Section 4.2, Example 21.b in [20])**.** *A real univariate polynomial p of degree d is nonnegative on the compact interval $[a, b]$ if and only if*

$$(1+z^2)^d p\left(a + \frac{(b-a)z^2}{1+z^2}\right) \geq 0, \quad \forall z \in \mathbb{R}.$$

The proof relies on the observation that the conditions:

- the rational function $g(z) = g_n(z)/g_d(z) = a + \left((b-a)z^2/(1+z^2)\right)$ has $[a, b]$ as image,
- the denominator $g_d(z)$ is positive on $z \in \mathbb{R}$,
- and $p(z) \geq 0$ on $z \in [a, b]$

are equivalent to $p(g(z)) \geq 0$ on $z \in \mathbb{R}$.

For every $1 \leq k \leq N_{\mathrm{p}}$, the condition (4.7):

$$P^{(k)}(z) = P_{\max}(z) - \sum_{j=1}^{n} a_j(k) P_j(z) \geq 0, \quad \forall z \in [0,1]$$

can be converted into:

$$\tilde{P}^{(k)}(z) = (1+z^2)^d P^{(k)}\left(\frac{z^2}{1+z^2}\right) \geq 0, \quad \forall z \in \mathbb{R},$$

and, denoting by $\mathbb{S}_+^{d+1}$ the set of $(d+1) \times (d+1)$ positive semidefinite matrices, into:

$$\begin{aligned} &\text{find } \mathbf{W}^{(k)} \in \mathbb{S}_+^{d+1} \\ &\text{such that } \tilde{P}^{(k)}(z) = \mathbf{f}(z)^T \mathbf{W}^{(k)} \mathbf{f}(z). \end{aligned} \tag{4.9}$$

Observe that the coefficients of $P^{(k)}(z)$, and thus the coefficients of $\tilde{P}^{(k)}(z)$ depend linearly on $a_j(k)$ $\forall j = 1, ..., n$. Therefore, the coefficients of $\mathbf{W}^{(k)}$ are themselves linear functions of $a_j(k)$. Henceforth, the MPC optimization problem with the polynomial approximation of the temperature constraints can be written as a Semidefinite Program (SDP). SDP are a subclass of *self-scaled* optimization problems (see [112]), that can be solved efficiently by Interior-Point Methods, such as the one implemented in the Matlab toolbox Sedumi [140].

Figure 4.1 depicts the feasible regions (solid line with dots) delimited by Equation (4.9) for a 2nd order POD model when polynomials of different degree are used. Notice how this approximation overlaps completely the approximation given by (4.6) for all the cases. It means that the error in the approximation given by (4.9) are mainly due to the errors of approximating the columns of $\mathbf{\Phi}_{\mathrm{T}}$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$ by polynomials.

For degree 10, the feasible region induced by the polynomial approximation (4.9) and by the original temperature constraints (4.4) are almost indistinguishable.

The new MPC controller based on polynomial approximations of the temperature constraints will be referred to as MPC-SDP. Its formulation is given as follows:

$$\min_{\mathbf{a}_{N_{\mathrm{p}}}, \Delta\mathbf{u}_{N_{\mathrm{c}}}, \mathbf{d}_{N_{\mathrm{p}}}, \xi} \sum_{i=1}^{N_{\mathrm{p}}} \|\mathbf{a}_{\mathrm{ref}}(k+i) - \mathbf{a}(k+i)\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_{\mathrm{c}}-1} \|\Delta\mathbf{u}(k+i)\|_{\mathbf{R}}^2 + \tag{4.10a}$$

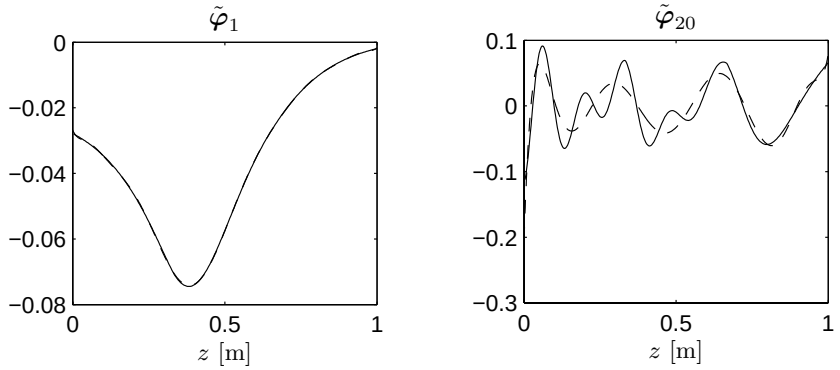


Figure 4.2: Polynomial approximations of the vectors $\tilde{\boldsymbol{\varphi}}_1$ and $\tilde{\boldsymbol{\varphi}}_{20}$. Solid line - vector. Dashed line - Approximation with a polynomial of degree 12.

$$+P_{\mathrm{Q}}\xi^2 + P_{\mathrm{L}}\xi$$

subject to

$$\begin{aligned}
&\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \tilde{\mathbf{F}}\mathbf{d}(k+i), && i = 0,\ldots,N_{\mathrm{p}}-1,\\
&\mathbf{d}(k+i+1) = \mathbf{d}(k+i), && i = 0,\ldots,N_{\mathrm{p}}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0,\ldots,N_{\mathrm{c}}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_{\mathrm{c}},\ldots,N_{\mathrm{p}}-1,\\
&\mathbf{u}_{\min} \le \mathbf{u}(k+i) \le \mathbf{u}_{\max}, && i = 0,\ldots,N_{\mathrm{c}}-1,\\
&\mathbf{W}^{(k+i)} \succeq 0, && i = 1,\ldots,N_{\mathrm{p}} \quad (4.10\mathrm{b})\\
&\tilde{P}^{(k+i)}(z) = \mathbf{f}(z)^T\mathbf{W}^{(k+i)}\mathbf{f}(z), && i = 1,\ldots,N_{\mathrm{p}} \quad (4.10\mathrm{c})\\
&\xi \ge 0,
\end{aligned}$$

with

$$\begin{aligned}
\tilde{P}^{(k+i)}(z) &= (1+z^2)^d P^{(k+i)}\left(\frac{z^2}{1+z^2}\right)\\
P^{(k+i)}(z) &= P_{\max}(z) + \eta(i)\xi - \sum_{j=1}^{n} a_j(k+i)P_j(z)\\
\mathbf{a}_{N_{\mathrm{p}}} &= [\mathbf{a}(k+1);\mathbf{a}(k+2);\ldots;\mathbf{a}(k+N_{\mathrm{p}})]\\
\mathbf{d}_{N_{\mathrm{p}}} &= [\mathbf{d}(k+1);\mathbf{d}(k+2);\ldots;\mathbf{d}(k+N_{\mathrm{p}})]\\
\Delta\mathbf{u}_{N_{\mathrm{c}}} &= [\Delta\mathbf{u}(k);\Delta\mathbf{u}(k+1);\ldots;\Delta\mathbf{u}(k+N_{\mathrm{c}}-1)]\,.
\end{aligned}$$

Here ξ is the slack variable (a scalar quantity) and $\eta(i) = 1/r_{\mathrm{c}}^{i-1}$, is a time-dependent weight ($r_{\mathrm{c}} > 1$). As it was explained in Section 4.2, the slack variable ξ and the time-dependent weight allow the MPC to deal with possible infeasibilities. The only difference of this formulation and the one of the MPC-QP controller is that (4.1b) has been replaced by (4.10b) and (4.10c). Observe that the semidefinite representation of the temperature constraint still yields linear matrix inequalities, which fall into the scope of interior-point methods for self-scaled programming.

This new MPC has the same tuning parameters as the MPC presented in Section 4.2, and it uses the same Kalman filter (3.17) as described in Section 3.4.1. We have set the degree of the polynomials $P_{\max}(z)$ and $P_1(z), P_2(z), \ldots, P_{20}(z)$ to $d = 12$. With this selection, the first seven vectors $\tilde{\boldsymbol{\varphi}}_1, \ldots, \tilde{\boldsymbol{\varphi}}_7$ are approximated very well. On the other hand, the last five vectors $\tilde{\boldsymbol{\varphi}}_{15}, \ldots, \tilde{\boldsymbol{\varphi}}_{20}$ (the less relevant ones) are approximated very poorly (see Figure 4.2). In general, the less important the POD basis function, the more oscillatory it is. If we want to improve the polynomial approximations, we would have to increase d, but this would lead to an increment in the number of constraints.

Unlike the MPC presented in Section 4.2 which deals with 24 000 temperature constraints, this MPC has only $(2d + 1) \times N_{\mathrm{p}} = 25 \times 80 = 2000$ linear equality constraints and $N_{\mathrm{p}} = 80$ Linear Matrix inequalities of dimension 13×13 for dealing with the temperature constraint of the reactor. Hence, a large reduction in the number of temperature constraints has been achieved by means of the polynomial approximations.

4.3.4 Simulation results

In order to perform the closed-loop simulations of the control systems proposed in this chapter, the same simulation environment described in Section 3.5 has been used.

In this section, we solve the optimization problem of the MPC controllers by means of *Sedumi*, a Matlab toolbox for optimization over symmetric cones [140]. It is important to remark that all the MPC controllers have been implemented using the condensed form of the MPC formulations presented before. It means that the cost function and constraints of the MPC-QP described by (4.1) have been expressed in terms of $\Delta\mathbf{u}(k)$ and ξ, and the formulation of the MPC controller based on the polynomial approximations

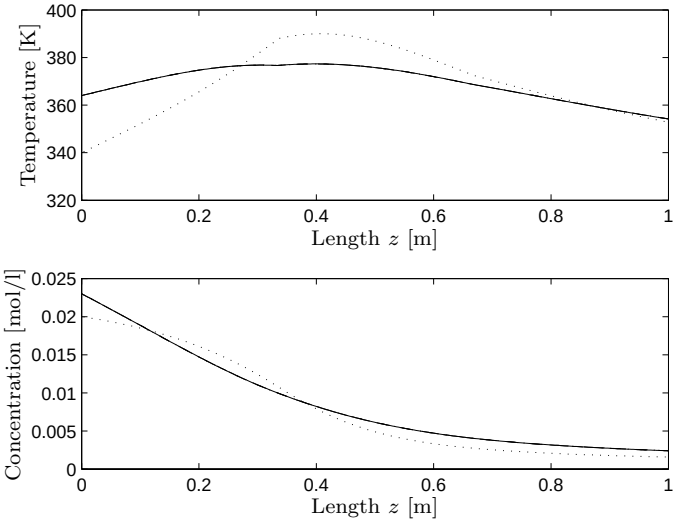


Figure 4.3: Steady-state temperature and concentration profiles of the reactor for Test 3. Dotted line - Reference. Solid line - MPC-QP. Dashed line - MPC-SDP. Dash-dotted line - MPC-NTC.

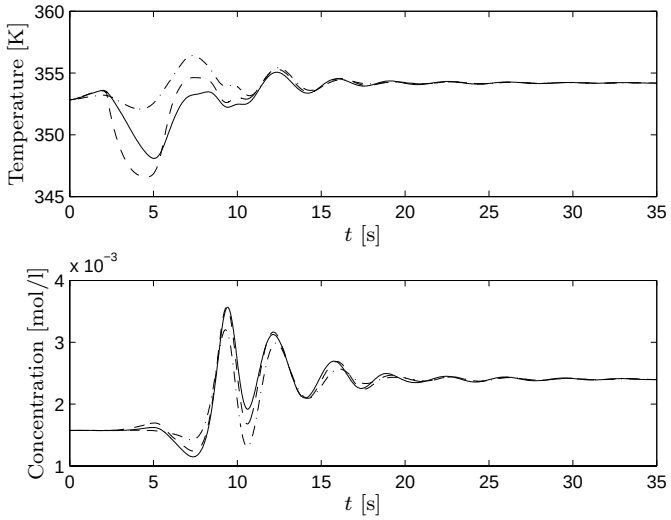


Figure 4.4: Temperature and concentration at the reactor outlet during Test 3. Solid line - MPC-QP. Dashed line - MPC-SDP. Dash-dotted line - MPC-NTC.

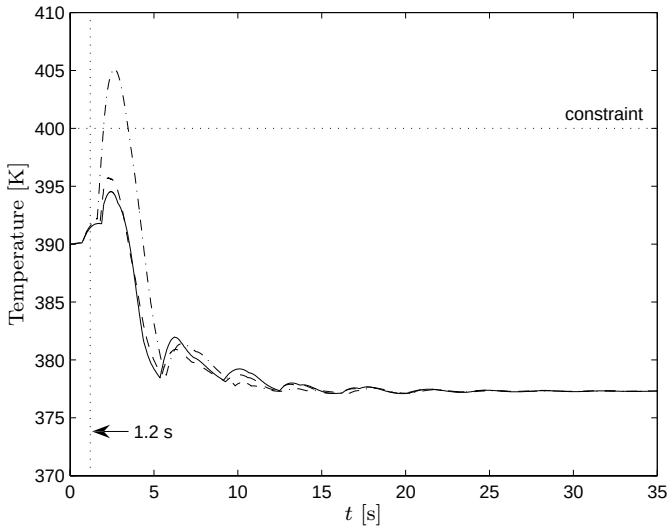


Figure 4.5: Maximal peak of the temperature profile during Test 3. Solid line - MPC-QP. Dashed line - MPC-SDP. Dash-dotted line - MPC-NTC.

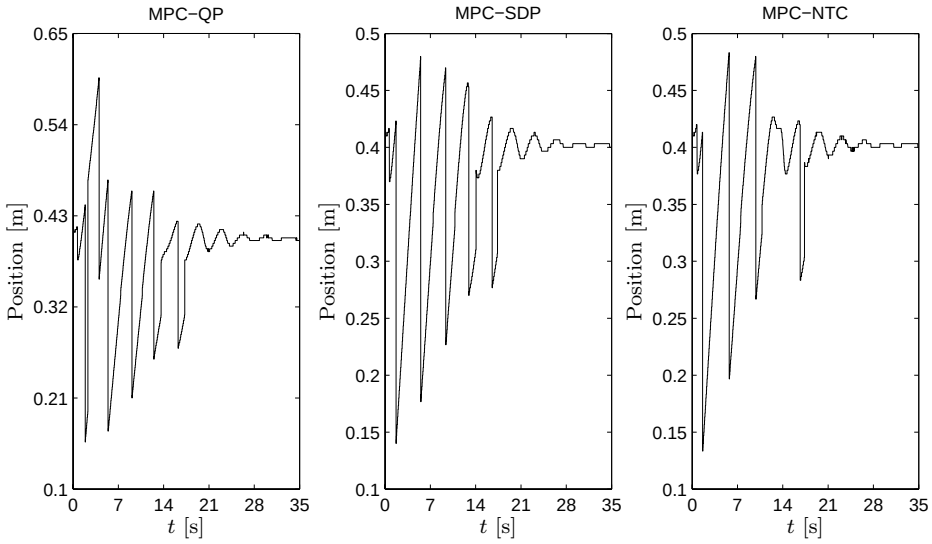


Figure 4.6: Position of the hot spot (maximal peak of the temperature profile) of the reactor during Test 3.

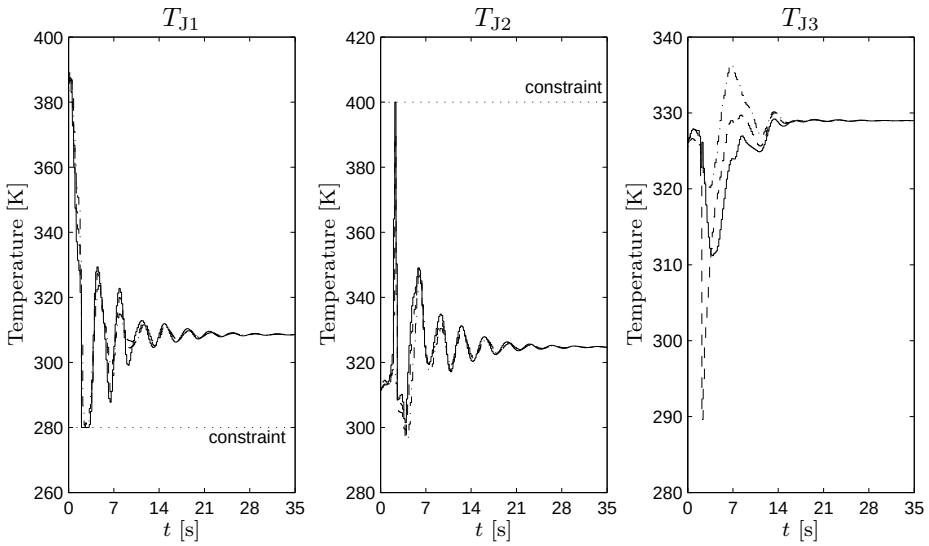


Figure 4.7: Control actions (jackets temperatures) of the MPC controllers along Test 3. Solid line - MPC-QP. Dashed line - MPC-SDP. Dash-dotted line - MPC-NTC.

(MPC-SDP) has been expressed in terms of $\Delta\mathbf{u}(k)$, ξ and the entries of the matrix $\mathbf{W}^{(k)}$.

Initially we carried out the Tests 1 and 2 proposed in Section 3.5 to compare the new MPC controllers MPC-QP and MPC-SDP and the MPC-NTC (MPC with No Temperature Constraints) controller described by (3.16) under typical disturbances. The simulations results were quite similar to the ones reported in Section 3.5 for the MPC-NTC case. Hence along Tests 1 and 2, the MPC-QP, MPC-SDP and MPC-NTC controllers kept the reactor working around the nominal operating profiles, there were no violations of the temperature constraint, the concentration in steady state at the reactor outlet was kept quite close to its nominal value, and the control actions were all the time within the allowed bounds.

The similarities in the results are due to the fact that the control systems were not operating close to the temperature constraints, and therefore during the tests, these constraints are not active in the MPC-QP and MPC-SDP controllers. In Section 3.5 two additional tests were suggested for evaluating the capability of the controllers of dealing with the temperature constraint of the reactor. From these two tests, only Test 3 pushes the reactor to

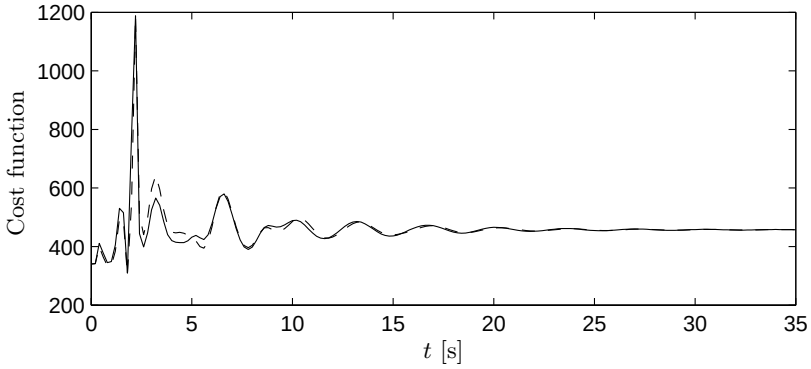


Figure 4.8: Cost function values during Test 3. Solid line - MPC-QP. Dashed line - MPC-SDP.

operating conditions where we can have temperature constraint violations when this constraint is not taken into account by the MPC formulation. This test is as follows:

- Test 3: the temperature and concentration of the feed flow are increased by 24 K and $3 \cdot 10^{-3}$ mol/l respectively. These disturbances are large in comparison with the typical ones.

Observe that under this test, the tubular reactor operates far from the operating profiles shown in Figure 3.4, and consequently there are considerable differences between the nonlinear model of the process and the linear POD model used by the controllers. Figures 4.3, 4.4, 4.5, 4.6 and 4.7 present the simulation results of Test 3.

Notice that in Figure 4.3 for all the cases, the steady state profiles of the reactor are overlapping.

In Figure 4.5 we can observe that for the case of the MPC-NTC controller, the temperature constraint is temporarily violated during 1.46 s with a maximal peak of 405.08 K. On the other hand, the MPC-QP and MPC-SDP controllers keep the temperature profile below 400 K along the test. Figure 4.6 shows the position of the hot spot (maximal peak of the temperature profile) of the reactor during the experiment. Regarding the control actions of the control systems, they were all the time within the allowed limits as can be observed in Figure 4.7.

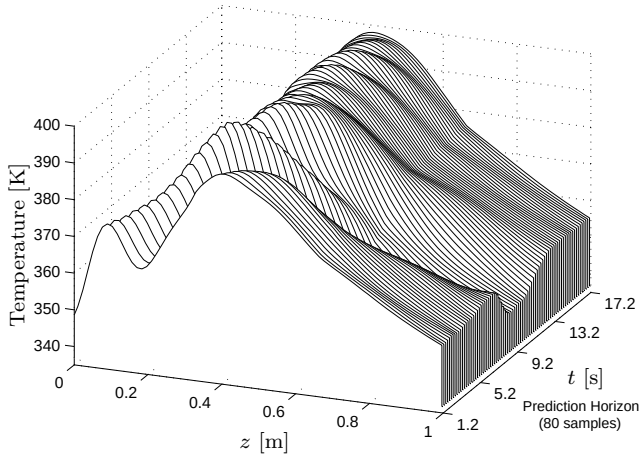


Figure 4.9: MPC-QP predictions of the temperature profile at $t = 1.2\,\mathrm{s}$ (Test 3).

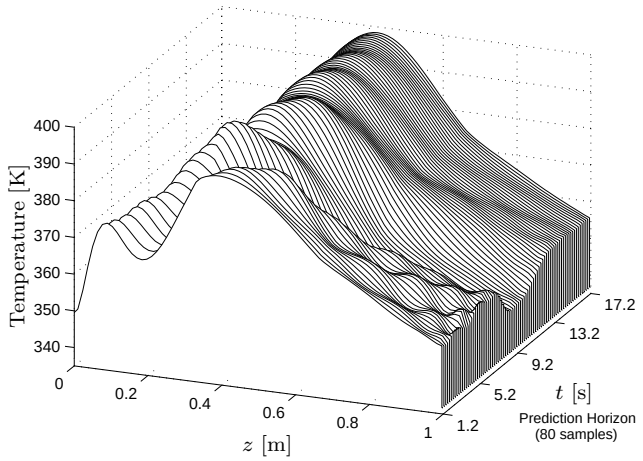


Figure 4.10: MPC-SDP predictions of the temperature profile at $t = 1.2\,\mathrm{s}$ (Test 3).

Figure 4.8 shows the cost function values obtained with the MPC-QP and MPC-SDP controllers along Test 3. We can see that the largest differences occur between $t = 1\,\mathrm{s}$ and $t = 5.6\,\mathrm{s}$ approximately. Notice that along this interval the temperature constraints are active and the differences in

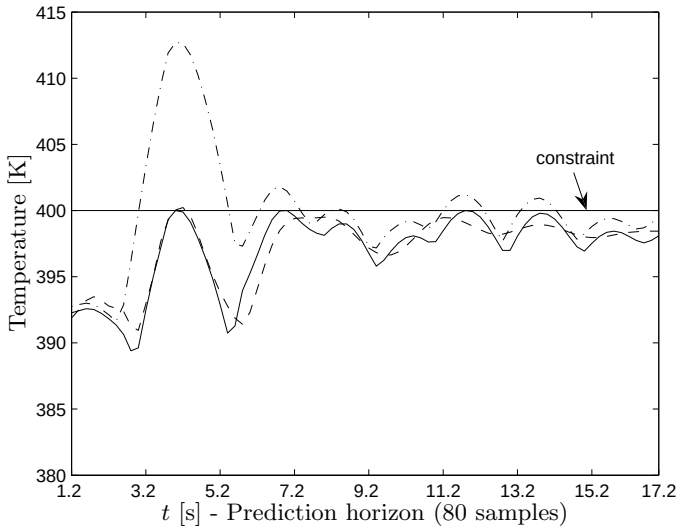


Figure 4.11: Predictions of the maximal peak of the temperature profile at $t = 1.2\,\mathrm{s}$ (Test 3). Solid line - MPC-QP. Dashed line - MPC-SDP. Dash-dotted line - MPC-NTC.

the cost function values are due to the polynomial approximation of these constraints. After this interval the difference between the cost function values becomes small and at the end practically negligible. We also can see the repercussions of the polynomial approximation in the plots of Figure 4.4, where the evolution in time of the concentration and temperature at the reactor outlet is shown for all the controllers.

The predictions of the temperature profile at $t = 1.2\,\mathrm{s}$ of the MPC-QP and MPC-SDP controllers are plotted in Figures 4.9 and 4.10 respectively. The maximal peak of these predictions can be found by projecting the plots shown in Figures 4.9 and 4.10 on the plane $z = 0$. Figure 4.11 shows the controllers' predictions of the maximal peak of the temperature profile at $t = 1.2\,\mathrm{s}$. From Figure 4.11 it is clear that the temperature constraints of MPC-QP and MPC-SDP are active. Both controllers keep the temperature below and on 400 K along the prediction horizon. The differences observed between the predictions of MPC-QP and MPC-SDP, are mainly due to the errors of approximating the temperature constraints with polynomials. If we want to reduce these discrepancies, we would have to increase the degree of the polynomials. However, this would lead to an increment in the number of

Table 4.1: Average time for solving the optimization problem

Control	$N_c = 5$ $N_p = 75$	$N_c = 10$ $N_p = 80$	$N_c = 15$ $N_p = 85$	$N_c = 20$ $N_p = 90$
MPC-QP (*Sedumi*)	6.74 s	13.01 s	21.33 s	32.89 s
MPC-QP (*Quadprog*)	0.28 s	0.31 s	0.53 s	0.76 s
MPC-SDP	3.91 s	5.77 s	8.09 s	11.2 s

Note: N_c and N_p are the control and prediction horizons in samples.

constraints and optimization variables which would increase the complexity of the optimization problem and therefore the time required to solve it.

Notice that in Test 3, the closed-loop response of the controlled system is different than the predicted one. Also observe that the steady state profiles of the reactor are far from the desired ones. None of these situations occurred during Tests 1 and 2. All of this is mainly due to considerable differences between the linear POD model used by the controllers and the observer, and the nonlinear model of the process. We have to keep in mind that during Test 3 the reactor is operating far away from the profiles (see Figure 3.4) where the nonlinear model of the reactor was linearized. It is quite clear that we have to incorporate the nonlinearities of the process into the POD model used by the controllers if we want to improve the performance of the control systems. Nevertheless this would lead to non-convex optimization problems that would require more advanced solvers. For instance, the optimization problems of the nonlinear MPC-QP and MPC-SDP controllers could be addressed by Sequential Quadratic Programming (SQP) and sequential SDP methods respectively.

Table 4.1 presents the average computation times (on a PC with an Intel Dual Core of 3 Ghz and a RAM memory of 2 GB) for solving the optimization problems of the MPC controllers during Test 3 for different control and prediction horizons. In this table we also have included the time of solving the optimization of the MPC-QP controller when a specialized QP solver like *Quadprog* is used. *Quadprog* is part of the Optimization Toolbox of Matlab [101] and it uses an active set method similar to that described in [57]. From Table 4.1 it is clear that the optimization of the MPC-SDP

Table 4.2: Number of variables, number of constraints and memory requirements when $N_c = 10$ and $N_p = 80$

Control	No. Opt. Variables	Number of Constraints				Memory (MB)
		Ineq.[a]	Equ.[b]	SOC[c]	LMI[d]	
MPC-QP (*Sedumi*)	32	24061	-	1	-	6.2
MPC-QP (*Quadprog*)	31	24061	-	-	-	6.02
MPC-SDP	7378	61	2065	2	80	0.67

[a]Inequality constraints.
[b]Equality constraints.
[c]Second Order Cone constraints.
[d]Linear Matrix Inequality constraints.

controller requires less time than the optimization of the MPC-QP controller when we use the same solver (*Sedumi*) for both cases. However if we use *Quadprog* (in general it is more efficient to solve a QP problem using a QP solver like *Quadprog* than using a more general tool like *Sedumi*) for solving the optimization of the MPC-QP controller, the time required is between 14 to 19 times shorter than the time needed to solve the optimization of the MPC-SDP controller.

Table 4.2 shows the number of optimization variables (including auxiliary variables), the number and kind of constraints and the memory requirements of the predictive controllers. It is important to remark that the MPC-SDP controller has been encoded using explicitly the primal representation in *Sedumi* whereas the MPC-QP (*Sedumi*) controller has been implemented using the dual formulation. Therefore, the values in Table 4.2 for these controllers correspond to the number of optimization variables and constraints in the primal (MPC-SDP) and dual space (MPC-QP) respectively. Notice that for the MPC-SDP case, the LMI constraints introduce a large number of variables. This is the main drawback of our approach. However in spite of this, the optimization problem for the MPC-SDP case requires less time than the case when *Sedumi* is used to solve the optimization of the MPC-QP controller. Nevertheless if we keep increasing the degree of the polynomials used to approximate the temperature constraints, we will reach a point where the time required for solving the optimization of the MPC-SDP controller would be larger than the time needed to solve the

optimization of MPC-QP with *Sedumi*.

Finally, from Table 4.2 we can see that the memory requirements (the memory needed to store the matrices that are given to the solver) of the MPC-SDP controller are significantly less than the memory demands of the MPC-QP controller (it does not matter the solver used). The MPC-SDP controller requires approximately 9 times less memory than the MPC-QP controller. Although our approach has not led to a reduction in the computational time (when the optimization of MPC-QP is performed with *Quadprog*), it certainly has led to a remarkable saving of memory.

4.4 Greedy selection algorithm

It has been observed that the coefficients of consecutive temperature constraints are quite similar. This is a consequence of the fact that the most relevant columns of $\mathbf{\Phi}_{\mathrm{T}} \in \mathbb{R}^{N\times n}$ (the part of the POD basis vectors that is associated to the temperature profile) are smooth (they do not oscillate too much). By taking into account these observations, we propose an algorithm for selecting a reduced set of constraints from the full set. The output of the algorithm would be a matrix $\mathbf{\Phi}_{\mathrm{R}} \in \mathbb{R}^{s_{\mathrm{c}}\times n}$ and a vector $\mathbf{T}_{\mathrm{R}} \in \mathbb{R}^{s_{\mathrm{c}}}$ which define the new set of temperature constraints,

$$\mathbf{\Phi}_{\mathrm{R}}\mathbf{a}(k) \leq \mathbf{T}_{\mathrm{R}}. \tag{4.11}$$

Here S_{c} is the number of selected constraints and n is the order of the POD model.

In this section we are going to adopt a different notation than the one used in the previous section in order to properly describe the algorithm. This notation is as follows:

- The ith row of the matrix $\mathbf{\Phi}_{\mathrm{T}}$ and the ith entry of the vector $\bar{\mathbf{T}}^{\Delta_{\max}}$ are denoted by $\mathbf{\Phi}_{\mathrm{T}}(i,:)$ and $\bar{\mathbf{T}}^{\Delta_{\max}}(i)$ respectively,
- The entry of the matrix $\mathbf{\Phi}_{\mathrm{T}}$ that lies in the ith row and the jth column is written as $\mathbf{\Phi}_{\mathrm{T}}(i,j)$,
- $\mathbf{\Phi}_{\mathrm{R}} = [\mathbf{\Phi}_{\mathrm{R}}; \mathbf{\Phi}_{\mathrm{T}}(i,:)]$ indicates that ith row of $\mathbf{\Phi}_{\mathrm{T}}$ is added at the bottom of the matrix $\mathbf{\Phi}_{\mathrm{R}}$, and $\mathbf{T}_{\mathrm{R}} = [\mathbf{T}_{\mathrm{R}}; \bar{\mathbf{T}}^{\Delta_{\max}}(i)]$ denotes that the ith entry of the vector $\bar{\mathbf{T}}^{\Delta_{\max}}$ is added at the bottom of the vector $\mathbf{T}_{\mathrm{R}}$.

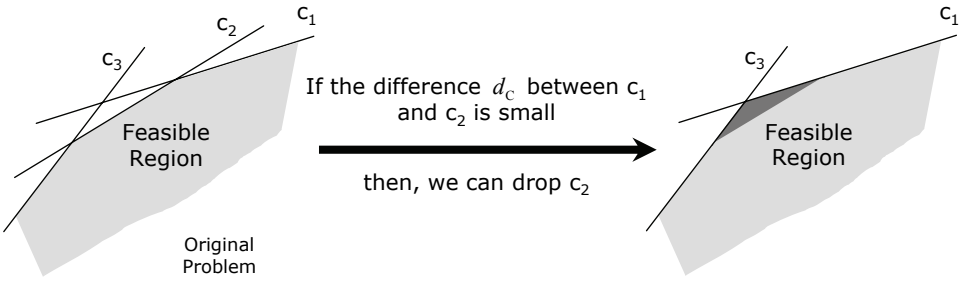


Figure 4.12: Operation 1 - Dropping a constraint of a 2nd order POD model. Here c_1, c_2 and c_3 are temperature constraints defined as follows. c_1 : $\mathbf{\Phi}_{\mathrm{T}}(1,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(1,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(1)$, c_2 : $\mathbf{\Phi}_{\mathrm{T}}(2,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(2,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(2)$, and c_3 : $\mathbf{\Phi}_{\mathrm{T}}(3,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(3,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(3)$.

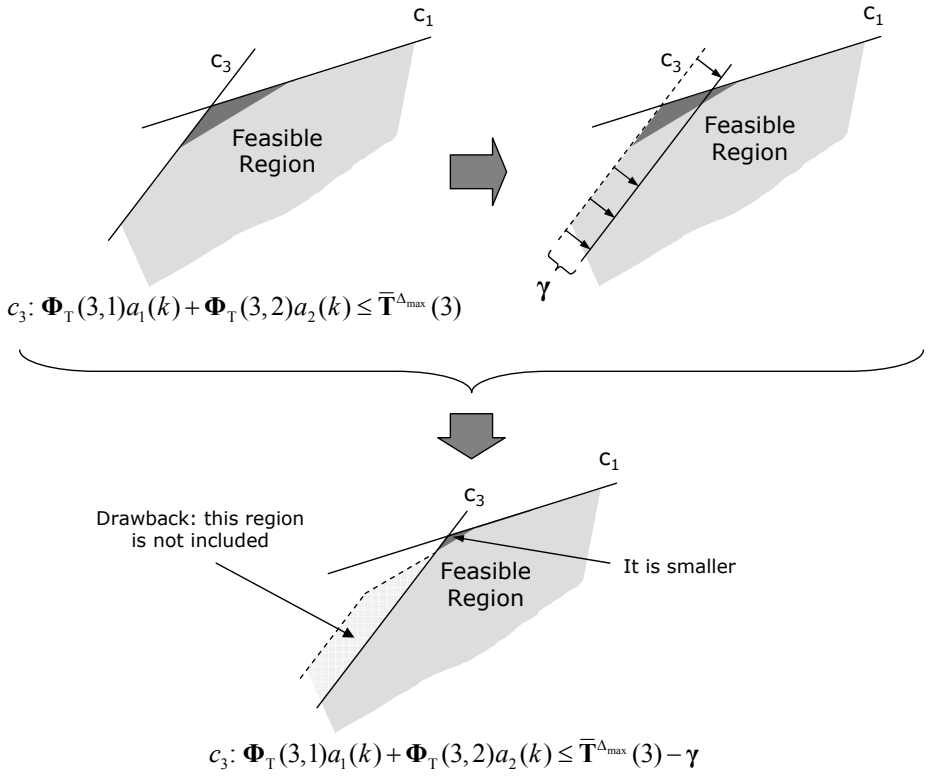


Figure 4.13: Operation 2 - Reducing the area that does not belong to the original feasible region after dropping a constraint.

The main ideas of the proposed algorithm can be explained as follows. Let us suppose that a 2nd order POD model has only 3 temperature constraints,

$$\begin{aligned}
\mathrm{c}_1 &: \mathbf{\Phi}_{\mathrm{T}}(1,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(1,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(1) \\
\mathrm{c}_2 &: \mathbf{\Phi}_{\mathrm{T}}(2,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(2,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(2) \\
\mathrm{c}_3 &: \mathbf{\Phi}_{\mathrm{T}}(3,1)a_1(k) + \mathbf{\Phi}_{\mathrm{T}}(3,2)a_2(k) \leq \bar{\mathbf{T}}^{\Delta_{\max}}(3)
\end{aligned}$$

delimiting the feasible region shown in Figure 4.12. Furthermore, let us define the following measure of distance

$$d_{\mathrm{C}}(p,i) = \frac{1}{n+1}\left(\sum_{j=1}^{n} |\mathbf{\Phi}_{\mathrm{T}}(p,j) - \mathbf{\Phi}_{\mathrm{T}}(i,j)| + |\bar{\mathbf{T}}^{\Delta_{\max}}(p) - \bar{\mathbf{T}}^{\Delta_{\max}}(i)|\right) \tag{4.12}$$

for determining how far or how close two constraints are. Here n is the number of POD coefficients or the order of the POD model. The distance $d_{\mathrm{C}}(p,i)$ is nothing else than the mean absolute error between the coefficients of the constraints that are being compared, in this case the pth and ith constraints.

In the selection algorithm we can distinguish two operations, namely, the dropping of a constraint and the reduction of the feasible region after dropping a constraint. These operations are illustrated in Figures 4.12 and 4.13 respectively. In Figure 4.12 we compare c_1 and c_2, and then we drop c_2 given that the distance $d_{\mathrm{C}}(1,2)$ between c_1 and c_2 is smaller than a predefined value. Notice that the new feasible region delimited by the reduced set of constraints c_1 and c_3, contains a small area (dark gray area in the figure) that was not part of the feasible region of the original problem. So, in order to reduce the size of this dark gray region, we introduce the so-called shrinking parameter γ to tighten nonconsecutive constraints leading to a reduction of the feasible region as it is depicted in Figure 4.13. Although we can reduce the size of this dark gray area or even get rid of it by selecting a γ large enough, the drawback of this measure is that we would remove part of the original feasible region. This is of course a conservative measure.

Our greedy selection algorithm is as follows:

1. Set $p = 1$, and select the first constraint: $\mathbf{T}_{\mathrm{R}} = \bar{\mathbf{T}}^{\Delta_{\max}}(1)$, $\mathbf{\Phi}_{\mathrm{R}} = \mathbf{\Phi}_{\mathrm{T}}(1,:)$.
2. For all $i = 2, \ldots, N-1$, perform:

(a) Calculate the difference between the pth and ith constraints using this formula:

$$d_{\mathrm{C}}(p,i) = \frac{1}{n+1}\left(\sum_{j=1}^{n} |\mathbf{\Phi}_{\mathrm{T}}(p,j) - \mathbf{\Phi}_{\mathrm{T}}(i,j)| + |\bar{\mathbf{T}}^{\Delta_{\max}}(p) - \bar{\mathbf{T}}^{\Delta_{\max}}(i)|\right)$$

(b) if $d_{\mathrm{C}}(p,i) \geq Sel$ then select the ith constraint:

- $\mathbf{\Phi}_{\mathrm{R}} = [\mathbf{\Phi}_{\mathrm{R}}; \mathbf{\Phi}_{\mathrm{T}}(i,:)]$.
- if $(i-p) > 1$ then $\mathbf{T}_{\mathrm{R}} = [\mathbf{T}_{\mathrm{R}}; \bar{\mathbf{T}}^{\Delta_{\max}}(i) - \gamma]$ else $\mathbf{T}_{\mathrm{R}} = [\mathbf{T}_{\mathrm{R}}; \bar{\mathbf{T}}^{\Delta_{\max}}(i)]$.
- Set $p = i$.

3. Select the last constraint: $\mathbf{T}_{\mathrm{R}} = [\mathbf{T}_{\mathrm{R}}; \bar{\mathbf{T}}^{\Delta_{\max}}(N)]$, $\mathbf{\Phi}_{\mathrm{R}} = [\mathbf{\Phi}_{\mathrm{R}}; \mathbf{\Phi}_{\mathrm{T}}(N,:)]$.

where N is the number of sections in which the reactor is divided and therefore the number of temperature constraints, $d_{\mathrm{C}}(p,i)$ is the mean absolute error between the coefficients of the pth and ith constraints, Sel is the minimum value of d_{C} that is required for selecting a constraint, and γ is the shrinking parameter used to tighten non consecutive constraints ($(i-p) > 1$).

Figures 4.14 and 4.15 show the feasible regions delimited by the constraints selected by the algorithm when different values of Sel and γ were used. For $Sel = 0.08$ and $\gamma = 0.01 = 0.8\,\mathrm{K}/T_{\mathrm{f}}$ ($T_{\mathrm{f}} = 80\,\mathrm{K}$), the algorithm selected 7 constraints (see Figure 4.14) from 300. These 7 constraints provide a fair approximation of the feasible region of the original problem. For $Sel = 0.022$ and $\gamma = 0.01$ (0.8 K), the algorithm chose 21 constraints (see Figure 4.15). From Figure 4.15, it is remarkable how the feasible region delimited by 300 temperature constraints can be approximated accurately by only 21 constraints. Notice also that the feasible region delimited by the reduced set of constraints is mostly inside the original feasible region. This is of course a consequence of using the shrinking parameter γ.

It is important to remark that the algorithm does not guarantee that the selected set of constraints is the optimal one, in the sense that it minimizes the difference between the feasible regions delimited by the full and the reduced set of constraints.

The formulation of the new MPC controller based on a reduced set of

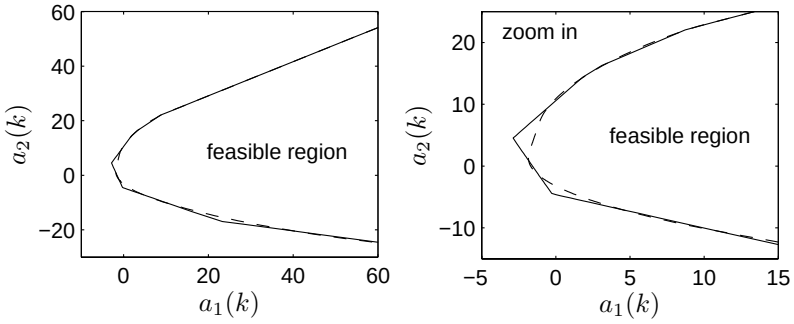


Figure 4.14: Feasible region delimited by the temperature constraints of a 2nd order POD model. Dashed line - Full set of constraints. Solid line - 7 constraints selected by the algorithm when $Sel = 0.08$ and $\gamma = 0.01$ (0.8 K).

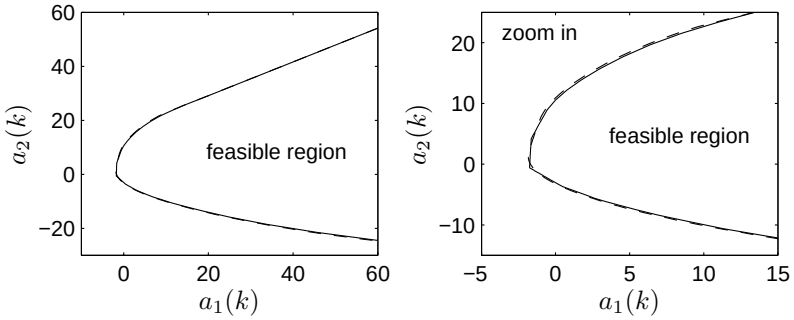


Figure 4.15: Feasible region delimited by the temperature constraints of a 2nd order POD model. Dashed line - Full set of constraints. Solid line - 21 constraints selected by the algorithm when $Sel = 0.022$ and $\gamma = 0.01$ (0.8K).

temperature constraints is given by

$$\min_{\mathbf{a}_{N_\mathrm{p}},\Delta\mathbf{u}_{N_\mathrm{c}},\mathbf{d}_{N_\mathrm{p}},\xi} \sum_{i=1}^{N_\mathrm{p}} \|\mathbf{a}_{\mathrm{ref}}(k+i) - \mathbf{a}(k+i)\|^2_{\mathbf{Q}} + \sum_{i=0}^{N_\mathrm{c}-1} \|\Delta\mathbf{u}(k+i)\|^2_{\mathbf{R}} + \quad (4.13a)$$

$$+P_\mathrm{Q}\xi^2 + P_\mathrm{L}\xi$$

subject to

$$\begin{aligned}
&\mathbf{a}(k+i+1) = \tilde{\mathbf{A}}\mathbf{a}(k+i) + \tilde{\mathbf{B}}\mathbf{u}(k+i) + \tilde{\mathbf{F}}\mathbf{d}(k+i), \quad i = 0,\ldots,N_\mathrm{p}-1,\\
&\mathbf{d}(k+i+1) = \mathbf{d}(k+i), \qquad\qquad\qquad\qquad\qquad\qquad\quad i = 0,\ldots,N_\mathrm{p}-1,
\end{aligned}$$

$$
\begin{aligned}
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i), && i = 0, \ldots, N_\mathrm{c}-1,\\
&\mathbf{u}(k+i) = \mathbf{u}(k+i-1), && i = N_\mathrm{c}, \ldots, N_\mathrm{p}-1,\\
&\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max}, && i = 0, \ldots, N_\mathrm{c}-1,\\
&\mathbf{\Phi}_\mathrm{R}\mathbf{a}(k+i) \leq \bar{\mathbf{T}}_\mathrm{R} + \mathbf{1}\cdot\eta(i)\xi, && i = 1, \ldots, N_\mathrm{p}, \quad (4.13b)\\
&\xi \geq 0,
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{a}_{N_\mathrm{p}} &= [\mathbf{a}(k+1); \mathbf{a}(k+2); \ldots; \mathbf{a}(k+N_\mathrm{p})]\\
\mathbf{d}_{N_\mathrm{p}} &= [\mathbf{d}(k+1); \mathbf{d}(k+2); \ldots; \mathbf{d}(k+N_\mathrm{p})]\\
\Delta\mathbf{u}_{N_\mathrm{c}} &= [\Delta\mathbf{u}(k); \Delta\mathbf{u}(k+1); \ldots; \Delta\mathbf{u}(k+N_\mathrm{c}-1)]\\
\eta(i) &= 1/r_\mathrm{c}^{i-1}, \quad r_\mathrm{c} > 1.
\end{aligned}
$$

Observe that the only difference of this formulation and the one of the MPC-QP controller described by (4.1), is the definition of the temperature constraints, where $\mathbf{\Phi}_\mathrm{T}$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$ have been substituted by $\mathbf{\Phi}_\mathrm{R}$ and $\mathbf{T}_\mathrm{R}$ in (4.1b), and the vector of 1's has been properly resized. From now on, this new MPC with a reduced set of temperature constraints will be referred to as MPC-QP-RS.

The MPC-QP-RS controller has the same tuning parameters as the MPC-QP controller presented in Section 4.2, and it uses the same Kalman filter (3.17) as described in Section 3.4.1. For this controller we have set $Sel = 0.03$ and $\gamma = 0.00625$ (0.5 K) in the algorithm, and it has selected $S_\mathrm{c} = 20$ constraints.

Unlike the MPC-QP which deals with 24 000 temperature constraints, this MPC has only $S_\mathrm{c} \times N_\mathrm{p} = 20 \times 80 = 1600$ constraints. Hence, a large reduction in the number of temperature constraints has been achieved thanks to the greedy selection algorithm proposed in this section. As it is presented in Table 4.3, this reduction leads to a big saving of memory, since the reduced set of constraints (0.42 MB) require 14.33 times less memory than the complete set (6.02 MB).

Finally, observe that the selection algorithm proposed in this section, allows us to find a suitable set of points where we can impose the temperature constraint in the MPC-PV controller (its formulation is in terms of physical variables) described by (3.18) when $\gamma = 0$. Keep in mind that each selected row of $\mathbf{\Phi}_\mathrm{T}$ and $\bar{\mathbf{T}}^{\Delta_{\max}}$ is associated to a specific point of the spatial domain of the reactor. Now, if we want to include the effect of the shrinking parameter

($\gamma > 0$) in the formulation of MPC-PV and reduce the number of slack variables, the following modifications should be done:

- In the cost function (3.18a) the term $\boldsymbol{\xi}^T \mathbf{P}_{\mathrm{Q}} \boldsymbol{\xi} + \mathbf{P}_{\mathrm{L}}^T \boldsymbol{\xi}$ should be replaced by $P_{\mathrm{Q}} \xi^2 + P_L \xi$, where P_{Q} and P_{L} are weighting factors ($P_{\mathrm{Q}} > 0, P_{\mathrm{L}} > 0$) and ξ is the slack variable (scalar quantity) that accounts for the maximum violation of the temperature constraint along the reactor and the prediction horizon.
- The set of inequality temperature constraints (3.18b) should be substituted by $\boldsymbol{\Phi}_{\mathrm{R}} \mathbf{a}(k+i) \leq \bar{\mathbf{T}}_{\mathrm{R}} + \mathbf{1} \cdot \eta(i) \xi, \quad \forall i = 1, \ldots, N_{\mathrm{p}}$.
- The inequality (3.18c) should be replaced by $\xi \geq 0$.

4.4.1 Simulation results

For the same reasons discussed in Section 4.3.4, the behavior of MPC-QP-RS was quite similar to those of the MPC-NTC (see Section 3.5) and MPC-QP controllers during Tests 1 and 2. Similarly, we use Test 3 to assess the ability of the MPC-QP-RS controller of handling the temperature constraint of the reactor. In this section the optimization problem of the MPC controllers is solved by means of *Quadprog* and they have been implemented using the condensed form of the MPC formulations presented previously. Figures 4.16, 4.17, 4.18, 4.19 and 4.20 show the simulation results during this test.

As it was the case for the MPC-SDP controller in Section 4.3.4, the steady state profiles of the reactor when the MPC-QP-RS is used, overlap with the profiles obtained with the MPC-QP and MPC-NTC controllers. This can be observed in Figure 4.16. In Figure 4.17 we can see the evolution in time of the temperature and concentration at the reactor outlet. Notice that the differences between the MPC-QP-RS and MPC-QP controllers are much smaller than the ones observed between MPC-SDP and MPC-QP (see Figure 4.4). From here it is clear that the full set of temperature constraints of MPC-QP is better approximated in the MPC-QP-RS case than in the MPC-SDP case.

In Figure 4.18 we can see the plot of the maximal peak of the temperature profile (hot spot), and in Figure 4.19 the plot of its position along Test 3. As it was pointed out in Section 4.3.4, for the case of the MPC-NTC controller the temperature constraint is temporary violated. This is not the case

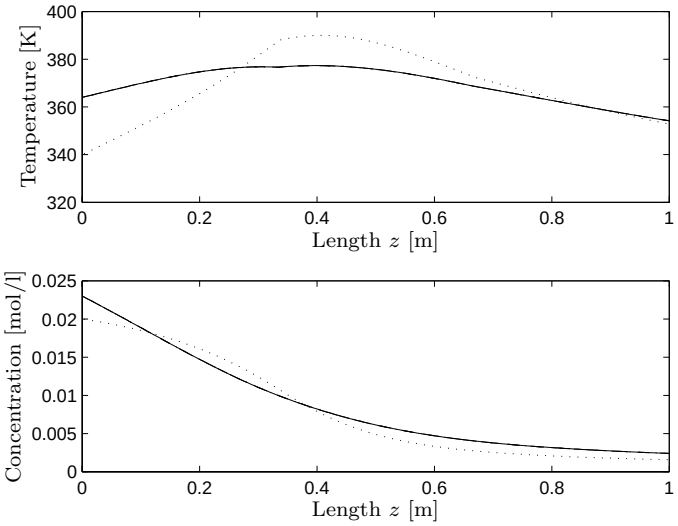


Figure 4.16: Steady-state temperature and concentration profiles of the reactor for Test 3. Dotted line - Nominal profile (reference). Solid line - MPC-QP. Dashed line - MPC-QP-RS. Dash-dotted line - MPC-NTC.

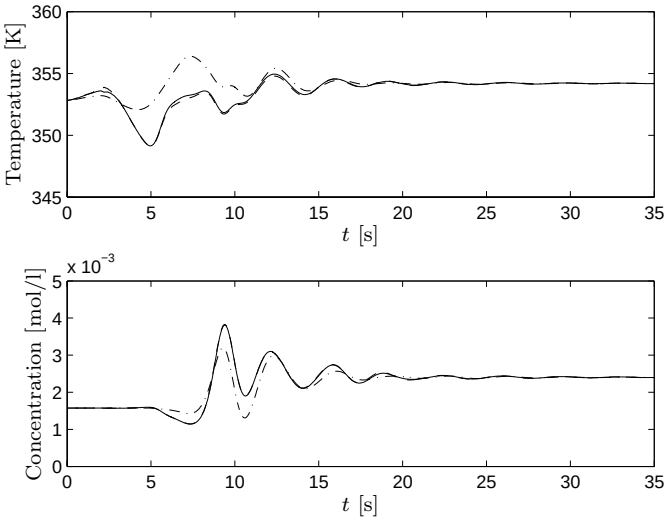


Figure 4.17: Temperature and concentration at the reactor outlet during Test 3. Solid line - MPC-QP. Dashed line - MPC-QP-RS. Dash-dotted line - MPC-NTC.

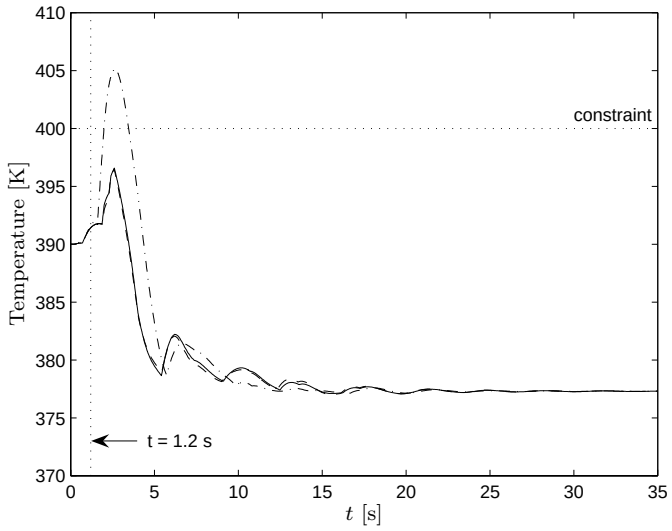


Figure 4.18: Maximal peak of the temperature profile during Test 3. Solid line - MPC-QP. Dashed line - MPC-QP-RS. Dash-dotted line - MPC-NTC.

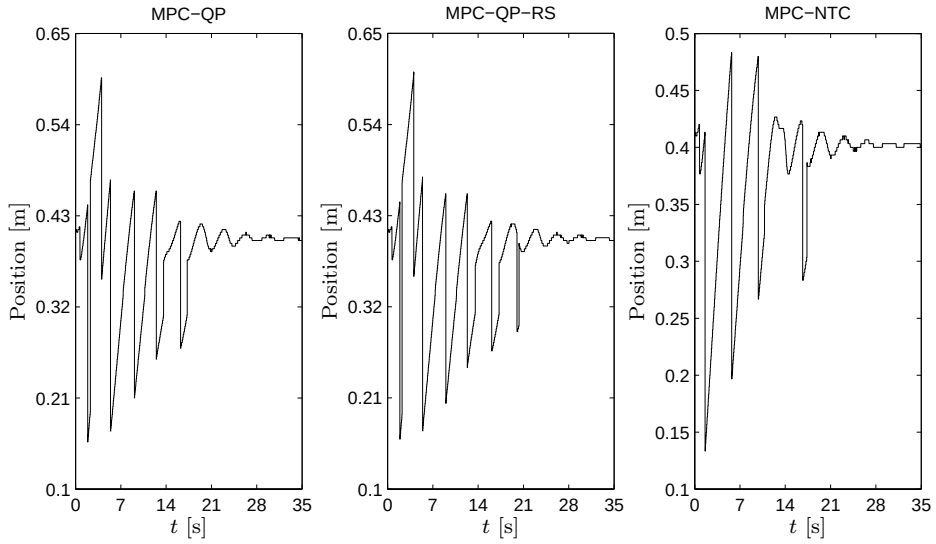


Figure 4.19: Position of the hot spot (maximal peak of the temperature profile) of the reactor during Test 3.

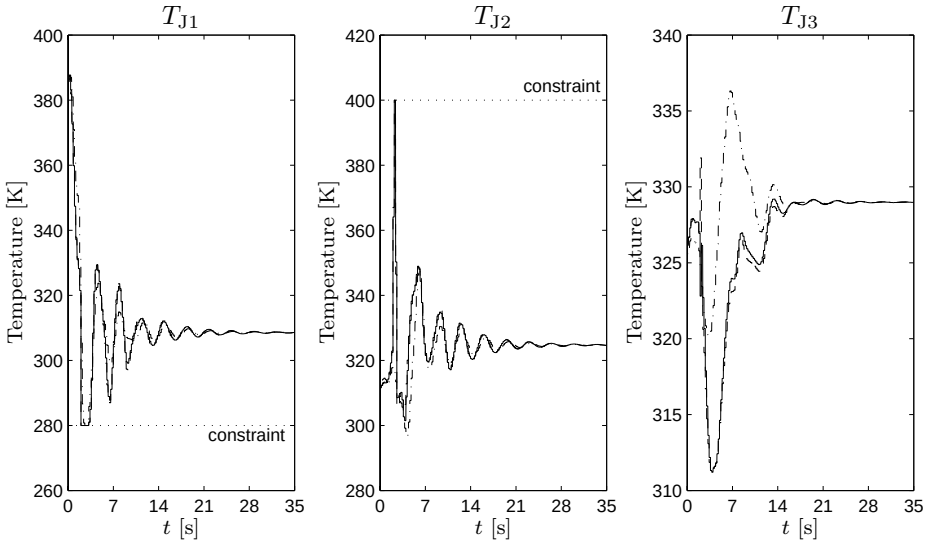


Figure 4.20: Control actions (jackets temperatures) of the MPC controllers along Test 3. Solid line - MPC-QP. Dashed line - MPC-QP-RS. Dash-dotted line - MPC-NTC.

for the MPC-QP and MPC-QP-RS controller that keep the temperature profile below 400 K. Notice also, that it is hard to see any difference in their responses.

Concerning the control actions of the MPC controllers, which are displayed in Figure 4.20, they are all the time within the allowed limits.

Figure 4.21 shows the controllers' predictions of the maximal peak of the temperature profile at $t = 1.2\,\mathrm{s}$. From Figure 4.21 it is evident that the temperature constraints of MPC-QP and MPC-QP-RS are active. Both controllers keep the temperature below and on 400 K along the prediction horizon, and the difference between their predictions are practically negligible. Note, however, that the predictions of the MPC-QP-RS controller are slightly under the predictions of MPC-QP. This is the effect of the use of the shrinking parameter γ in the greedy selection algorithm.

Table 4.3 presents the number of optimization variables, the number of linear inequality constraints and the memory requirements of the predictive controllers. In addition, this table shows the average computation times (on a PC with an Intel Dual Core of 3 Ghz and a RAM memory of 2 GB) for

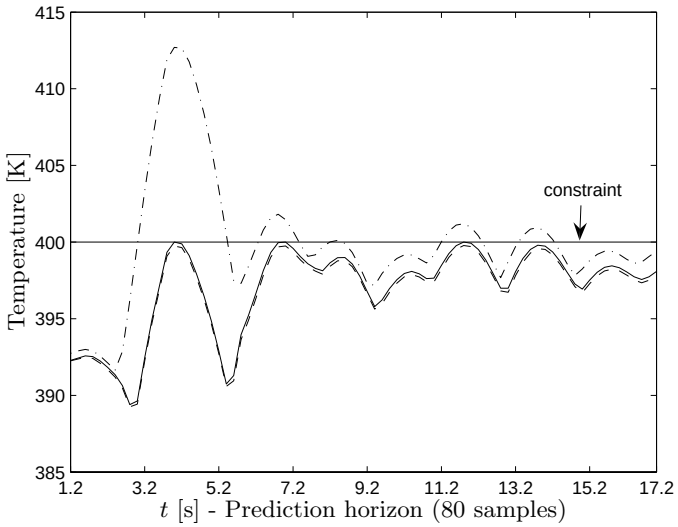


Figure 4.21: Predictions of the maximal peak of the temperature profile at $t = 1.2\,\mathrm{s}$. Solid line - MPC-QP. Dashed line - MPC-QP-RS. Dash-dotted line - MPC-NTC.

Table 4.3: Number of variables, number of constraints and average time for solving the optimization problem

Control	No. Opt. Variables	No. inequality constraints	Memory (MB)	t_{opt}
MPC-QP	31	24061	6.02	0.31 s
MPC-QP-RS	31	1661	0.42	0.023 s

t_{opt} = Average time for solving the optimization problem.
QP solver = Quadprog.

solving the optimization problems of the MPC controllers during Test 3. These average times were 0.31 s and 0.023 s for the MPC-QP and MPC-QP-RS cases respectively. The MPC-QP-RS controller requires 13.48 times less time than the MPC-QP controller for solving the optimization. To sum up, the reduction in the number of temperature constraints by means of the algorithm proposed in this section, has not only conduced to a considerable

saving of memory, but also it has led to a remarkable reduction in the computational effort.

4.5 Conclusions

First of all in this chapter we have presented an extension of the MPC-NTC controller proposed in Section 3.4, which takes into account the temperature constraint of the reactor. We have shown that this extension has to handle a very large number of temperature constraints which demands a considerable amount of memory and computational power. In this chapter we have proposed two methods to tackle this problem.

In the first method, part of the basis vectors derived with the POD technique are approximated with univariate real polynomials. Afterwards, the theory of positive polynomials is used for approximating the temperature constraints by means of Linear Matrix Inequalities and linear equality constraints. The method leads to a significant reduction in the number of constraints which conduces to a considerable saving of the memory. However the computational time needed for solving the optimization problem of the predictive controller based on the polynomial approximations, is much larger than the time required for solving the original problem. What mainly limits the computational gain of this technique is the large number of variables that are introduced by the LMI constraints. From this study it is clear that with this positive polynomial approach the resulting optimization problem is more complex than the original one. Nevertheless this approach guarantees the fulfillment of the temperature constraint at every point of the reactor.

The predictive controller based on the polynomial approximation presented a good behavior, and it was able to deal with the temperature constraints quite well.

This approach, that we only applied to linear system models so far, can in a straightforward way be generalized to the case of nonlinear MPC and would then lead to the interesting problem class of nonlinear SDP problems that can, e.g., be addressed by the sequential SDP methods proposed and investigated in [42, 50, 55].

In the second method, we exploit the fact that the coefficients of consecutive constraints are quite similar in order to select a reduced set of constraints

from the complete set. This method leads to a significant reduction in the number of constraints, which conduces to a considerable saving of memory, and a substantial reduction in the computational time required for solving the optimization of the MPC controller. Note however that unlike the positive polynomial approach, in this method we do not have any command on the temperature between the discretization points.

The predictive controller based on the reduced set of constraints presented a good behavior and it was able to deal with the temperature constraints quite well. Additionally, its behavior was practically identical to the behavior of the predictive controller based on the complete set of constraints.

Future research is necessary in order to find out which of the techniques proposed works best for which kind of applications.

Chapter 5

Performance Improvement in Model Simulation

5.1 Introduction

Several studies have reported that we can get a large model-order reduction by using POD and Galerkin projection. Nevertheless the computation saving offered by the reduced order models is small when nonlinear or Linear Time Variant (LTV) systems are considered. The reason of this limitation lies in the fact that we need the full spatial information from the original high-dimensional systems in order to evaluate the reduced-order models.

In [10–12] a method known as Missing Point Estimation (MPE) is proposed for tackling this problem. In this method the Galerkin projection is conducted only on some pre-selected state variables instead of the entire set. The remaining state variables are estimated by means of the POD basis vectors. It has been reported that this technique can save considerable computational effort.

In this chapter we present two alternative techniques for accelerating the evaluation of nonlinear POD models.

In the first method a Multi-Layer Perceptron (MLP) neural network is used to approximate the nonlinear vector function of the POD models [7]. Given that the time for evaluating the trained MLP can be very short, we can obtain a significant saving of computational time.

The second method is mainly intended for accelerating nonlinear POD models derived from input-affine high-dimensional systems with polynomial nonlinearities [6]. It turns out that by exploiting their polynomial nature, we can construct compact and efficient representations that can be evaluated much faster. Besides, the computational gain can be increased even more, if we use sequential feature selection algorithms for choosing the most relevant monomials (suboptimal solution) of these representations. Notice that this approach might also be applied to high-dimensional systems with non-polynomial nonlinearities, provided that these nonlinearities can be approximated by low degree polynomials.

Usually, the model reduction of a stable model by POD often results in a reduced order model that is stable as well. However, this is generally not guaranteed [119]. By using Lyapunov's indirect method we can obtain an eigenvalue constraint that can be used for guaranteeing the local stability of a POD model with polynomial nonlinearities. Since this constraint leads to a non-smooth optimization problem, which is difficult to solve in its original form, in this chapter we also present two manners of dealing with this constraint. In both cases, the local stability is guaranteed independently of the data used, although the quality of the reduced-order model still depends on the quality of the data.

This chapter is organized as follows. Section 5.2 presents a description of the dynamical system that will be used to explain our techniques, the nonlinear heat transfer problem in a one-dimensional bar. In Section 5.3 the derivation of a reduced order model for the bar by means of POD and Galerkin projection is discussed. Section 5.4 explains our approach for speeding up nonlinear POD models by using neural networks. In Section 5.5 we introduce our technique for accelerating the nonlinear POD model found in Section 5.3 by exploiting its polynomial nature. Section 5.6 discuss how the local stability of POD models with polynomial nonlinearities can be guaranteed. In Section 5.7 we present some validation and simulation results. Finally Section 5.8 presents some concluding remarks.

5.2 Nonlinear heat transfer in a one-dimensional bar

The system under study is the silicon bar shown in Figure 5.1. The bar has attached an actuator which provides a uniformly distributed heat flux $u(t)$ between $z = z_\mathrm{a}$ and $z = z_\mathrm{b}$. Additionally, an external heat flux $d(t)$ is applied uniformly along the bar whose ends are kept at 25°C (ambient temperature) all the time.

If only temperature variations in the z-direction are considered, the dynamics of the temperature $T(z,t)$ of the bar can be modeled by the following nonlinear PDE:

$$\rho C_\mathrm{p} \frac{\partial T(z,t)}{\partial t} = \frac{\partial}{\partial z}\left(\kappa\left(T(z,t)\right)\frac{\partial T(z,t)}{\partial z}\right) + V(z,t) \tag{5.1}$$

with the following initial and boundary conditions:

$$\begin{aligned} T(z, t=0) &= 25^\circ\mathrm{C} \\ T(z=0,t) &= T(z=L,t) = 25^\circ\mathrm{C}. \end{aligned}$$

Here ρ is the material density in $[\mathrm{kg}\cdot\mathrm{m}^{-3}]$, C_p is the heat capacity in $[\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}^{-1}]$, $\kappa(T)$ is the temperature dependent heat conductivity in $[\mathrm{J}\cdot\mathrm{s}^{-1}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}]$, t is the time in [s], z is the spatial coordinate in [m] and $V(z,t)$ is the heat source applied to the bar at position z and time t in $[\mathrm{W}\cdot\mathrm{m}^{-3}]$. $V(z,t)$ is defined as follows:

$$V(z,t) = \begin{cases} d(t) + u(t), & z_\mathrm{a} \le z \le z_\mathrm{b} \\ d(t), & \text{elsewhere.} \end{cases}$$

The relation between the temperature and the heat conductivity $\kappa(T)$ is described by a polynomial of degree 3,

$$\kappa(T) = \kappa_0 + \kappa_1 T + \kappa_2 T^2 + \kappa_3 T^3 \tag{5.2}$$

where $\kappa_0 = 36$, $\kappa_1 = -0.1116$, $\kappa_2 = 1.7298\times10^{-4}$ and $\kappa_3 = -1.78746\times10^{-7}$ are real coefficients in the appropriated units.

The length of the bar is $L = 0.1$ m and the remaining numerical values of the model parameters are: $\rho = 3970$ kg $\cdot$ m^{-3}, $C_\mathrm{p} = 766$ J $\cdot$ kg^{-1} $\cdot$ K^{-1}, $z_\mathrm{a} = 0.005$ m and $z_\mathrm{b} = 0.04$ m.

The operating ranges in $[\mathrm{W} \cdot \mathrm{m}^{-3}]$ of $d(t)$ and $u(t)$ are $-500 \cdot 10^3 \leq d(t) \leq 500 \cdot 10^3$ and $-1500 \cdot 10^3 \leq u(t) \leq 1500 \cdot 10^3$ respectively. Some of the previous numerical values were inspired on the values given in [156].

For simulation purposes, it is necessary to reduce the infinite dimensionality of (5.1) by discretizing the spatial domain. To this end, the partial derivatives with respect to space were replaced by backward (the inner spatial derivative) and forward (the outer spatial derivative) difference approximations. This is equivalent to replace the second partial derivative with respect to space by a central difference approximation in the linear version of the heat equation where κ is kept constant (see for example (2.23) and (2.25)). The discretized model of the bar is given by the following set of nonlinear ordinary differential equations:

$$\frac{dT_i}{dt} = c_1\Big(\kappa(T_{i+1})T_{i+1} - \big(\kappa(T_{i+1}) + \kappa(T_i)\big)T_i + \kappa(T_i)T_{i-1}\Big) + c_2 V_i \quad \text{(5.3a)}$$

$$\text{for } i = 1, \ldots, N-1,$$

with

$$\begin{aligned} c_1 &= \frac{1}{\rho C_\mathrm{p} \left(\Delta z\right)^2} \\ c_2 &= \frac{1}{\rho C_\mathrm{p}} \\ T_0 &= T_N = 25^\circ\mathrm{C} \end{aligned} \quad \text{(5.3b)}$$

where N is the number of sections in which the bar is divided, Δz is the length of each section, and T_i and V_i are the temperature and heat flux at the point $z_i = i\Delta z$.

If we define $\mathbf{T}(t) \in \mathbb{R}^{N-1} = [T_1(t), T_2(t), \ldots, T_{N-1}(t)]^T$ as the vector containing the temperature of the grid points from z_1 till z_{N-1} every time instant, we can write Equation (5.3) as follows:

$$\dot{\mathbf{T}}(t) = \mathbf{F}\left(\mathbf{T}(t)\right) + \mathbf{B}_1 d(t) + \mathbf{B}_2 u(t) \quad \text{(5.4)}$$

where $\mathbf{F}\left(\mathbf{T}(t)\right) : \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$ is a vector-valued or vector function which contains the nonlinear terms of the model and whose ith component function is given by

$$F_i(\mathbf{T}(t)) = c_1\Big(\kappa(T_{i+1}(t))T_{i+1}(t) - \big(\kappa(T_{i+1}(t)) + \kappa(T_i(t))\big)T_i(t) +$$

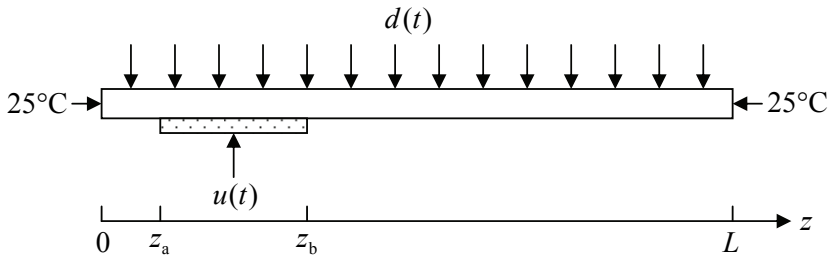


Figure 5.1: Silicon bar.

$$+ \kappa(T_i(t))T_{i-1}(t)\Big),$$

and $\mathbf{B}_1$ and $\mathbf{B}_2$ are vectors defined as: $\mathbf{B}_1 \in \mathbb{R}^{N-1} = [c_2, c_2, \ldots, c_2]^T$, $\mathbf{B}_2 \in \mathbb{R}^{N-1} = [0, \ldots, 0, c_2, \ldots, c_2, 0, \ldots, 0]^T$. The position of the nonzero elements in $\mathbf{B}_2$ corresponds to the position of the grid points that are in contact with the actuator.

The spatial domain was divided into $N = 500$ sections which means that (5.4) has $N - 1 = 499$ states. With such amount of states the design of a control system for the bar is not an easy task. In addition, the simulation of (5.4) demands a considerable amount of computational resources. In order to tackle this situation, in the next section a reduced order model (few number of equations and states) of the bar will be obtained by means of POD and Galerkin Projection.

5.3 Nonlinear POD model of the system

Given that the initial state of (5.4) does not provide information about the system dynamics, we are going to work with the temperature deviations with respect to the ambient temperature (25°C). Consequently, the vector $\mathbf{T}(t)$ is split as follows:

$$\mathbf{T}(t) = \mathbf{T}^{\Delta}(t) + \mathbf{T}^{*}$$

where $\mathbf{T}^{\Delta}(t) \in \mathbb{R}^{N-1}$ is the vector containing the deviations of the temperature profile and $\mathbf{T}^{*} \in \mathbb{R}^{N-1}$ is a constant vector which contains the initial temperature profile of the bar (ambient temperature).

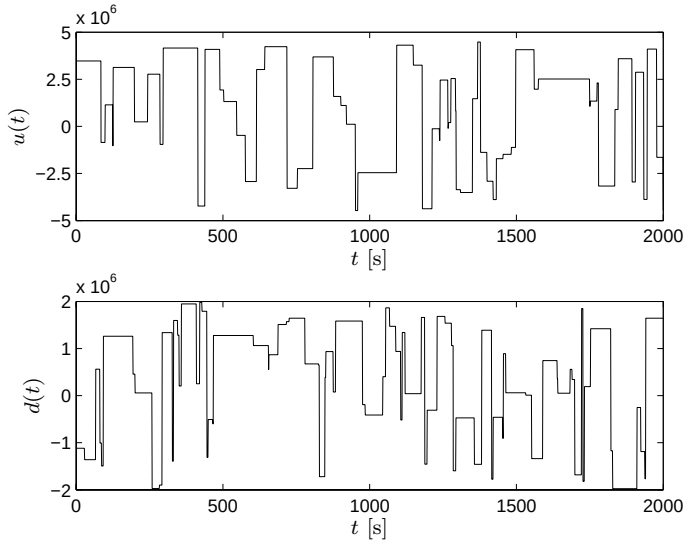


Figure 5.2: Pseudo random multilevel noise signals used in the generation of the snapshot matrix $\mathbf{T}_{\text{snap}}$. Amplitudes in $[\text{W} \cdot \text{m}^{-3}]$.

Similarly as it was done in Sections 2.4.3 and 3.3, the reduced order model of (5.4) is derived in four steps as follows:

1. **Generation of the Snapshot Matrix**. The snapshot matrix $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{499\times 2001}$ has been built by collecting the evolution of the deviations of the temperature profile when Pseudo Random Multilevel Noise Signals (PRMNS) were applied to the process inputs $u(t)$ and $d(t)$,

$$\mathbf{T}_{\text{snap}} = \left[\mathbf{T}^{\Delta}(t=0), \mathbf{T}^{\Delta}(t=\Delta t), \ldots, \mathbf{T}^{\Delta}\left(t=(N_{\text{d}}-1)\Delta t\right)\right]. \quad (5.5)$$

   Here, $N_{\text{d}} = 2001$ is the number of samples gathered using a sampling time Δt of 1 s. The excitation signals can be observed in Figure 5.2. A commutation probability of 3% was set for the signals and the amplitudes in $[\text{W}\cdot\text{m}^{-3}]$ of $d(t)$ and $u(t)$ were restricted to the intervals $[-500\cdot 10^3, 500\cdot 10^3]$ and $[-1500\cdot 10^3, 1500\cdot 10^3]$ respectively.

2. **Derivation of the POD basis vectors.** We have found the POD basis vectors $\mathbf{\Phi} \in \mathbb{R}^{499\times 499} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{499}]$ by performing the SVD (the POD basis vectors are the left singular vectors) of the

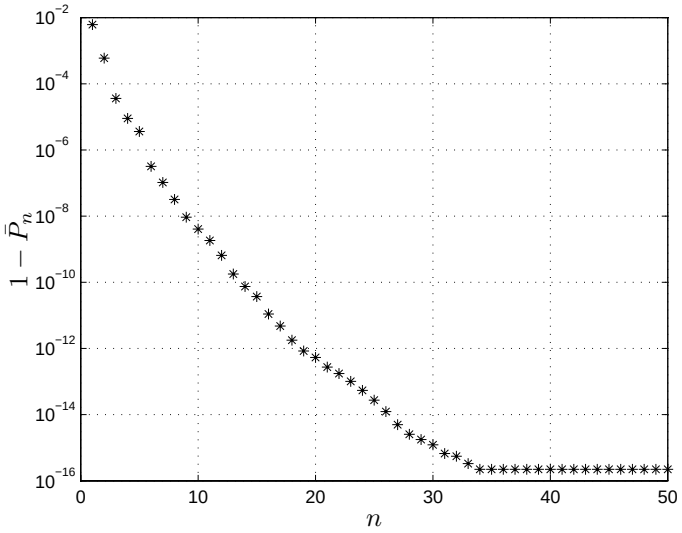


Figure 5.3: The logarithmic plot of $1 - \bar{P}_n$ which is used to determine the truncation degree of the POD basis vectors.

snapshot matrix $\mathbf{T}_{\text{snap}}$,

$$\mathbf{T}_{\text{snap}} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$$

where $\mathbf{\Phi} \in \mathbb{R}^{499\times 499}$ and $\mathbf{\Psi} \in \mathbb{R}^{2001\times 2001}$ are unitary matrices and $\mathbf{\Sigma} \in \mathbb{R}^{499\times 2001}$ is a matrix containing the singular values in its main diagonal.

3. **Selection of the most relevant POD basis vectors.** We have chosen the n most relevant POD basis vectors based on the energy criterion discussed in Section 2.2.1. Figure 5.3 shows the plot of $1-\bar{P}_n$ (see Equation (2.7)) for the first 50 basis vectors. For this problem we have selected the first $n = 6$ POD basis vectors based on their truncation degree $1-\bar{P}_n = 3.194\cdot 10^{-7}$. The selected basis vectors can be observed in Figure 5.4.

   The 6th order approximation of $\mathbf{T}^{\Delta}(t)$ is then given by

$$\mathbf{T}_n^{\Delta}(t) = \sum_{j=1}^{6} a_j(t)\boldsymbol{\varphi}_j = \mathbf{\Phi}_n\mathbf{a}(t) \tag{5.6}$$

   where $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_6]$ and $\mathbf{a}(t) = [a_1(t), a_2(t), \ldots, a_6(t)]^T$.

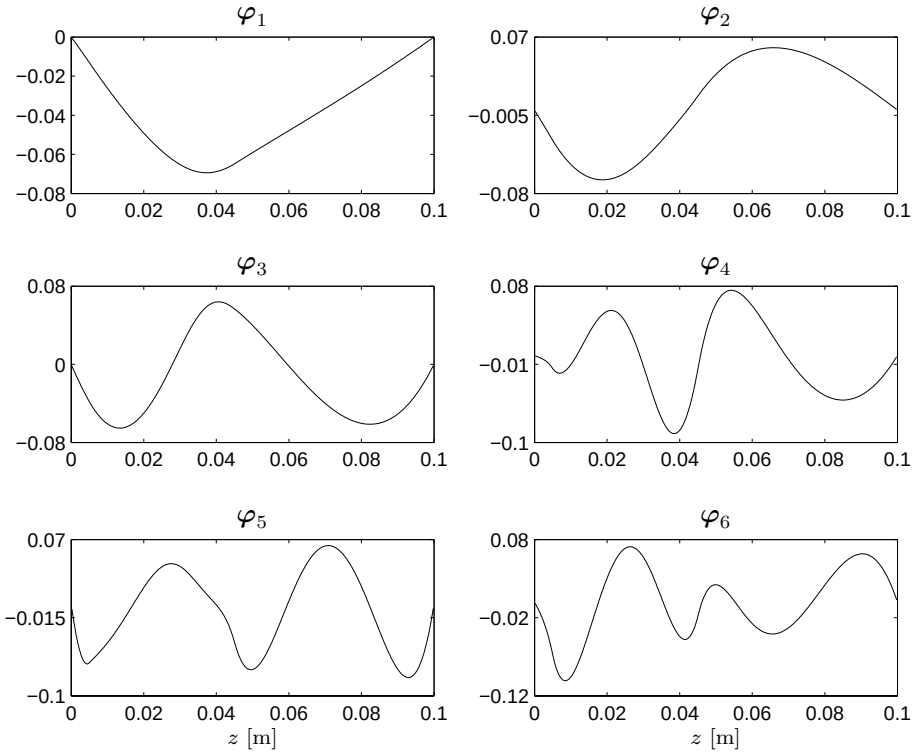


Figure 5.4: Selected POD basis vectors.

4. **Construction of the model for the first n=6 POD coefficients.** The dynamic model for the POD coefficients is derived by performing the Galerkin projection of the nonlinear model (5.4) on the space spanned by the selected basis vectors $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_6]$. Thus, if we replace $\mathbf{T}(t)$ by its nth order approximation $\mathbf{T}_n(t) = \mathbf{T}_n^{\Delta} + \mathbf{T}^* = \mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*$ in Equation (5.4), and we apply the inner product criterion to the resulting equation, we have that

$$\left\langle \mathbf{\Phi}_n\dot{\mathbf{a}}(t), \boldsymbol{\varphi}_j \right\rangle = \left\langle \mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right) + \mathbf{B}_1 d(t) + \mathbf{B}_2 u(t), \boldsymbol{\varphi}_j \right\rangle, \qquad (5.7)$$

$$\forall j = 1, 2, \ldots, n = 6.$$

By evaluating the inner product in (5.7),

$$\mathbf{\Phi}_n^T\mathbf{\Phi}_n\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T\mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right) + \mathbf{\Phi}_n^T\mathbf{B}_1 d(t) + \mathbf{\Phi}_n^T\mathbf{B}_2 u(t)$$

$$\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T\mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right) + \mathbf{\Phi}_n^T\mathbf{B}_1 d(t) + \mathbf{\Phi}_n^T\mathbf{B}_2 u(t) \qquad (5.8)$$

we get the model for the first $n = 6$ POD coefficients. Hence, the reduced order model of the bar with only 6 states is given by,

$$\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T \mathbf{F}\left(\mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{5.9}$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*$$

where $\tilde{\mathbf{B}}_1 = \mathbf{\Phi}_n^T \mathbf{B}_1$ and $\tilde{\mathbf{B}}_2 = \mathbf{\Phi}_n^T \mathbf{B}_2$. Finally if we define a new vector-valued function $\mathbf{f} : \mathbb{R}^6 \to \mathbb{R}^6$ as $\mathbf{f}\left(\mathbf{a}(t)\right) = \mathbf{\Phi}_n^T \mathbf{F}\left(\mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*\right)$, then the reduced order model of the bar can be written more compactly as follows:

$$\dot{\mathbf{a}}(t) = \mathbf{f}\left(\mathbf{a}(t)\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{5.10a}$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*. \tag{5.10b}$$

In general, it should be clear that we do not know the compact expression of $\mathbf{f}(\mathbf{a}(t))$ in (5.10a). So, in order to simulate the reduced order model, the ODE solver has to evaluate indirectly $\mathbf{f}(\mathbf{a}(t))$. Firstly, the solver has to map the state of the reduced order model $\mathbf{a}(t)$ into the original high-dimensional space by means of this linear transformation $\mathbf{T}_n^{\Delta}(t) = \mathbf{\Phi}_n \mathbf{a}(t)$. Secondly it has to evaluate the resulting high-dimensional state vector $\mathbf{T}_n^{\Delta}(t)$ in the vector function $\mathbf{F}(\mathbf{T}_n^{\Delta}(t) + \mathbf{T}^*)$ of (5.4), and finally it has to map the results of this evaluation to the low dimensional space by pre-multiplying them by $\mathbf{\Phi}_n^T$. The evaluation of $\mathbf{f}(\mathbf{a}(t))$ is done as many times as it is required by the ODE solver within each integration step. Hence the indirect evaluation of $\mathbf{f}(\mathbf{a}(t))$ is the bottleneck that limits severely the computational gain of the nonlinear POD model. In order to overcome this situation, in the next sections we present two approaches for speeding up the evaluation of nonlinear POD models like (5.10). In the first method we use feedforward neural networks and in the second one we exploit the polynomial nature of these POD models.

5.4 Acceleration of POD models by using neural networks

In order to speed up the evaluation of (5.10a) we propose to approximate the vector function $\mathbf{f} : \mathbb{R}^6 \to \mathbb{R}^6$ by using a multi-layer perceptron neural network. In this way we eliminate the necessity of evaluating the vector

function $\mathbf{F} : \mathbb{R}^{499} \rightarrow \mathbb{R}^{499}$ of the full order model and we can save a considerable amount of time.

As it is well-known, a multi-layer perceptron can learn any nonlinear input-output mapping given an adequate number of hidden neurons (each one with a nonlinear activation function) in its hidden layers [63]. In addition, the time required for calculating the MLP output can be quite short since only few matrix multiplications, vector additions and function evaluations are necessary. Due to these characteristics, an MLP is a suitable choice for approximating the vector function $\mathbf{f}$ in (5.10a).

In order to generate the input and output data required for training, validating and testing the MLP, firstly the POD model (5.10) was excited with PRMNS signals and the evolution of the state vector $\mathbf{a}(\mathrm{t})$ was collected. From the test the following data sets were constructed:

$$\begin{aligned}\mathcal{U} &= \{u(0), u(\Delta t), \ldots, u(10000\Delta t)\}\,,\\ \mathcal{D} &= \{d(0), d(\Delta t), \ldots, d(10000\Delta t)\}\,,\\ \mathcal{A} &= \{\mathbf{a}(0), \mathbf{a}(\Delta t), \ldots, \mathbf{a}(10000\Delta t)\}\,.\end{aligned}$$

In the experiment 10001 samples were gathered with a sampling time Δt equal to 1 s. The commutation probability of the PRMNS signals was set to 3% and the amplitudes in $[\mathrm{W}\cdot\mathrm{m}^{-3}]$ of $d(t)$ and $u(t)$ were restricted to the intervals $[-600\cdot 10^3, 600\cdot 10^3]$ and $[-1800\cdot 10^3, 1800\cdot 10^3]$ respectively. Notice that these intervals are 20% larger than the operating ranges defined in Section 5.2. This enlarges the range of validity of our approximation with the MLP.

If we define a vector $\mathbf{y}(t) \in \mathbb{R}^6$ as follows:

$$\mathbf{y}(t) = \dot{\mathbf{a}}(t) - \tilde{\mathbf{B}}_1 d(t) - \tilde{\mathbf{B}}_2 u(t), \tag{5.11}$$

then (5.10a) can be cast as $\mathbf{y}(t) = \mathbf{f}(\mathbf{a}(t))$. By using (5.10a) and the data sets $\mathcal{U}, \mathcal{D}$ and $\mathcal{A}$, we can easily calculate $\dot{\mathbf{a}}(t)$ at each sampling time and afterwards $\mathbf{y}(t)$ by means of (5.11). The evolution in time of $\mathbf{y}(t)$ is then compiled in the following data set

$$\mathcal{Y} = \{\mathbf{y}(0), \mathbf{y}(\Delta t), \ldots, \mathbf{y}(10000\Delta t)\}\,.$$

In order to make the training of the MLP more efficient, the input data $\mathcal{A}$ and the target outputs $\mathcal{Y}$ were normalized for zero mean and unit variance

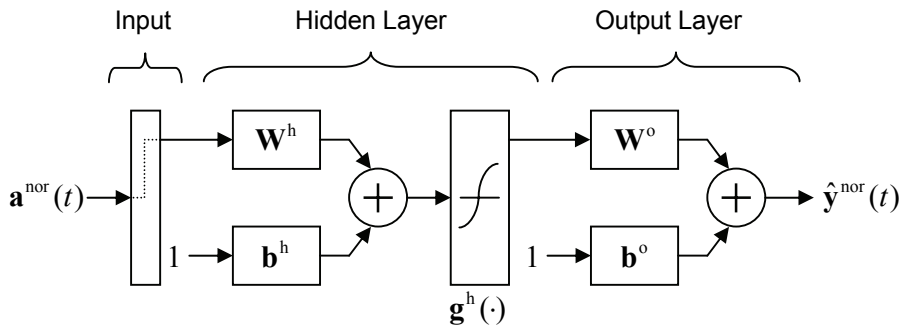


Figure 5.5: Structure of the multi-layer perceptron. $\mathbf{a}^{\text{nor}}(t) \in \mathbb{R}^6$ and $\hat{\mathbf{y}}^{\text{nor}}(t) \in \mathbb{R}^6$

by applying the normalization functions $\mathbf{h} : \mathbb{R}^6 \to \mathbb{R}^6$ and $\mathbf{v} : \mathbb{R}^6 \to \mathbb{R}^6$ to each element of the datasets $\mathcal{A}$ and $\mathcal{Y}$ respectively. The ith component function of the vector functions $\mathbf{h}$ and $\mathbf{v}$ is defined as

$$a_i^{\text{nor}}(t) = h_i\left(\mathbf{a}(t)\right) = \frac{a_i(t) - \bar{a}_i}{\sigma_{a_i}}, \tag{5.12}$$

$$y_i^{\text{nor}}(t) = v_i\left(\mathbf{y}(t)\right) = \frac{y_i(t) - \bar{y}_i}{\sigma_{y_i}}, \tag{5.13}$$

where $\bar{a}_i$, $\bar{y}_i$ and σ_{a_i}, σ_{y_i} are the mean and the standard deviation of $a_i(t)$ and $y_i(t)$ in $\mathcal{A}$ and $\mathcal{Y}$ respectively.

At the moment of using the MLP after training, the input data has to be normalized by using (5.12) and the output of the neural network needs to be restored using the inverse function of (5.13) whose ith component function is defined as follows:

$$\hat{y}_i(t) = v_i^{-1}\left(\hat{\mathbf{y}}^{\text{nor}}(t)\right) = \hat{y}_i^{\text{nor}}(t)\sigma_{y_i} + \bar{y}_i.$$

Here the "hat" on top of y_i and y_i^{nor} are used to stress that the output of the MLP is just an approximation of the target output. The data sets containing the normalized input data and output targets are denoted as $\mathcal{A}^{\text{nor}}$ and $\mathcal{Y}^{\text{nor}}$ respectively.

The structure of the MLP neural network designed for this problem is presented in Figure 5.5. The number of inputs and the number of output neurons is determined by the number of POD coefficients. So, there are 6 inputs in the input layer and 6 neurons in the output layer. Regarding the

number of hidden layers, the MLP has only one. It has been shown that merely one hidden layer suffices for approximating any function given an adequate number of hidden neurons with continuous nonlinear activation functions [67]. The output of the MLP is given by the following expression:

$$\hat{\mathbf{y}}^{\text{nor}}(t) = \mathbf{W}^{\text{o}} \cdot \mathbf{g}^{\text{h}} \left(\mathbf{W}^{\text{h}} \cdot \mathbf{a}^{\text{nor}}(t) + \mathbf{b}^{\text{h}} \right) + \mathbf{b}^{\text{o}} \tag{5.14}$$

where N_{hn} is the number of hidden neurons, $\mathbf{W}^{\text{h}} \in \mathbb{R}^{N_{\text{hn}} \times n}$ is the matrix of weights that links the input layer to the hidden layer, the entry W^{h}_{ji} of $\mathbf{W}^{\text{h}}$ corresponds to the connection weight from the ith input neuron to the jth neuron in the hidden layer, $\mathbf{b}^{\text{h}} \in \mathbb{R}^{N_{\text{hn}}}$ is the vector containing the bias weight of each neuron of the hidden layer, $\mathbf{W}^{\text{o}} \in \mathbb{R}^{n \times N_{\text{hn}}}$ is the matrix of weights that links the hidden layer to the output layer, the entry W^{o}_{ji} of $\mathbf{W}^{\text{o}}$ is the connection weight from the ith hidden neuron to the jth neuron in the output layer, $\mathbf{b}^{\text{o}} \in \mathbb{R}^{n}$ is the vector that contains the bias weight of each neuron of the output layer, and $\mathbf{g}^{\text{h}}(\cdot) : \mathbb{R}^{N_{\text{hn}}} \to \mathbb{R}^{N_{\text{hn}}}$ is a vector-valued function whose component functions are the nonlinear activation functions of the hidden neurons. The ith component function of $\mathbf{g}^{\text{h}}(\cdot)$ is a hyperbolic tangent function which is defined by the following equation

$$g^{\text{h}}_i(\mathbf{s}^{\text{h}}) = \frac{e^{2s^{\text{h}}_i} - 1}{e^{2s^{\text{h}}_i} + 1},$$

where $\mathbf{s}^{\text{h}} \in \mathbb{R}^{N_{\text{hn}}}$ is the vector containing the weighted sum of each hidden neuron.

The MLP was trained by using the Levenberg-Marquardt (LM) backpropagation algorithm [62] which is available in the Matlab Neural Network Toolbox [102]. In general, this algorithm offers a good speed of convergence and acceptable memory requirements when it is used for approximating functions with networks that contain up to a few hundred weights. In order to avoid the overfitting of the MLP, the early stopping method was used during the training, and therefore the data ($\mathcal{A}^{\text{nor}}$ and $\mathcal{Y}^{\text{nor}}$) was divided into 3 sets: the training set with 7001 data points, the validation set with 1500 data points and the test set with 1500 data points. The training set is used by the training algorithm for updating the network weights and biases, the validation set is used for detecting the overfitting during the training stage and the test set is used for testing the generalization capabilities of the MLP. The test set is never used during the training stage. The data was divided by cycling samples (interleaved data division) between training set, validation

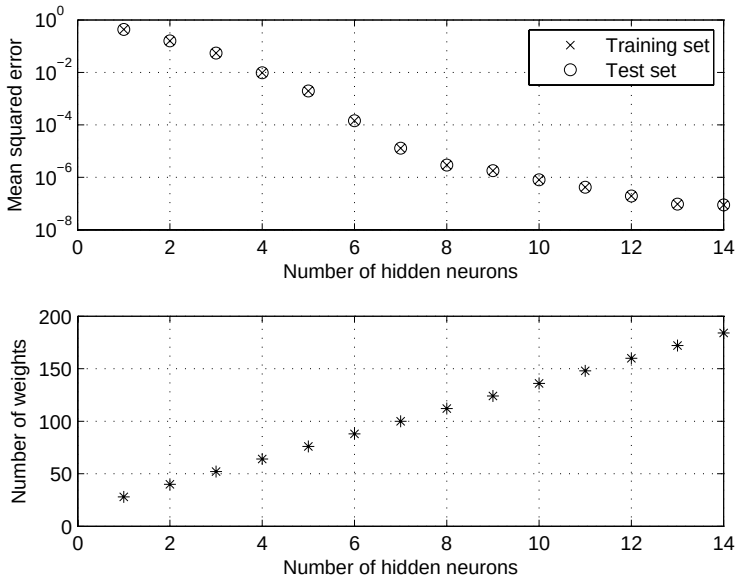


Figure 5.6: Mean squared error and size of the trained MLP in function of the number of hidden neurons.

set, and test set according to percentages. These percentages were 70%, 15% and 15% for the training, validation and test sets respectively.

Thus far, there is no theory that establishes the number of hidden units that are needed to achieve some prescribed degree of accuracy in a function approximation problem. Therefore, in order to find a suitable number of hidden units, we increased one by one the number of neurons of the hidden layer while checking the error in the approximation of the vector function $\mathbf{f}$. The Mean Squared Error (MSE) function was selected to measure the performance of the neural network. The top plot of Figure 5.6 shows the MSE between the target outputs and the MLP output in function of the number of hidden neurons. Each value shown in the plot corresponds to the smallest MSE found after 7 training sessions where the weights of the net were randomly initialized. The bottom plot of Figure 5.6 presents the number of weights of the network in function of the number of hidden units. We chose the MLP with $N_{\text{hn}} = 10$ hidden neurons since it provides a good trade-off between the accuracy of the function approximation and the size of the network (136 weights), which determines how fast the MLP can be evaluated and therefore the computational gain that can be obtained. The

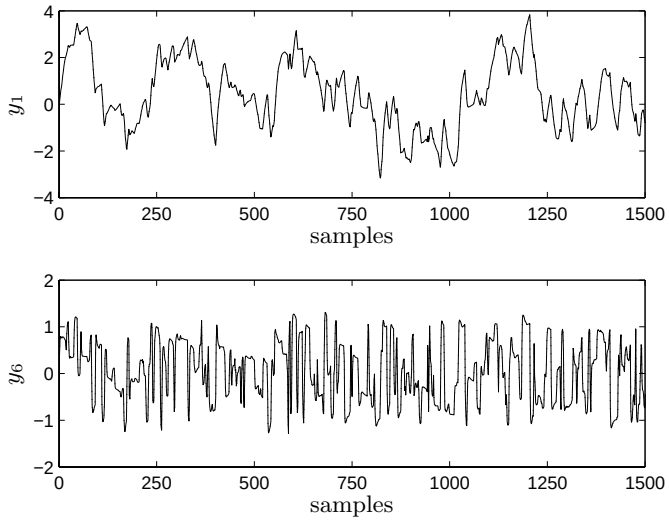


Figure 5.7: MLP test performance for the outputs $y_1(t)$ and $y_6(t)$. Solid line - Data points (targets). Dashed line - MLP.

MSE of this MLP for the training set is 8.0662×10^{-7} and for the test set is 8.0975×10^{-7}. These errors were achieved after 6000 epochs of training. Figure 5.7 shows the MLP output and the original data points (targets) for $y_1(t)$ and $y_6(t)$ when the test set is used. The MLP output is practically overlapping the data points, and it is really difficult to see any difference. It is clear that the network has learned the nonlinear input-output mapping $\mathbf{f}$ with a high degree of accuracy, and additionally the net has shown a good generalization capability. One factor that contributes to have very small MSE errors is the absence of noise in the data.

Finally, the equation of the POD model where the function $\mathbf{f}$ has been approximated by an MLP is the following one:

$$\dot{\mathbf{a}}(t) = \hat{\mathbf{f}}\left(\mathbf{a}(t)\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{5.15}$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*$$

with

$$\hat{\mathbf{f}}\left(\mathbf{a}(t)\right) = \mathbf{v}^{-1}\left(\mathbf{W}^{\mathrm{o}} \cdot \mathbf{g}^{\mathrm{h}}\left(\mathbf{W}^{\mathrm{h}} \cdot \mathbf{h}\left(\mathbf{a}(t)\right) + \mathbf{b}^{\mathrm{h}}\right) + \mathbf{b}^{\mathrm{o}}\right).$$

From now on, this POD model will be referred to as Neural-POD model.

5.5 Polynomial POD models

In this section we exploit the polynomial nature of the vector function $\mathbf{f}(\mathbf{a}(t))$ in (5.10a), in order to find an alternative representation of $\mathbf{f}(\mathbf{a}(t))$ that can be evaluated much faster.

We start by noticing that the mth component function of $\mathbf{f}(\mathbf{a}(t)) = \mathbf{\Phi}_n^T \mathbf{F}(\mathbf{T}_n(t))$ can be written as

$$f_m(\mathbf{a}(t)) = \sum_{i=1}^{N-1} \varphi_{i,m} F_i(\mathbf{T}_n(t)) \tag{5.16}$$

where $\varphi_{i,m}$ is the ith entry of the basis vector $\boldsymbol{\varphi}_m$, $N = 500$ is the number of sections in which the bar was divided, $n = 6$ is the number of selected basis vectors, $\mathbf{T}_n(t)$ is the nth order approximation of $\mathbf{T}(t)$ and F_i is the ith component function of $\mathbf{F}(\mathbf{T}_n(t))$ defined as follows:

$$\begin{aligned} F_i(\mathbf{T}_n(t)) = c_1 \Big(& \kappa(\tilde{T}_{i+1}(t))\tilde{T}_{i+1}(t) - \big(\kappa(\tilde{T}_{i+1}(t)) + \kappa(\tilde{T}_i(t))\big)\tilde{T}_i(t) + \\ & + \kappa(\tilde{T}_i(t))\tilde{T}_{i-1}(t) \Big), \end{aligned} \tag{5.17}$$

where c_1 is a constant given by (5.3b), $\kappa(T) : \mathbb{R} \to \mathbb{R}$ is the scalar polynomial function defined in (5.2) and $\tilde{T}_i(t)$ is the ith entry of the vector $\mathbf{T}_n(t)$ that is given by

$$\tilde{T}_i(t) = \sum_{j=1}^{n} \varphi_{i,j} a_j(t) + T_i^* \tag{5.18}$$

where $\varphi_{i,j}$ is the ith entry of the basis vector $\boldsymbol{\varphi}_j$, a_j is the jth POD coefficient and T_i^* is the ith entry of $\mathbf{T}^*$ which contains the initial temperature profile of the bar. If we replace (5.18) in (5.17),

$$\begin{aligned} F_i\left(\mathbf{a}(t)\right) = c_1 \Bigg(& \kappa\left(\sum_{j=1}^{n} \varphi_{(i+1),j} a_j(t) + T_{i+1}^*\right)\left(\sum_{j=1}^{n} \varphi_{(i+1),j} a_j(t) + T_{i+1}^*\right) - \\ & - \left(\kappa\left(\sum_{j=1}^{n} \varphi_{(i+1),j} a_j(t) + T_{i+1}^*\right) + \kappa\left(\sum_{j=1}^{n} \varphi_{i,j} a_j(t) + T_i^*\right)\right) \times \\ & \times \left(\sum_{j=1}^{n} \varphi_{i,j} a_j(t) + T_i^*\right) + \end{aligned}$$

$$+\kappa \left(\sum_{j=1}^{n} \varphi_{i,j} a_j(t) + T_i^* \right) \left(\sum_{j=1}^{n} \varphi_{(i-1),j} a_j(t) + T_{i-1}^* \right) \Bigg) \tag{5.19}$$

then we obtain the component function F_i in terms of the POD coefficients $a_j(t)\ \forall j = 1, \ldots, n$ associated to the selected basis vectors. Notice that this function is a multivariate polynomial of degree $d_{\mathrm{p}} = 4$ in $a_j(t)\ \forall j = 1, \ldots, n$. Since $f_m(\mathbf{a}(t))$ (see (5.16)) is a linear combination of the functions $F_i(\mathbf{a}(t))$ $\forall i = 1, \ldots, N-1$, $f_m(\mathbf{a}(t))$ is also a multivariate polynomial of degree $d_{\mathrm{p}} = 4$ in $a_j(t)\ \forall j = 1, \ldots, n$.

After replacing (5.19) in (5.16) and simplifying the resulting expression, we obtain the following general representation for $f_m(\mathbf{a}(t))$,

$$\begin{aligned} f_m(\mathbf{a}(t)) = w_{m,0} + w_{m,1} a_1(t) + \cdots + w_{m,n} a_n(t) + \\ + w_{m,(n+1)} a_1^2(t) + w_{m,(n+2)} a_1(t) a_2(t) + \cdots + w_{m,(r-1)} a_n^{d_{\mathrm{p}}}(t) \end{aligned} \tag{5.20}$$

where $w_{m,i}$ is the real coefficient of the ith monomial ($i = 0, 1, 2, \ldots, r-1$) of the function $f_m(\mathbf{a}(t))$, d_{p} is the largest possible degree for each monomial and r is the number of monomials which is given by the following formula [60]:

$$r = \sum_{j=0}^{d_{\mathrm{p}}} \left(\begin{array}{c} n+j-1 \\ j \end{array} \right) = \sum_{j=0}^{d_{\mathrm{p}}} \frac{(n+j-1)!}{j!\,(n-1)!} \,. \tag{5.21}$$

For this particular problem, the number of monomials per component function is equal to 210. The monomials (without coefficients) $a_{i_1} a_{i_2} \cdots a_{i_s}$ of degree s ($1 \leq s \leq d_{\mathrm{p}}$) in (5.20) are built from the set $\mathcal{C}$,

$$\mathcal{C} = \left\{ \langle i_1, i_2, \cdots, i_s \rangle : 1 \leq i_1, i_2, \ldots, i_s \in \mathbb{Z}^+ \leq n, i_1 \leq i_2 \leq \cdots \leq i_s \right\}$$

which contains the possible combinations of the indexes $i_1, i_2, \ldots, i_s$.

By using the general representation of $\mathbf{f}(\mathbf{a}(t))$ defined in (5.20) through its component functions, we can recast the POD model (5.10) as follows:

$$\begin{aligned} \dot{\mathbf{a}}(t) &= \mathbf{f}(\mathbf{a}(t)) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \\ \mathbf{T}_n(t) &= \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^* \end{aligned} \tag{5.22}$$

with

$$\mathbf{f}(\mathbf{a}(t)) = \left[f_1(\mathbf{a}(t)), \ldots, f_m(\mathbf{a}(t)), \ldots, f_n(\mathbf{a}(t)) \right]^T$$

$$f_m(\mathbf{a}(t)) = w_{m,0} + w_{m,1}a_1(t) + \cdots + w_{m,n}a_n(t) + \\ + w_{m,(n+1)}a_1^2(t) + w_{m,(n+2)}a_1(t)a_2(t) + \cdots + w_{m,(r-1)}a_n^{d_\mathrm{p}}(t).$$

From now on, this compact representation of (5.10) will be referred to as Polynomial POD model or P-POD model. Note that now the evaluation of $\mathbf{f}$ does not require the use of the high-dimensional vector function $\mathbf{F}$ anymore, and consequently we can expect to evaluate $\mathbf{f}$ much faster.

In general, if the discretized system used to approximate the PDE or PDEs is input affine and its nonlinearity is of polynomial type, then the vector function $\mathbf{f}(\mathbf{a}(t))$ can be represented efficiently by multivariate polynomials in terms of the POD coefficients. Notice that we might generate polynomial POD models from discretized systems with non-polynomial nonlinearities, provided that these nonlinearities can be approximated by low degree polynomials.

5.5.1 Calculation of the coefficients of the polynomials

In Section 5.4, we defined the vector $\mathbf{y}(t) \in \mathbb{R}^n$ (see Equation (5.11)) as

$$\mathbf{y}(t) = \dot{\mathbf{a}}(t) - \tilde{\mathbf{B}}_1 d(t) - \tilde{\mathbf{B}}_2 u(t),$$

and we stated that (5.10a) can be cast as $\mathbf{y}(t) = \mathbf{f}(\mathbf{a}(t))$.

The coefficients of the polynomials $f_m(\mathbf{a}(t))$ $\forall m = 1, \ldots, n$ in (5.22) are computed in such a way that the fitting to the data generated by (5.11) at the time instants $t = k\Delta t$, for $k = 0, 1, 2, \ldots, N_{\mathrm{d}-1}$,

$$\mathbf{y}_{N_\mathrm{d}} \in \mathbb{R}^{n \cdot N_d} = [\mathbf{y}(0); \mathbf{y}(1); \ldots; \mathbf{y}(k) = \mathbf{y}(t = k\Delta t); \ldots; \mathbf{y}(N_{\mathrm{d}-1})]$$

is optimal in the least squares sense. The vector $\mathbf{y}_{N_\mathrm{d}}$ is built from the data gathered during the snapshots experiment (see Section 5.3) where N_d samples of the temperature profile were collected. This vector is assembled as follows. From the snapshots experiment we know the values of $u(k) = u(t = k\Delta t)$ and $d(k) = d(t = k\Delta t)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$, and from the SVD of the snapshot matrix we have the evolution of $\mathbf{a}(k) = \mathbf{a}(t = k\Delta t)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$ (obtained by taking the first $n = 6$ rows of $\mathbf{\Sigma}\mathbf{\Psi}^T$). By using (5.10a) and the values of $u(k)$, $d(k)$ and $\mathbf{a}(k)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$, we calculate $\dot{\mathbf{a}}(k) = \dot{\mathbf{a}}(k\Delta t)$ and afterwards $\mathbf{y}(k) = \mathbf{y}(k\Delta t)$ by means of (5.11) for the same time instants.

Notice that the mth component function of $\mathbf{f}(\mathbf{a}(t))$ defined at the sampling times $t = k\Delta t$ (with $k = 0, 1, 2, \ldots$) can be written as follows:

$$f_m\left(\mathbf{a}(k)\right) = f_m\left(\mathbf{a}(t = k\Delta t)\right) = \boldsymbol{\alpha}(k)\tilde{\mathbf{w}}_m \tag{5.23}$$

where $\boldsymbol{\alpha}(k) \in \mathbb{R}^{1\times r}$ is a row vector defined as

$$\boldsymbol{\alpha}(k) = \Big[1, a_1(k), a_2(k), \ldots, a_n(k), a_1^2(k), a_1(k)a_2(k), \ldots, a_n^2(k), \ldots, a_1^{d_\mathrm{p}}(k), a_1^{d_\mathrm{p}-1}(k)a_2(k), \ldots, a_n^{d_\mathrm{p}}(k)\Big],$$

and $\tilde{\mathbf{w}}_m \in \mathbb{R}^r$

$$\tilde{\mathbf{w}}_m = \left[w_{m,0}, w_{m,1}, w_{m,2}, \ldots, w_{m,(r-1)}\right]^T$$

is a vector containing the coefficients of the monomials. From here the vector function $\mathbf{f}(\mathbf{a}(k)) = \mathbf{f}(\mathbf{a}(t = k\Delta t))$ can be compactly formulated in this way

$$\mathbf{f}(\mathbf{a}(k)) = \boldsymbol{\Lambda}(k)\mathbf{w} \tag{5.24}$$

where

$$\boldsymbol{\Lambda}(k) \in \mathbb{R}^{n\times(r\cdot n)} = \begin{bmatrix} \boldsymbol{\alpha}(k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\alpha}(k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\alpha}(k) \end{bmatrix},$$

and

$$\mathbf{w} \in \mathbb{R}^{r\cdot n} = [\tilde{\mathbf{w}}_1; \tilde{\mathbf{w}}_2; \ldots; \tilde{\mathbf{w}}_m; \ldots; \tilde{\mathbf{w}}_n]$$

is a vector containing all the coefficients that we want to find.

The predictions of (5.24) regarding $\mathbf{y}(k)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$, are given by the following expression

$$\hat{\mathbf{y}}_{N_\mathrm{d}} = \boldsymbol{\Omega}\mathbf{w} \tag{5.25}$$

where $\hat{\mathbf{y}}_{N_\mathrm{d}} \in \mathbb{R}^{n\cdot N_\mathrm{d}}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{(n\cdot N_\mathrm{d})\times(r\cdot n)}$ are defined as

$$\begin{aligned} \hat{\mathbf{y}}_{N_\mathrm{d}} &= [\mathbf{f}(\mathbf{a}(0)); \mathbf{f}(\mathbf{a}(1)); \ldots; \mathbf{f}(\mathbf{a}(k)); \ldots; \mathbf{f}(\mathbf{a}(N_\mathrm{d} - 1))] \\ \boldsymbol{\Omega} &= [\boldsymbol{\Lambda}(0); \boldsymbol{\Lambda}(1); \ldots; \boldsymbol{\Lambda}(k); \ldots; \boldsymbol{\Lambda}(N_\mathrm{d} - 1)]\,. \end{aligned}$$

As it was mentioned before, the coefficients $\mathbf{w}$ of the vector function $\mathbf{f}$ are found by means of the least squares method in which the minimization problem

$$\min_{\mathbf{w}} \; J = \left(\mathbf{y}_{N_\mathrm{d}} - \hat{\mathbf{y}}_{N_\mathrm{d}}\right)^T \left(\mathbf{y}_{N_\mathrm{d}} - \hat{\mathbf{y}}_{N_\mathrm{d}}\right) \tag{5.26}$$

is solved. We can obtain an equivalent optimization problem in terms of $\mathbf{w}$ by substituting (5.25) in (5.26) and ignoring the resulting constant term,

$$\min_{\mathbf{w}} \ \frac{1}{2}\mathbf{w}^T \left(\mathbf{\Omega}^T\mathbf{\Omega}\right)\mathbf{w} - \left(\mathbf{y}_{N_\mathrm{d}}^T\mathbf{\Omega}\right)\mathbf{w}. \tag{5.27}$$

As it is well-known, the solution of this minimization problem can be found by solving this equation: $\left(\mathbf{\Omega}^T\mathbf{\Omega}\right)\mathbf{w} = \mathbf{\Omega}^T\mathbf{y}_{N_\mathrm{d}}$. In this study this equation is solved by using the QR (orthogonal, triangular) decomposition of $\mathbf{\Omega}^T\mathbf{\Omega}$ and back substitution. This procedure is numerically more stable than inverting the matrix product $\mathbf{\Omega}^T\mathbf{\Omega}$.

Finally, it is important to remark that in the calculation of the monomial coefficients the data ($\mathbf{a}(k)$, $\mathbf{y}(k)$, for $k = 0, 1, 2, \ldots, N_d - 1$) was normalized first (by dividing by the standard deviation) in order to avoid numerical problems. Consequently, at the moment of evaluating the derivative of the P-POD model we have to carry out the appropriated conversions. For the sake of clarity and simplicity in our derivations, we did not include the normalization part in the previous equations.

5.5.2 Reduction of the number of monomials

From (5.21) it is clear that the number of monomials of the P-POD model can be very large. In fact, this number increases exponentially with the number of POD coefficients, compromising the computational gain of the P-POD model. We can tackle this situation by properly selecting a reduced set of monomials whose combination provides a good approximation of the vector function $\mathbf{f}(\mathbf{a}(t))$. To this end, in this section we propose the use of the sequential feature selection methods [61, 81, 105, 152]. Here we will use the term "features" for referring to as the "monomials" of $\mathbf{f}\left(\mathbf{a}(t)\right)$.

Sequential feature selection methods have basically two components: the objective function, called the criterion, which the method seeks to minimize over all feasible feature subsets, and the sequential search algorithm. The algorithm adds or removes features from a candidate subset while evaluating the criterion. In general, an exhaustive evaluation of all possible feature subsets is infeasible, since we have to deal with $2^r = 2^{210} = 1.64 \times 10^{63}$ candidates. This is why a suboptimal search strategy is necessary for directing the feature selection process as it explores the space of all possible combination of features. Representative examples of sequential search algorithms include the Sequential Forward Selection (SFS), the Sequential

Backward Selection (SBS), the Plus-L Minus-R Selection (LRS) and the Bidirectional Search (BDS) among others [152]. The first two search strategies, SFS and SBS, are the simplest greedy search algorithms that we can use. In the sequential forward selection, the features are sequentially added to an empty candidate set until a predetermined number of features has been added. In contrast, in the sequential backward selection, the features are sequentially removed from a full candidate set until a predefined number of features has been eliminated.

Let us define the set $\mathcal{S}$ as the set containing the positions of the selected features or monomials from vector $\boldsymbol{\alpha}(k)$ at a certain moment. We will denote the vector that contains these features as $\boldsymbol{\alpha}^{\mathcal{S}}(k)$ and the matrix $\boldsymbol{\Omega}$ (see Section 5.5.1) constructed from $\boldsymbol{\alpha}^{\mathcal{S}}(k)$ and the data points as $\boldsymbol{\Omega}^{\mathcal{S}}$. If we designate the vector comprising the coefficients of the selected features as $\mathbf{w}^{\mathcal{S}}$, then the predictions $\hat{\mathbf{y}}^{\mathcal{S}}_{N_\mathrm{d}} \in \mathbb{R}^{n \cdot N_\mathrm{d}}$ of the P-POD model with a reduced set of monomials about $\mathbf{y}(k)$ for $k = 0, 1, 2, \ldots, N_d - 1$, can be written as

$$\hat{\mathbf{y}}^{\mathcal{S}}_{N_\mathrm{d}} = \boldsymbol{\Omega}^{\mathcal{S}} \mathbf{w}^{\mathcal{S}}. \tag{5.28}$$

In this study, we use as objective function $J^{\mathcal{S}}$ or criterion the Sum Squared Error (SSE) between $\hat{\mathbf{y}}^{\mathcal{S}}_{N_\mathrm{d}}$ and $\mathbf{y}_{N_\mathrm{d}}$,

$$J^{\mathcal{S}} = \left(\mathbf{y}_{N_\mathrm{d}} - \hat{\mathbf{y}}^{\mathcal{S}}_{N_\mathrm{d}}\right)^T \left(\mathbf{y}_{N_\mathrm{d}} - \hat{\mathbf{y}}^{\mathcal{S}}_{N_\mathrm{d}}\right) \tag{5.29}$$

and as a search algorithm the sequential forward selection. We chose SFS instead of SBS because SFS is computationally less demanding due to the fact that the criterion function $J^{\mathcal{S}}$ is evaluated over smaller subsets of monomials.

For the evaluation of each candidate feature subset, a K-fold cross-validation scheme is used by the feature selection algorithm. In K-fold cross-validation the available data (the observations or data points are the rows of $\mathbf{y}_{N_\mathrm{d}}$ and $\boldsymbol{\Omega}^{\mathcal{S}}$) is partitioned in K samples. Of the K samples, a single sample is retained as the validation data for evaluating the candidate feature subset and computing its corresponding SSE, and the remaining $K - 1$ samples (training data) are used for calculating the coefficients of the monomials via least squares. The cross-validation process is then repeated K times (the folds), with each of the K samples used exactly once as the validation data. In order to generate a single goodness-of-fit measure of the candidate, the K SSEs are summed and the result divided by the number of observations or data points. Once this goodness-of-fit measure has been calculated for

all the candidate feature subsets, the algorithm picks the subset with the smallest measure.

The advantage of using K-fold cross-validation is that all the observations in the dataset are eventually used for both training and validation. For this problem a 10-fold cross-validation scheme was used.

In the next section it will be shown that the local stability around the origin of the P-POD models depends only on the linear terms, and consequently the algorithm was properly configured to include these terms in the candidate feature subsets.

After running the feature selection algorithm, $N_{\text{sm}} = 25$ features or monomials were finally chosen ($J^{\mathcal{S}} = 1.3727 \cdot 10^{-4}$). These selected features are pointed out by the set $\mathcal{S}^*$, which contains the suboptimal solution found by the algorithm. Once this set is established, the coefficients of the monomials $\mathbf{w}^{\mathcal{S}^*} \in \mathbb{R}^{N_{\text{sm}} \cdot n}$ are computed via least squares.

Lastly, the polynomial POD model with a reduced set of monomials, which will be referred to as P-POD-RS model, has the following form,

$$\dot{\mathbf{a}}(t) = \mathbf{f}^{\mathcal{S}^*}(\mathbf{a}(t)) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{5.30}$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*$$

with

$$\mathbf{f}^{\mathcal{S}^*}(\mathbf{a}(t)) = \mathbf{\Lambda}^{\mathcal{S}^*}(t)\mathbf{w}^{\mathcal{S}^*}$$

$$\mathbf{\Lambda}^{\mathcal{S}^*}(t) \in \mathbb{R}^{n \times (N_{\text{sm}} \cdot n)} = \begin{bmatrix} \boldsymbol{\alpha}^{\mathcal{S}^*}(t) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\alpha}^{\mathcal{S}^*}(t) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\alpha}^{\mathcal{S}^*}(t) \end{bmatrix}$$

where

$$\begin{aligned} \boldsymbol{\alpha}^{\mathcal{S}^*}(t) = \big[& a_1(t), a_2(t), a_3(t), a_4(t), a_5(t), a_6(t), a_1^2(t), a_1(t)a_2(t), a_1(t)a_3(t), \\ & a_1(t)a_4(t), a_1(t)a_5(t), a_1(t)a_6(t), a_2^2(t), a_2(t)a_3(t), a_2(t)a_4(t), \\ & a_2(t)a_5(t), a_2(t)a_6(t), a_3(t)^2, a_3(t)a_4(t), a_3(t)a_5(t), a_3(t)a_6(t), \\ & a_4^2(t), a_1(t)^2 a_2(t), a_1^4(t), a_1^2(t)a_2(t)a_6(t) \big] \end{aligned}$$

contains the 25 selected monomials.

5.6 Polynomial POD models with stability guarantee

A property that is desirable to preserve in the model reduction process is the stability of the original model. In this aspect, the model reduction of an stable model by POD often results in a reduced model that is stable as well. Nevertheless, this is generally not guaranteed. Eventually we can end up with an unstable model, specially when the quality of the available data is poor [119].

For explaining our methods, in this section we are going to use the polynomial POD model with the full set of monomials described by (5.22). However, these methods can be also applied in a very straightforward way to the case of the polynomial POD model with a reduced set of monomials (see (5.30)).

Since global stability of a nonlinear system is in most cases hard or even impossible to prove, this study will be focused on the local stability of the P-POD model (5.22) around the origin. The stability of (5.22) is analyzed through its autonomous counterpart,

$$\dot{\mathbf{a}}(t) = \breve{\mathbf{f}}(\mathbf{a}(t)) = \mathbf{f}(\mathbf{a}(t))\big|_{w_{1,0}=w_{2,0}=\cdots=w_{n,0}=0}. \tag{5.31}$$

which is found by making $u(t)$, $d(t)$ and the constant terms $w_{1,0}, w_{2,0}, \ldots, w_{n,0}$ of the vector function $\mathbf{f}(\mathbf{a}(t))$ equal to zero. Notice that the constant monomials can be seen as the entries of an additional vector $\tilde{\mathbf{B}}_{\mathbf{3}} = [w_{1,0}, w_{2,0}, \ldots, w_{n,0}]^T$ multiplied by an extra input $u_{\mathrm{e}}(t)$ equal to one.

We can derive a stability condition for the autonomous system (5.31) by using the Lyapunov's indirect method. According to this method, the stability of the origin in (5.31) is inferred from the stability of the linearized system,

$$\delta\dot{\mathbf{a}}(t) = \mathbf{A}\delta\mathbf{a}(t). \tag{5.32}$$

Here $\delta\mathbf{a}(t) \in \mathbb{R}^n$ is the deviation variable and $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the jacobian matrix defined as follows:

$$\mathbf{A} = \left.\frac{\partial \breve{\mathbf{f}}}{\partial \mathbf{a}}\right|_{\mathbf{a}=\mathbf{0}} = \left.\begin{bmatrix} \frac{\partial \breve{f}_1}{\partial a_1} & \frac{\partial \breve{f}_1}{\partial a_2} & \cdots & \frac{\partial \breve{f}_1}{\partial a_n} \\ \frac{\partial \breve{f}_2}{\partial a_1} & \frac{\partial \breve{f}_2}{\partial a_2} & \cdots & \frac{\partial \breve{f}_2}{\partial a_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \breve{f}_n}{\partial a_1} & \frac{\partial \breve{f}_n}{\partial a_2} & \cdots & \frac{\partial \breve{f}_n}{\partial a_n} \end{bmatrix}\right|_{\mathbf{a}=\mathbf{0}}$$

$$= \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{bmatrix}.$$

According to the Lyapunov's indirect method, the origin is asymptotically stable if $\mathbf{A}$ is Hurwitz, i.e., all the eigenvalues of $\mathbf{A}$ have a negative real part, $\mathrm{Re}[\lambda_i(\mathbf{A})] < 0$, $\forall i = 1, \ldots, n$. Given that the jacobian matrix $\mathbf{A}$ is made of the coefficients of the linear terms, it is clear that these coefficients are the only ones that affect the stability of the P-POD model around the origin. If we add the eigenvalue constraint $\mathrm{Re}[\lambda_i(\mathbf{A}(\mathbf{w}))] < 0$ to the minimization problem (5.27), then we end up with the following non-convex optimization problem:

$$\min_{\mathbf{w}} \ \frac{1}{2}\mathbf{w}^T \left(\mathbf{\Omega}^T \mathbf{\Omega}\right) \mathbf{w} - \left(\mathbf{y}_{N_\mathrm{d}}^T \mathbf{\Omega}\right) \mathbf{w} \tag{5.33a}$$

subject to

$$\mathrm{Re}[\lambda_i(\mathbf{A}(\mathbf{w}))] < 0, \quad \text{for } i = 1, \ldots, n. \tag{5.33b}$$

The eigenvalue constraint makes problem (5.33) non-smooth [127], which in general is difficult to solve in its current form [25]. In order to overcome this situation, in this section we present two manners of dealing with this constraint. In our first approach, we replace the eigenvalue constraint by an LMI, which is a relaxation of the original constraint, providing a sufficient condition for the local stability of the P-POD model. In our second approach, the eigenvalue constraint is substituted by a nonlinear matrix equality and an LMI. Unlike the first approach, these replacements give sufficient and necessary conditions for the local stability of the reduced order model. In both cases, the local stability is guaranteed independently of data used. Nevertheless, the quality of the reduced-oder model still depends on the data used in the model reduction process.

5.6.1 Semidefinite problem formulation

A relaxation of the eigenvalue constraint (5.33b) is given in the following lemma:

Lemma 5.1 ([119]). *Let* $\mathbf{A}$ *be a square matrix. If the Hermitian part of* $\mathbf{A}$*, i.e.,* $\frac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{H}}\right)$*, is negative definite, then* $\mathbf{A}$ *is Hurwitz.*

Proof: See [119]. ■

By using Lemma 5.1, we can modify the optimization problem (5.33) as follows:

$$\min_{\mathbf{w}} \ \frac{1}{2}\mathbf{w}^T \left(\mathbf{\Omega}^T\mathbf{\Omega}\right)\mathbf{w} - \left(\mathbf{y}_{N_\mathrm{d}}^T\mathbf{\Omega}\right)\mathbf{w} \tag{5.34a}$$

subject to

$$-\frac{1}{2}\left(\mathbf{A}(\mathbf{w}) + \mathbf{A}(\mathbf{w})^T\right) - \mu\mathbf{I} \succeq 0 \tag{5.34b}$$

where $\mu > 0$ is a positive number used to guarantee the strictly positive definiteness of $-\frac{1}{2}\left(\mathbf{A}(\mathbf{w}) + \mathbf{A}(\mathbf{w})^{\mathbf{T}}\right)$. The inclusion of μ is necessary given that most solvers do not work with strict positive/negative definite constraints. The optimization problem (5.34) is nothing else than a SDP problem that can be solved efficiently by interior point methods, such as the one implemented in the Matlab Toolbox Sedumi [140].

It should be clear that not all the Hurwitz matrices have negative-definite hermitian parts (the Lemma 5.1 only provides a sufficient condition for the local stability of the model), and therefore the stability constraint (5.34b) might be very conservative.

In the following subsection a non-conservative replacement of the eigenvalue constraint (5.33b) will be discussed.

5.6.2 Nonlinear semidefinite problem formulation

Alternatively as it was done in the previous subsection, we can substitute the eigenvalue constraint (5.33b) by the well-known Lyapunov 's matrix equality (this equality is derived from the application of the Lyapunov's direct method on linear time-invariant systems) presented in the next theorem.

Theorem 5.1. *Given the autonomous system,* $\delta\dot{\mathbf{a}}(t) = \mathbf{A}\delta\mathbf{a}(t)$*, the origin* $\delta\mathbf{a}(t) = 0$ *is asymptotically stable if and only if, for any symmetric positive definite matrix* $\mathbf{Q}$*, there exists a symmetric positive definite matrix* $\mathbf{P}$ *such that,*

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = \mathbf{0}.$$

Theorem 5.1 gives a sufficient and necessary condition for the stability of the P-POD model around the origin. By using this theorem we can adapt

the minimization problem (5.33) in the following way,

$$\min_{\mathbf{w},\mathbf{P}} \ \frac{1}{2}\mathbf{w}^T \left(\mathbf{\Omega}^T\mathbf{\Omega}\right)\mathbf{w} - \left(\mathbf{y}_{N_\mathrm{d}}^T\mathbf{\Omega}\right)\mathbf{w} \tag{5.35a}$$

subject to

$$\mathbf{A}(\mathbf{w})^T\mathbf{P} + \mathbf{P}\mathbf{A}(\mathbf{w}) + \mathbf{Q} = \mathbf{0} \tag{5.35b}$$

$$\mathbf{P} - \tilde{\mu}\mathbf{I} \succeq 0 \tag{5.35c}$$

where $\mathbf{Q} \in \mathbb{S}^n_{++}$ is a given positive definite matrix, and $\tilde{\mu}$ is a positive number used to guarantee the strictly positive definiteness of $\mathbf{P}$. Problem (5.35) is a nonlinear SDP problem which in general is non-convex. More precisely, we have an optimization problem that involves a Bilinear Matrix Inequality (BMI), $\mathbf{A}(\mathbf{w})^T\mathbf{P}+\mathbf{P}\mathbf{A}(\mathbf{w}) \prec 0$ or $\mathbf{A}(\mathbf{w})^T\mathbf{P}+\mathbf{P}\mathbf{A}(\mathbf{w})+\tilde{\mu}\mathbf{I} \preceq 0$. We can solve problem (5.35) by using the Matlab toolbox PENBMI, which implements a general-purpose solver for BMI problems. The algorithm implemented in PENBMI is described in [83] and it is based on the augmented Lagrangian method. It can be viewed as a generalization to nonlinear semidefinite problems of the penalty-barrier-multiplier method originally introduced in [21] for convex optimization [65].

In order to get a feasible starting point ($\mathbf{w}_0$, $\mathbf{P}_0$) for the BMI solver, and properly set the constraints (5.35b) and (5.35c), we can solve the convex optimization problem stated in (5.34). Thus, the starting point and constraints parameters might be fixed as follows:

- $\mathbf{w}_0$ is obtained directly by solving (5.34)
- $\mathbf{P}_0 = 0.5 \cdot \mathbf{I}$
- $\mathbf{Q} = -0.5\left(\mathbf{A}_0 + \mathbf{A}_0^T\right)$, where $\mathbf{A}_0$ is constructed from $\mathbf{w}_0$
- $0 < \tilde{\mu} \leq 0.5$.

Unlike (5.34b), the stability constraint given by (5.35b) and (5.35c) is non-conservative, but leads to a non-convex optimization problem.

5.6.3 Numerical example

Consider the following autonomous linear system (from [119]),

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{A}}\mathbf{x}(t) \tag{5.36}$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} -2.3 & 0.5 & 0 & 0 & 2.3 & -0.8 \\ 1.5 & -2.5 & 0 & 0 & -4.5 & 1.5 \\ -22.8 & -0.5 & -25 & 0 & -2.3 & 0.8 \\ -33 & -0.6 & 15.6 & -25.8 & -2.6 & 13.5 \\ 8.4 & 1.0 & 3.7 & 2.8 & -0.5 & -1.4 \\ 25 & 0.9 & 11.1 & 8.4 & 3.9 & -6.0 \end{bmatrix}.$$

The eigenvalues of $\tilde{\mathbf{A}}$ are -0.4704, -1.5865, $-2.5335 \pm 1.5514i$, -24.9998 and -29.9763 and clearly the model is stable. In this example we are interested in deriving a reduced order model (a P-POD model with only linear terms) of (5.36) by using POD. To this end, we first built the snapshot matrix,

$$\mathbf{X}_{\text{snap}} = \big[\mathbf{x}(0), \mathbf{x}(t = \Delta t), \mathbf{x}(t = 2\Delta t), \ldots, \mathbf{x}(t = 480\Delta t)\big], \quad \Delta t = 0.025\,\text{s}$$

from the evolution of the state vector $\mathbf{x}(t)$ starting from the initial condition $\mathbf{x}(t = 0) = [-0.5, 1, 0.5, 1, 0.5, 1]^T$. Afterwards, we derived the POD basis vectors,

$$\mathbf{\Phi} = \begin{bmatrix} -0.0401 & 0.3354 & 0.1237 & -0.2280 & 0.5646 & 0.7070 \\ 0.0573 & -0.7177 & 0.1541 & 0.3907 & 0.5525 & 0.0016 \\ 0.0402 & -0.3364 & -0.1234 & 0.2224 & -0.5661 & 0.7072 \\ 0.4758 & -0.3934 & -0.2819 & -0.7335 & 0.0370 & -0.0031 \\ 0.3238 & -0.0222 & 0.9099 & -0.1388 & -0.2180 & -0.0010 \\ 0.8138 & 0.3225 & -0.1959 & 0.4343 & 0.0820 & 0.0020 \end{bmatrix}$$

by computing the SVD of $\mathbf{X}_{\text{snap}}$. The singular values of $\mathbf{X}_{\text{snap}}$ and their associated $1 - \bar{P}_n$ (see Equation (2.7)) values are : $\sigma_1 = 12.7308$ $(1 - \bar{P}_1 = 5.21 \cdot 10^{-2})$, $\sigma_2 = 2.9377$ $(1 - \bar{P}_2 = 1.6 \cdot 10^{-3})$, $\sigma_3 = 0.4285$ $(1 - \bar{P}_3 = 5.49 \cdot 10^{-4})$, $\sigma_4 = 0.3059$ $(1 - \bar{P}_4 = 1.858 \cdot 10^{-6})$, $\sigma_5 = 0.0178$ $(1 - \bar{P}_5 = 4.878 \cdot 10^{-11})$, and $\sigma_6 = 9.133 \cdot 10^{-5}$ $(1 - \bar{P}_6 = 0)$.

From the previous values, it is clear that the subspace spanned by the first 3 or 4 basis vectors contains most of the "energy" in the data. Therefore, we derived POD models of orders 3 and 4 by solving the optimization problems (5.27), (5.34) and (5.35). The POD model without stability constraint obtained by solving (5.27) or by using Galerkin will be referred to as P-POD-NC. The POD models that include the SDP and NSDP stability constraints will be referred to as P-POD-SDP and P-POD-NSDP respectively.

Table 5.1 presents the eigenvalues of the state matrix of each POD model as well as the cost function value at the solution of the associated optimization

Table 5.1: Numerical Results

Model	Order $n = 3$		Order $n = 4$	
	J	λ_i	J	λ_i
P-POD-NC	-81	-2.8236 -0.6355 $+0.1010$	-173.1056	-0.4729 $-2.4805 \pm 1.4922i$ -29.1026
P-POD-SDP	-80.0536	-3.0384 -2.0838 -0.6233	-168.9984	-0.5544 $-4.0173 \pm 1.2011i$ -30.0831
P-POD-NSDP	-80.9989	-2.8239 -0.6347 -0.0046	-173.1056	-0.4729 $-2.4805 \pm 1.4922i$ -29.1026

J is the cost function value at the solution of the optimization problems (5.27), (5.34) and (5.35).

λ_i are the eigenvalues of the state matrix of the P-POD model.

$\mu = \tilde{\mu} = 10^{-8}$.

problem. In addition, Figures 5.8 and 5.9 show the response of the POD models of order 3 and 4 to the initial conditions $\mathbf{x}(t = 0) = [-0.5, 1, 0.5, 1, 0.5, 1]^T$ respectively. Note that for generating the time response of the models we are using the same initial conditions than the ones used in the generation of the snapshot matrix. It should be clear that we could use different initial conditions, but in that case we can not expect accurate predictions from the POD models since their derivation is based on data that only consider the trajectory of the states when the initial conditions are the mentioned ones (the data is not rich enough).

For the case of $n = 3$, only the P-POD-NC model is unstable due to its eigenvalue at 0.1010 that is outside the stability region. By comparing the eigenvalues of the P-POD-NC and P-POD-NSDP models, we can see that the unstable pole 0.1010 of P-POD-NC is replaced by an stable one (-0.0046) in the P-POD-NSDP case (the other eigenvalues are practically the same). However, this pole makes the model dynamics very slow and this condition could lead to a misinterpretation of Figure 5.8, since the steady state values of the states in the P-POD-NSDP case (dash-dotted line), seem to be different from zero. In reality the states go very slowly towards 0 in approximately 1000 s. Notice also that the cost function value in the P-POD-NSDP case is smaller than in the POD-SDP case, but from Figure

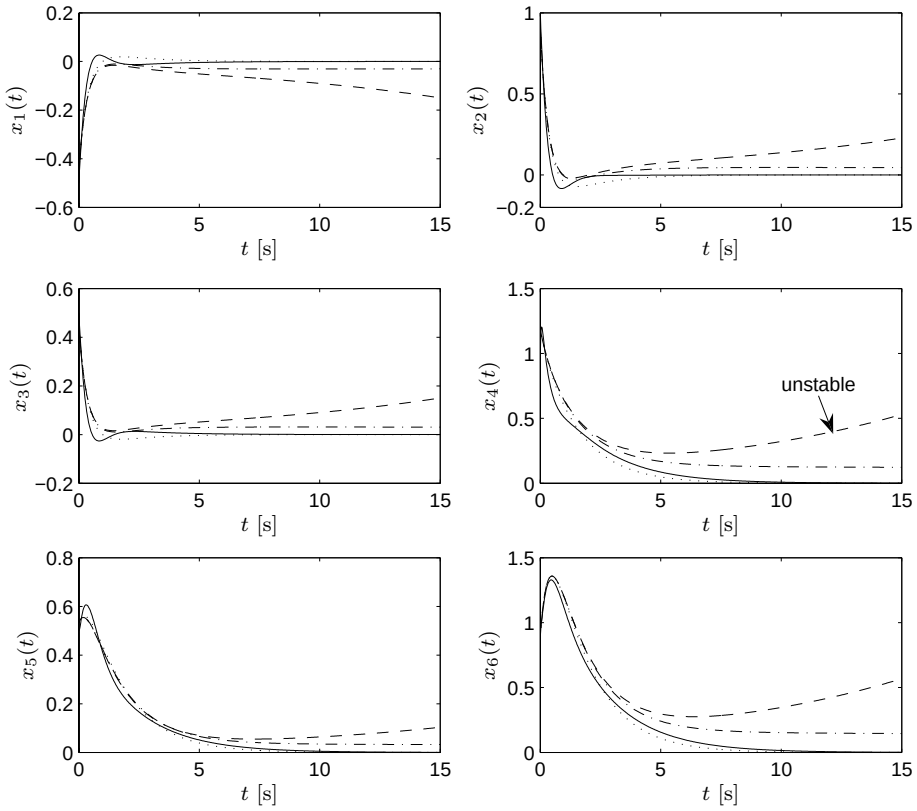


Figure 5.8: Response of the POD models of order 3 to the initial conditions $\mathbf{x}(t=0) = [-0.5, 1, 0.5, 1, 0.5, 1]^T$. Solid line - Full order model. Dashed line - P-POD-NC model. Dotted line - P-POD-SDP model. Dash-dotted line - P-POD-NSDP model.

5.8 it is evident that the P-POD-SDP model offers a better approximation of the original system. These observations might look contradictory, but we have to bear in mind that the cost function value only indicates how well the polynomials fit the data, and not how accurate the POD model will be.

From Table 5.1 it is clear that the cost function value in the P-POD-SDP case is larger than in the P-POD-NC and P-POD-NSDP cases. Furthermore from Figure 5.9 we can see some small differences in the responses of P-POD-SDP and (5.36). These differences are not observed in the P-POD-NC and P-POD-NSDP cases where the response of both these models and the original system (5.36) are practically overlapping. All this is consequence of

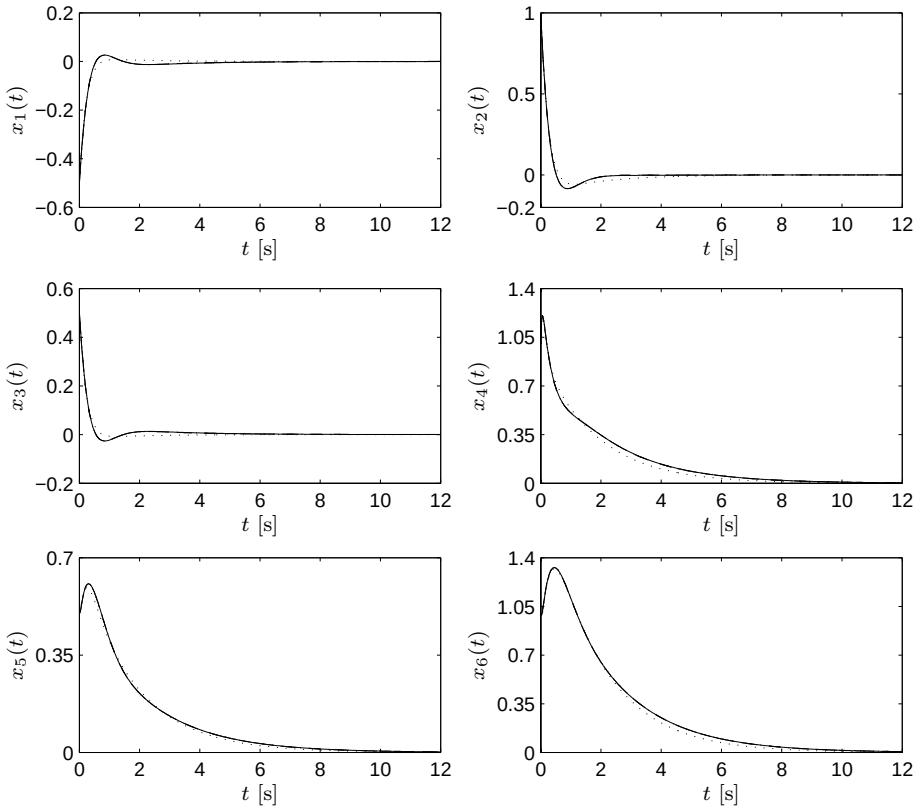


Figure 5.9: Response of the POD models of order 4 to the initial conditions $\mathbf{x}(t=0) = [-0.5, 1, 0.5, 1, 0.5, 1]^T$. Solid line - Full order model. Dashed line - P-POD-NC model. Dotted line - P-POD-SDP model. Dash-dotted line - P-POD-NSDP model.

the conservative nature of the SDP constraint. Nevertheless, it is important to keep in mind that this constraint is only imposed on the linear terms. Therefore if we are dealing with higher degree P-POD models, the limitation in the search space would be less significant, since the optimization algorithm would have several unconstrained optimization variables to play with.

Although the NSDP constraint is non-conservative, its drawback is to lead to a non-convex optimization where we have to deal with the local minima problem. When $n = 4$ we can see that the P-POD-NSDP and P-POD-NC models are identical. However, in general this is not always the case, and we can end up with different models since the optimization with the NSDP

constraint might be trapped in a local minimum.

The approaches considered in this section for guaranteeing the local stability of polynomial POD models, demand much more computational burden than the calculation of polynomial POD models by using least squares. Particularly the calculation of the P-POD model with the NSDP constraint is the most demanding case, and this fact might limit its practical applicability when the number of monomials increases as a consequence of augmenting either the number of selected POD basis vectors or the degree of the polynomials.

5.7 Validation and simulation results

In this section we will validate and compare the reduced-order models of the bar that were presented in Sections 5.3, 5.4 and 5.5, as well as a POD model derived using the Missing Point Estimation (MPE) technique. This technique, which is introduced in [11, 12], is used for accelerating nonlinear and linear time variant POD models.

In the MPE method, the Galerkin projection is conducted only on some preselected state variables or points of the spatial domain, instead of the entire set. The remaining state variables are estimated by means of the POD basis vectors. The fact of using a subset of points of the spatial domain leads to a reduction of the time required for evaluating the POD model. In this study, we found a POD model for the bar using this technique, and such a model will be referred to as MPE-POD model. The selection of points was done by using the second screening criterion and the greedy algorithm described in [11,12]. Since the boundary conditions must be satisfied by the reduced-order model, the points adjacent to them have to be included. A total of 182 points from 499 were selected, including the 2 points adjacent to the ends of the bar. This number of points offers a good compromise between the accuracy of the model and its computational gain.

In order to validate and evaluate the different POD models of the bar, the following tests were carried out:

- Test 1: A step of magnitude $1200 \cdot 10^3$ W $\cdot$ m^{-3} is applied to $u(t)$ and a step of magnitude $500 \cdot 10^3$ W $\cdot$ m^{-3} is applied to $d(t)$.

- Test 2: Steps of magnitude $-1100 \cdot 10^3$ W$\cdot$m^{-3} and $-400 \cdot 10^3$ W$\cdot$m^{-3} are applied to $u(t)$ and $d(t)$ respectively.
- Test 3: A step of magnitude $500 \cdot 10^3$ W$\cdot$m^{-3} is applied to $u(t)$ and a step of magnitude $-200 \cdot 10^3$ W$\cdot$m^{-3} is applied to $d(t)$.
- Test 4: Steps of magnitude $-400 \cdot 10^3$ W$\cdot$m^{-3} and $300 \cdot 10^3$ W$\cdot$m^{-3} are applied to $u(t)$ and $d(t)$ respectively.

Given that the discretized model of the bar (5.4) consists of a set of stiff differential equations, an ODE solver that can deal with this condition has to be used. Hence, we used the function *ode23tb* of Matlab which implements TR-BDF2, an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two [132]. Along this work, not only (5.4) was solved with *ode23tb*, but also the ODEs describing each of the reduced-order models. The solver was configured with a variable integration step and with a relative tolerance of 10^{-5} in all the cases. The initial conditions for the POD models were given by $\mathbf{a}(0) = \mathbf{\Phi}_n^T \mathbf{T}_n^{\Delta}(0) = \mathbf{0}$.

In Table 5.2 we present the computational gain of the POD models with respect to the full order model (5.4), and we also include a measure of their accuracy. In this table, $\Delta T_{\max}$ is the largest temperature deviation (error) of the POD models regarding the high-dimensional model (5.4) along the entire test, and G_{d} and G_{s} quantify the computational gain of the POD models with respect to the full order model. They are defined as follows:

$$G_{\mathrm{d}} = \frac{\tilde{t}_{\mathrm{fom}}}{\tilde{t}_{\mathrm{pod}}} \tag{5.37}$$

$$G_{\mathrm{s}} = \frac{t_{\mathrm{fom}}}{t_{\mathrm{pod}}} \tag{5.38}$$

where t_{fom} and t_{pod} are the times spent by the ODE solver for simulating the full order model and the POD model respectively, and $\tilde{t}_{\mathrm{fom}}$ and $\tilde{t}_{\mathrm{pod}}$ are the average times for calculating the derivatives along the test of the full order model and the POD model respectively. The values of G_{d} and G_{s} in Table 5.2 are average values found after 15000 and 1500 runs respectively. All the simulations were carried out on a PC with an Intel dual core of 3 Ghz and a RAM memory of 2 GB.

Figures 5.10, 5.11, 5.12 and 5.13 show the maximum temperature deviation of the different POD models with respect to the full order model along

Table 5.2: Performance of the POD models

Test	POD model [a]			Neural-POD model [b]			MPE-POD model (182)			P-POD model [c]			P-POD-RS model [d]		
	G_{d}	G_{s}	$\Delta T_{\max}$	G_{d}	G_{s}	$\Delta T_{\max}$	G_{d}	G_{s}	$\Delta T_{\max}$	G_{d}	G_{s}	$\Delta T_{\max}$	G_{d}	G_{s}	$\Delta T_{\max}$
1	0.97	2.02	0.423	9.66	8.153	0.689	2.08	3.75	0.695	5.61	5.00	0.423	12.89	8.22	0.551
2	0.97	3.61	0.279	9.32	11.69	0.278	2.08	5.76	0.45	5.47	8.80	0.279	12.88	12.48	0.275
3	0.97	4.52	0.0518	9.32	14.03	0.0519	2.10	7.45	0.0581	5.48	9.83	0.0518	12.91	13.85	0.0518
4	0.97	3.52	0.0594	9.30	11.43	0.0593	2.09	5.58	0.088	5.48	8.34	0.0594	12.92	12.72	0.0594

[a] See (5.10).
[b] See (5.15).
[c] See (5.22).
[d] See (5.30).
G_{d} is the computational gain in the calculation of the derivatives.
G_{s} is the computational gain in the simulation of the model.
$\Delta T_{\max}$ is the largest temperature deviation (error) in degrees Celsius of the POD model.

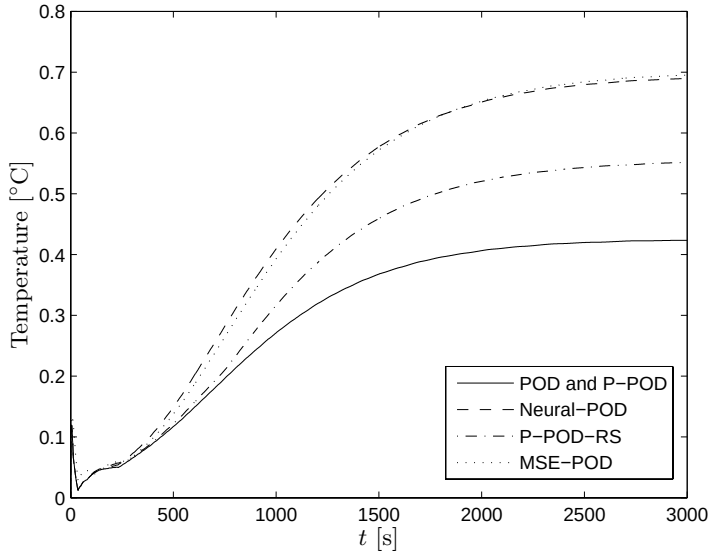


Figure 5.10: Maximum temperature deviation of the POD models with respect to the full order model along Test 1.

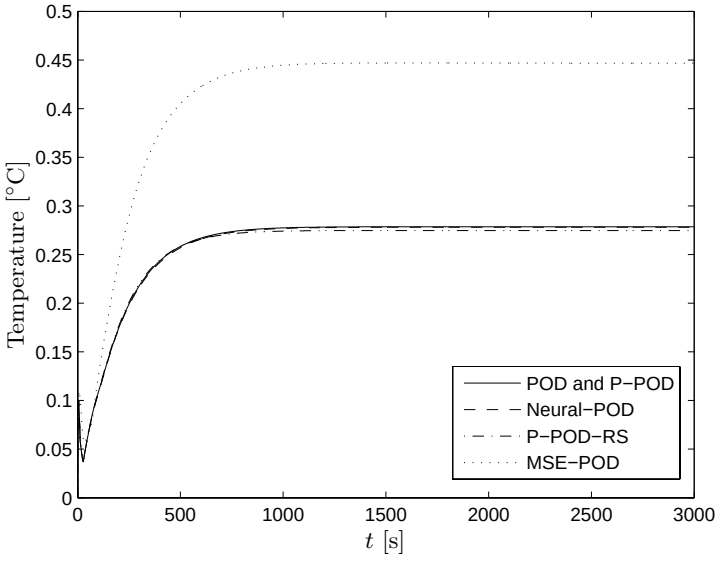


Figure 5.11: Maximum temperature deviation of the POD models with respect to the full order model along Test 2.

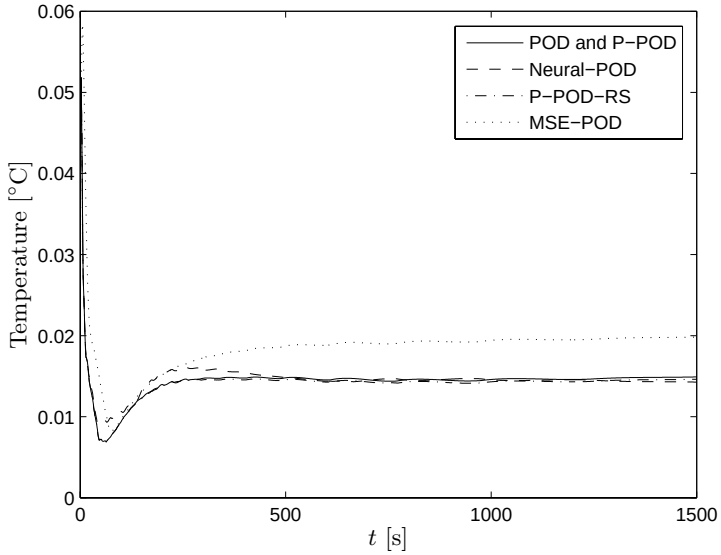


Figure 5.12: Maximum temperature deviation of the POD models with respect to the full order model along Test 3.

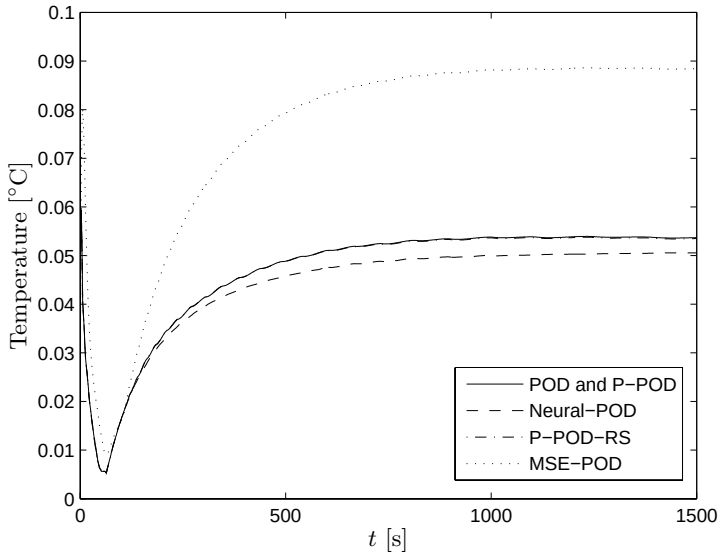


Figure 5.13: Maximum temperature deviation of the POD models with respect to the full order model along Test 4.

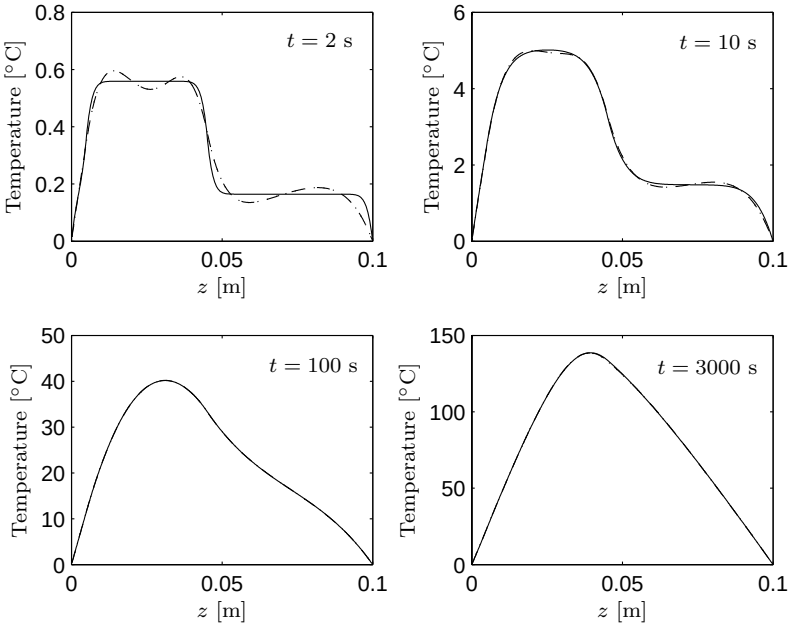


Figure 5.14: Temperature profile at different time steps during Test 1. Solid line - Full order model (5.4). Dashed line - POD model and P-POD model. Dotted line - Neural-POD model. The temperatures are relative to the ambient temperature (25°C).

the tests, and Figures 5.14 and 5.15 depict the evolution of the temperature profile of the bar during Test 1, the most severe test. In general, the accuracy of the POD models is good in spite of the big model-order reduction. Notice however that the MPE-POD model tends to be the less accurate model. We might select more points to improve its accuracy but this would compromise its computational gain. From Figures 5.14 and 5.15 we can observe how the temperature profiles of the POD models practically overlap each other.

From Table 5.2 we can notice that the derivatives of the P-POD model are calculated about 5.7 times faster than in the POD case. This computational gain has been achieved by exploiting the polynomial nature of the vector function $\mathbf{f}(\mathbf{a}(t))$ of (5.10) that allow us to write $\mathbf{f}(\mathbf{a}(t))$ in an efficient and compact form. This gain makes the simulation of the P-POD model about 2.4 times faster than the simulation of the POD model. Given that both models are equivalent, they have the same accuracy. Regarding the P-POD-

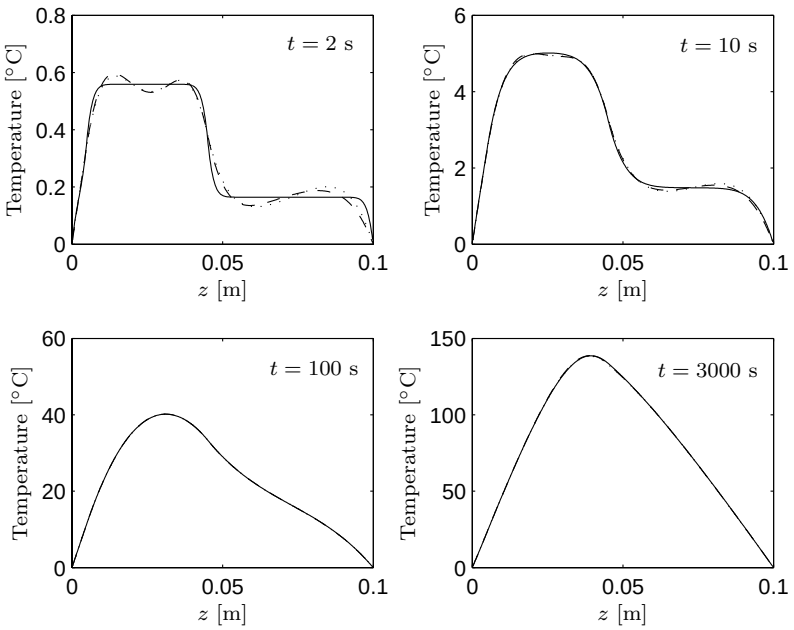


Figure 5.15: Temperature profile at different time steps during Test 1. Solid line - Full order model (5.4). Dashed line - P-POD-RS model. Dotted line - MPE-POD model. The temperatures are relative to the ambient temperature (25°C).

RS model, the calculation of the derivatives is performed about 13.3 times faster than in the POD case and 2.3 times faster than in the P-POD case. This significant computational gain has been obtained by properly selecting a subset of monomials of the P-POD model. The impact of this gain on the simulation time is also remarkable. The simulation of P-POD-RS requires about 3.6 times less time than the simulation of the POD model. In addition, the difference in their responses is very small. The largest difference occurs during Test 1, and this difference is only of 0.13°C.

For the Neural-POD case, the derivatives are calculated around 9.7 times faster than in the POD model. This considerable gain has been achieved by approximating the vector function $\mathbf{f}(\mathbf{a}(t))$ with a multi-layer perceptron. The impact of this gain on the simulation time is such that the simulation of the Neural-POD model requires about 3.3 times less time than the simulation of the POD model. Also notice that the difference between the

POD model and the Neural-POD model is very small. The largest difference in their responses arises in the course of Test 1, and this difference is merely of 0.26°C.

According to the results presented in Table 5.2, the P-POD-RS and Neural-POD models are in that order the fastest reduced-order models of the bar. Although the MPE-POD model was faster than the POD model, it was neither as fast as the P-POD-RS, Neural-POD and P-POD models nor as accurate as them. At this point it is important to emphasize that the MPE method is a general technique that does not exploit the particular characteristics of the kind of problems treated in this chapter, and therefore the results obtained were expected. It is undoubted that the incorporation of prior knowledge about a problem on a given algorithm improves the results.

5.8 Conclusions

In this chapter we have presented two methods for speeding up the evaluation of nonlinear POD models. In the first method, the nonlinear vector function of the POD models is approximated by a multi-layer perceptron which in general can be evaluated much faster than the original vector function. In this approach both the computational gain and the capability of learning accurately complex nonlinear mappings are limited by the size of the network. The larger the MLP, the lesser is the computational gain but the better is the ability of learning complex nonlinear mappings and vice versa. Therefore, the size of the net has to be chosen in such a way that it provides a good trade-off between accuracy and computational gain. One of the main difficulties of this approach is to find a suitable number of hidden neurons for the MLP. Additionally, since the neural networks suffer of the local minima problem, the training of the MLP has to be performed several times starting from different initial conditions in order to make sure that the network is not trapped in a local minimum. However, the convergence to the global minimum can not be guaranteed. This problem gets worse as the number of neurons with nonlinear activation functions increases. Another aspect that we have to bear in mind, is that the approximation of the vector function offered by the neural network is only valid within a specific range. Beyond this range the approximation fails, and the Neural-POD model is not reliable anymore.

The second method is mostly aimed for accelerating nonlinear POD

models derived from input-affine high-dimensional systems with polynomial nonlinearities. We have shown that by exploiting their polynomial nature, we can generate compact and efficient representations (P-POD models) which can be evaluated much faster than the POD models whose vector functions require the full spatial information of the original model. In addition, we have discussed how the sequential feature selection algorithms can help us to accelerate even more these P-POD models by selecting their most relevant monomials (suboptimal solution). Note that this approach might be also employed in the case of input-affine high-dimensional systems with non-polynomial nonlinearities, as long as these nonlinearities can be approximated by low degree polynomials. This condition is advised in order to keep the number of monomials as small as possible. A notable advantage of this approach with respect to the first method is that we have to solve a convex optimization problem. However its drawback is that the number of monomials can be very large.

Furthermore in this chapter, we have discussed two ways of guaranteeing the local stability of POD models with polynomial nonlinearities. In our first approach we end up with an SDP optimization problem whose constraint provides a sufficient condition for the local stability of the models. The problem of this method is that the constraint can be very conservative. In the second approach we finish up with an NSDP optimization problem whose constraints unlike the first approach, give necessary and sufficient conditions for the stability of the POD models. The drawback of this formulation is that we have to deal with a non-convex optimization problem. In both methods the local stability of the POD models is guaranteed independently of the data used. However the quality of the reduced-order models still depends on the quality of data employed in the model reduction process.

Among the POD models derived for the bar, the simulation results showed that the largest computational gains were obtained with the P-POD-RS and Neural-POD models. On one hand we have to point out that the success of the polynomial POD models resides in the fact of taking advantage of the polynomial nature of the problem. On the other hand in the neural network approach, the nature of the problem is not exploited and consequently given its general character, this method can be applied to a wider class of systems, although with the difficulties mentioned before.

Further research is necessary in order to evaluate the approaches proposed in this chapter on dynamic systems with harder nonlinearities.

Chapter 6

General Conclusions

6.1 Concluding remarks

This dissertation considers two main research subjects. First, it describes the application of a set of techniques such as proper orthogonal decomposition, Galerkin projection, model predictive control, Kalman filtering, and sum of squares decomposition (from theory of positive polynomials), to the design of advanced control schemes for tubular chemical reactors. Second, it discusses how to improve the performance in model simulation of reduced-order models derived by using proper orthogonal decomposition and Galerkin projection from nonlinear high-dimensional systems, which are obtained after discretizing the nonlinear partial differential equations that model many processes.

Regarding the first research topic of this thesis, we have the following concluding remarks:

- The design of the predictive control schemes for the tubular reactor treated in this dissertation has been possible due to the significant model order reduction (from 600 states to only 20 states) achieved by means of the POD and Galerkin projection techniques.

- In spite of the linearization and discretization of the nonlinear PDEs that model the reactor, and the dramatic model order reduction obtained using POD for generating the model on which the control

systems proposed are based, the control schemes performed very well when typical disturbances were applied to the reactor (Tests 1 and 2 defined in Section 3.5). However, the performance of the control schemes was compromised when the reactor was subjected to large disturbances (Tests 3 and 4 described in Section 3.5). The degradation in their performance is mainly due to the fact that under large disturbances, the reactor operates far from the operating profiles where the nonlinear model was linearized, and therefore the differences between the nonlinear dynamics of the process and the linear POD model on which the control schemes are based become considerable.

- Typically, POD-based predictive controllers are formulated in terms of POD coefficients which have no physical meaning, and the control of the variables of interest is achieved indirectly by controlling these coefficients. However, we have shown that formulations in terms of physical variables are suitable alternatives that make the tuning of the predictive controllers more intuitive and the definition of their control goals more flexible.

- In Chapter 4, we have shown that the theory of positive polynomials provides a way of reducing the large number of state/output constraints (linear inequalities) that usually characterizes the POD-based predictive controllers. In this approach the feasible region delimited by the large set of inequality constraints is approximated by the one delimited by a few linear matrix inequalities. It has been shown for the case of the reactor that this methodology leads to a significant reduction in the number of temperature constraints which conduces to a considerable saving of memory. Nevertheless the computational time needed for solving the optimization of the predictive controller based on the LMI constraints, is much larger than the time required for solving the original problem. The limitation comes from the fact that the LMI constraints introduce a large number of variables. It is clear that with this positive polynomial approach the resulting optimization problem is more complex than the original one. However we want to emphasize that this technique guarantees the fulfillment of the temperature constraint at every point of the reactor.

- For reducing the large number of state/output constraints characterizing the POD-based predictive controllers, we also proposed in Chapter 4 a greedy algorithm which selects a reduced set of constraints from the full set, by exploiting the similarities between the coefficients of

consecutive constraints. This algorithm proved to be quite effective at the moment of being applied to one of the predictive controllers designed for the reactor. The number of temperature constraints was reduced from 24 000 to only 1600, leading to a considerable saving of memory, and a substantial reduction in the computational time required for solving the optimization within the MPC controller. In addition, the behavior of the controller based on the reduced set of constraints was practically identical to the one based on the complete set. Note however that unlike the positive polynomial approach, we do not have any command on the temperature between the discretization points.

Finally, we have the subsequent concluding remarks concerning the second research subject of this dissertation:

- The use of a multilayer perceptron for approximating the nonlinear vector function of the POD models, has demonstrated to be a possible way of speeding up their evaluation. However, the local minima problem in the training process of the network, can be a limiting factor of the applicability of this approach in systems with a considerable amount of POD coefficients.
- We have shown that the acceleration of POD models derived from input-affine high-dimensional systems with polynomial nonlinearities, can be achieved by taking advantage of the polynomial nature of the models for generating compact and efficient formulations (polynomial POD models). In addition, the use of the sequential feature selection algorithms has proven to be quite useful for boosting the computational gain of these polynomial POD models. Although this approach is mostly intended for systems with polynomial nonlinearities, it might be applied to models with non-polynomial nonlinearities provided that these nonlinearities can be approximated well enough by low degree polynomials. In addition, it has been shown that by using the Lyapunov's theory, the local stability of the resulting POD models with polynomial nonlinearities can be guaranteed.

6.2 Future research

The research presented in this work can be further extended in several directions.

- Along Chapters 3 and 4, it has been pointed out that the tuning of the MPC controllers for the tubular reactor considered in this thesis, has been done in a conservative way (for being able to handle large disturbances). In addition, it has been observed how the performance of the control systems is degraded when large disturbances are applied to the process. These issues are the consequence of not taking into account the nonlinear nature of the reactor in the design of the control schemes. Therefore, a natural extension of the predictive control systems proposed in this thesis, is the incorporation of the nonlinear characteristics of the process into the POD model on which they are based.

- It is clear that the quality of a POD model depends on the quality of the data collected during the snapshot experiment, and this fact makes of the construction of the snapshot matrix a key step in the model reduction process by POD. Many people [8, 40, 58, 87] have addressed several issues regarding the snapshots generation, but thus far, there is not a definitive way to decide [13],

  - How many snapshots are necessary for having good information about the system,
  - How long the simulations should be run to generate the snapshots,
  - Which initial conditions should be used, or
  - How to incorporate control information (in the case of control problems).

  Consequently the design of the snapshot experiment is still an open problem that demands the attention of the researchers.

- Along this work we have not considered the dispersion/diffusion phenomena and the dynamics of the heat exchangers in the jackets of the reactor. The inclusion of these characteristics into the reactor model would make it more realistic, and still not so specific that it compromises the generality of the results and conceptual contributions of the study. Therefore an extension of the presented

work would be the application and pertinent adjustment of the control strategies and algorithms proposed thus far, to the case where the dispersion/diffusion phenomena and the dynamics of the heat exchangers are considered in the reactor model.

- In this dissertation we have addressed the problem of speeding up the evaluation of nonlinear POD models by using neural networks and by exploiting the polynomial nature of some systems. In spite of the good results obtained, we can not say that we have found a definitive solution to this problem, and consequently more research is needed in this direction.

- For the case when POD is applied to multidimensional systems, that is, systems with more than one spatial coordinate, the columns of the snapshot matrix are typically assembled by stacking the values of the physical variable at every grid point. This stacking procedure ignores any possible Cartesian structure that may be present, and commonly produces a snapshot matrix with a large number of rows. If the spatial domain has d coordinates and each of them has N grid points, the number of rows of the snapshot matrix is equal to N^d. It is clear that the number of rows is typically an exponential function of d, and this fact may make the computation of the basis vectors problematic when the number of grid points increases. In order to tackle this situation, in [148, 153], an alternative spectral expansion to the one used in this thesis (and in most of the studies regarding POD) is proposed, which takes into account the multidimensional nature of the spatial coordinates by using a tensor representation. In this approach the basis vectors are computed using the Higher-Order Singular Value Decomposition (HOSVD) [88]. Alternatively in [149,150], a method known as tensor SVD is introduced for computing the basis vectors. However, it should be clear that there is not a straightforward generalization of the algebraic concept of singular values and singular value decompositions to tensors or multi-way arrays. Therefore more research regarding the derivation of optimal basis vectors for multidimensional systems is required.

Bibliography

[1] O. M. Agudelo, M. Baes, J. J. Espinosa, M. Diehl, and B. De Moor. Positive polynomial constraints for POD–based model predictive controllers. *IEEE Transactions on Automatic Control*, 54(5):988–999, May 2009.

[2] O. M. Agudelo, J. J. Espinosa, and B. De Moor. Application of POD and predictive control techniques to the control of the temperature profile of an one–dimensional bar. In *Proceedings of the XII Latin–American Congress on Automatic Control (CLCA 2006)*, pages 127–132, Salvador (Bahia), Brasil, October 2006.

[3] O. M. Agudelo, J. J. Espinosa, and B. De Moor. Control of a tubular chemical reactor by means of POD and predictive control techniques. In *Proceedings of the European Control Conference 2007 (ECC'07)*, pages 1046–1053, Kos, Greece, July 2007.

[4] O. M. Agudelo, J. J. Espinosa, and B. De Moor. POD–based predictive controller with temperature constraints for a tubular reactor. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, pages 3537–3542, New Orleans, USA, December 2007.

[5] O. M. Agudelo, J. J. Espinosa, and B. De Moor. Algorithm for reducing the number of constraints of POD–based predictive controllers. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC 2008)*, pages 4743–4748, Cancun, Mexico, December 2008.

[6] O. M. Agudelo, J. J. Espinosa, and B. DeMoor. Acceleration of POD models with polynomial nonlinearities. *Internal Report 09-81, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2009. Submitted to IEEE Transactions on Control Systems Technology.*

[7] O. M. Agudelo, J. J. Espinosa, and B. DeMoor. Acceleration of nonlinear POD models: a neural network approach. In *Proceedings of the European Control Conference 2009 (ECC'09)*, pages 1547–1552, Budapest, Hungary, August 2009.

[8] H. Aling, S. Banerjee, A. K. Bangia, V. Cole, J. Ebert, A. Emami-Naeini, K. F. Jensen, I. G. Kevrekidis, and S. Shvartsman. Nonlinear model reduction

for simulation and control of rapid thermal processing. In *Proceedings of the American Control Conference*, volume 4, pages 2233–2238, Albuquerque, New Mexico, 1997.

[9] A. C. Antoulas. *Approximation of Large-Scale Dynamical systems– Advances in Design and Control*. SIAM, Philadelphia, PA, USA, 2005.

[10] P. Astrid. Fast reduced order modeling technique for large scale LTV systems. In *Proceedings of American Control Conference 2004*, volume 1, pages 762–767, 2004.

[11] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Technishche Universiteit Eindhoven, Eindhoven (Netherlands), November 2004.

[12] P. Astrid, S. Weiland, K. Willcox, and T. Backs. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, November 2008.

[13] J. A. Atwell. *Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (USA), April 2000.

[14] J. A. Atwell and B. B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and Computer Modelling*, 33:1–19, 2001.

[15] J. A. Atwell and B. B. King. Reduced order controllers for spatially distributed systems via proper orthogonal decomposition. *SIAM J. Sci. Comput.*, 26(1):128–151, 2005.

[16] Anil K. Bangia, Paul F. Batcho, Ioannis G. Kevrekidis, and George Em. Karniadakis. Unsteady two–dimensional flows in complex geometries: Comparative bifurcation studies with global eigenfunction expansions. *SIAM J. Sci. Comput.*, 18(3):775–805, 1997.

[17] H. T. Banks, R. C. H. del Rosario, and R. C. Smith. Reduced–order model feedback control design: Numerical implementation in a thin shell model. *IEEE Transactions on Automatic Control*, 45(7):1312–1324, July 2000.

[18] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM*, 43(6):1002–1045, 1996.

[19] E. Beltrami. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11:98–106, 1873.

[20] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, USA, 2001.

[21] A. Ben-Tal and M. Zibulevsky. Penalty–barrier multiplier methods for convex programming problems. *SIAM J. Optim.*, 7:347–366, 1997.

[22] G. Berkooz. Observations on the proper orthogonal decomposition. In T. B. Gatski, S. Sarkar, and C. G. Speziale, editors, *Studies in Turbulence*, pages 229–247, New York, 1991. Springer–Verlag.

[23] G. Berkooz, P. J. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.*, 25:539–575, 1993.

[24] P. Beyer, S. Benkadda, and X. Garbet. Proper orthogonal decomposition and Galerkin projection for a three–dimensional plasma dynamical system. *Physical Review E*, 61(1):813–823, Jan 2000.

[25] A.M Blanco and J.A. Bandoni. Optimal design of stable process. *Lat. Am. Appl. Res.*, 33(2):123–128, 2003.

[26] L. G. Bleris and M. V. Kothare. Reduced order distributed boundary control of thermal transients in microsystems. *IEEE Transactions on control systems technology*, 13(6):853–867, November 2005.

[27] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Sringer, 1998.

[28] N. K. Bose. *Applied multidimensional systems theory*. Van Nostrand Reinhold, 1982.

[29] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, United Kingdom, 2004.

[30] A. Brandolin, M. Lacunza, P. Ugrin, and N. Capiati. High pressure polymerization of ethylene. an improved mathematical model for industrial tubular reactors. *Polymer Reaction Engineering*, 4:193–241, 1996.

[31] D. H. Chambers, R. J. Adrian, P. Moin, D. S. Stewart, and H. J. Sung. Karhunen–loève expansion of burgers' model of turbulence. *Phys. Fluids*, 31(9):2573–2582, September 1988.

[32] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, April 2000.

[33] P. D. Christofides. Robust control of parabolic PDE systems. *Chemical Engineering Science*, 53(16):2949 – 2965, 1998.

[34] P. D. Christofides. *Nonlinear and Robust Control of PDE Systems. Methods and Applications to Transport–Reaction Processes*. Systems & Control: Foundations & Applications. Birkhäuser, 2001.

[35] P. D. Christofides and P. Daoutidis. Feedback control of hyperbolic PDE systems. *AIChE Journal*, 42:3063 – 3086, 1996.

[36] P. D. Christofides and P. Daoutidis. Robust control of hyperbolic PDE systems. *Chemical Engineering Science*, 53(1):85 – 105, 1998.

[37] J. P. Cusumano, M. T. Sharkady, and B. W. Kimble. Experimental measurements of dimensionality and spatial coherence in the dynamics of a flexible-beam impact oscillator. *Philosophical Transactions: Physical Sciences and Engineering*, 347(1683):421–438, May 1994.

[38] C. R. Cutler and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. In *AIChE National Meeting*, Houston, TX, 1979.

[39] P. Danckwertz. Continuous flow systems. *Chemical Engineering Science*, 2:1–13, 1953.

[40] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag. Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders. *Phys. Fluids A*, 3(10):2337–2354, October 1991.

[41] D. Del Vecchio and N. Petit. Boundary control for an industrial under–actuated tubular chemical reactor. *Journal of Process Control*, 15:771–784, 2005.

[42] M. Diehl, F. Jarre, and C. Vogelbusch. Loss of superlinear convergence for an SQP–type method with conic constraints. *SIAM Journal on Optimization*, 16(4):1201–1210, 2006.

[43] J. R. Dormand and P. J. Prince. A family of embedded runge-kutta formulae. *J. Comp. Appl. Math.*, 6:19–26, 1980.

[44] S. Dubljevic and P. D. Christofides. Predictive output feedback control of parabolic partial differential equations (PDEs). *Industrial & Engineering Chemistry Research*, 45(25):8421–8429, 2006.

[45] S. Dubljevic and P. D. Christofides. Predictive output feedback control of parabolic PDEs. In *Proceedings of the American Control Conference 2006*, pages 1155–1160, Minneapolis, Minnesota, USA, June 2006.

[46] S. Dubljevic, P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of transport-reaction processes. *Computers and Chemical Engineering*, 29(11–12):2335–2345, 2005.

[47] M. Fahl. Computation of POD basis functions for fluid flows with lanczos methods. *Math. Comput. Modeling*, 34:91–1078, 2001.

[48] M. Fahl and E. W. Sachs. Reduced order modelling approaches to PDE–constrained optimization based on proper orthogonal decomposition. In L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, editors, *Large–Scale PDE–Constrained Optimization*, pages 268–280, Heidelberg, Germany, 2003. Springer–Verlag.

[49] F. Fang, C. C. Pain, I. M. Navon, G. J. Gorman, M. D. Piggott, P. A. Allison, P.E. Farrell, and A. J. H. Goddard. A POD reduced order unstructured mesh ocean modelling method for moderate reynolds number flows. *Ocean Modelling*, 28(1–3):127–136, 2009.

[50] B. Fares, D. Noll, and P. Apkarian. Robust control via sequential semidefinite programming. *SIAM Journal on Control and Optimization*, 40(6):1791–1820, 2002.

[51] B. F. Feeny and R. Kappagantu. On the physical interpretation of proper orthogonal modes in vibrations. *Journal of Sound and Vibration*, 211(4):607–616, 1998.

[52] R. Quinta Ferreira, C. Costa, and S. Masetti. Reverse-flow reactor for a selective oxidation process. *Chemical Engineering Science*, 54:4615–4627, 1999.

[53] M. Fjeld and B. Ursin. Approximate lumped models of a tubular chemical reactor, and their use in feedback and feedforward control. In *Proceedings of the 2nd IFAC Symposium on Multivariable Technical Control Systems*, pages 1–18, North Holland, Amsterdam, 1971.

[54] G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, second edition, 1990.

[55] R. W. Freund and F. Jarre. A sensitivity analysis and a convergence result for a sequential semidefinite programming method. Technical report, Bell Laboratories, Murray Hill, New Jersey, USA, 2003.

[56] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice-a survey. *Automatica*, 25(3):335–348, 1989.

[57] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, UK, 1981.

[58] M. D. Graham and I. G. Kevrekidis. Alternative approaches to the karhunen–loève decomposition for model reduction and data analysis. *Computers Chem. Eng.*, 20(5):495–506, 1996.

[59] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low–order models. Part I–open–loop model development. *International Journal for Numerical Methods in Engineering*, 44(7):945–972, 1999.

[60] M. M. Gupta, L. Jin, and N. Homma. *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. Wiley–IEEE press, April 2003.

[61] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[62] M. T. Hagan and M. Menhaj. Training feed–forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

[63] S. Haykin. *Neural Networks, a comprehensive foundation*. Prentice Hall, Upper Saddle River, New jersey, 1999.

[64] M. Hazenberg, P. Astrid, and S. Weiland. Low order modeling and optimal control design of a heated plate. In *Proceedings of the 5th European Control Conference*, Cambridge, September 2003.

[65] D. Henrion, J. Löfberg, Michal Kočvara, and Michael Stingl. Solving polynomial static output feedback problems with PENBMI. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 7581–7586, Seville, Spain, December 2005.

[66] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherence Structure, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 1996.

[67] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[68] M. Hovd and R. D. Braatz. Handling state and output constraints in MPC using time–dependent weights. In *Proceedings of the American Control Conference (ACC)*, Arlington, USA, 2001.

[69] M. Hovd and R. D. Braatz. On the use of soft constraints in MPC controllers for plants with inverse response. In *Proceedings of the 6th IFAC symposium on Dynamics and control of process Systems*, Jejudo Island, Korea, 2001.

[70] L. Huisman. *Control of glass melting processes based on reduced CFD models*. PhD thesis, Technishche Universiteit Eindhoven, Eindhoven (Netherlands), March 2005.

[71] L. Huisman and S. Weiland. Identification and model predictive control of an industrial glass feeder. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID–2003)*, pages 1685–1689, Rotterdam, The Netherlands, August 2003.

[72] L. Huisman and S. Weiland. Reduced model based LQG control of an industrial glass feeder. In *Proceedings of the 10th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems: Theory and Applications*, volume 1, pages 421–426, Osaka, Japan, July 2004.

[73] A. Iollo, S. Lanteri, and J. A. Désidéri. Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, March 2000.

[74] I. T. Jollife. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[75] C. Jordan. Mémoire sur les formes bilinéaires. *J. Math. Pures Appl., Deuxième Série*, 19:35–54, 1874.

[76] C. Jordan. Sur la réduction des formes bilinéaires. *Comptes Rendus de l'Academie Sciences, Paris*, 78:614–617, 1874.

[77] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions ASME, Series D, Journal of basic engineering*, 82:34–45, March 1960.

[78] G. M. Kepler, H. T. Tran, and H. T. Banks. Reduced order model compensator control of species transport in a CVD reactor. *Optimal Control Application & Methods*, 21(4):143–160, 2000.

[79] M. Kirby, J. P. Boris, and L. Sirovich. A proper orthogonal decomposition of a simulated supersonic shear layer. *Int. J. Numerical Methods Fluids*, 10:411–428, 1990.

[80] M. Kirby and L. Sirovich. Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[81] R. Kohavi and H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324, 1997.

[82] D. D. Kosambi. Statistics in function space. *J. Indian Math. Soc.*, 7:76–88, 1943.

[83] M. Kočvara and M. Stingl. PENNON – a code for convex nonlinear and semidefinite programming. *Opt. Methods and Software*, 20(3):317–333, 2003.

[84] E. Kreuzer and O. Kust. Proper orthogonal decompositionan efficient means of controlling self-excited vibrations of long torsional strings. In *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, pages 105–110, 1996.

[85] K. Kunisch and S. Volkwein. Control of the burgers equation by a reduced–order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102(2):345–371, August 1999.

[86] H. Kwakernaak and R. Sivan. *Linear optimal control systems*. Wiley-Interscience, 1972.

[87] S. Lall, J. E. Marsden, and S. Glavaški. Empirical model reduction of controlled nonlinear systems. Technical Report CIT-CDS-98-008, Control and Dynamical Systems, California Institute of Technology, June 1998.

[88] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[89] J. H. Lee and B. Cooley. Recent advances in model predictive control and other related areas. In J. C. Kantor, C. E. Garcia, and B. Carnahan, editors, *Proceedings of the Fifth international conference on chemical process control AICHE and CACHE*, pages 201–216, 1997.

[90] A. Liakopoulos, P. A. Blythe, and H. Gunes. A reduced dynamical model of convective flows in tall laterally heated cavities. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 453(1958):663–672, 1997.

[91] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applications–Part I: Theory. *Journal of Sound and Vibration*, 252(3):527–544, May 2002.

[92] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applications–Part II: Model reduction for MEMS dynamical systems. *Journal of Sound and Vibration*, 256(3):515–532, 2002.

[93] W. Z. Lin, Y. J. Zhang, and E. P. Li. Proper orthogonal decomposition in the generation of reduced order models for interconnects. *IEEE Transactions on Advanced Packing*, 31(3):627–636, 2008.

[94] M. Loève. *Probability Theory*. D. Van Nostrand Company, Inc, New Jersey, second edition, 1960.

[95] F. Logist. *Model based optimization and control of chemical reactors with distributed parameters*. PhD thesis, Katholieke Universiteit Leuven, Leuven (Belgium), January 2008.

[96] D. G. Luenberger. An introduction to observers. *IEEE transactions on Automatic Control*, 16(6):596–602, December 1971.

[97] J. Lumley and P. Blossey. Control of turbulence. *Annual Review of Fluid Mechanics*, 30:311–327, 1998.

[98] J. L. Lumley. *Stochastic Tools in Turbulence*. Academic Press, New York, 1970.

[99] H. V. Ly and H. T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Journal of Mathematical and Computer Modeling*, 33:223–236, 2001.

[100] H. V. Ly and H. T. Tran. Proper orthogonal decomposition for flow calculations and optimal control in a horizontal CVD reactor. *Quarterly of Applied Mathematics*, 60:631–656, 2002.

[101] Mathworks, USA. *Optimization Toolbox 3: user's guide*, 2006.

[102] Mathworks, USA. *Neural Network Toolbox 6. User's guide*, 2008.

[103] D. Q. Mayne. Nonlinear model predictive control: An assessment. In J. C. Kantor, C. E. Garcia, and B. Carnahan, editors, *Proceedings of the Fifth international conference on chemical process control AICHE and CACHE*, pages 217–231, 1997.

[104] D. Q. Mayne, J. B. Rawlings C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[105] A. Miller. *Subset Selection in Regression. Monographs on Statistics and Applied Probability 95.* Chapman & Hall/CRC, Boca Raton, FL, second edition, 2002.

[106] B. Mishra. *Algorithmic Algebra.* Sringer-verlag, 1999.

[107] M. Morari and J. H. Lee. Model predictive control: The good, the bad, and the ugly. In Y. Arkun and W. H. Ray, editors, *Chemical Process Control (CPC IV). Forth International Conference on Chemical Process Control*, pages 419–444, Amsterdan, 1991. Elsevier.

[108] K. G. Murty and S. N. Kabadi. Some NP–complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[109] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[110] T. Nakamura, S. Takamura, and M. Asada. Behaviour–based map representation for a sonar–based mobile robot by statistical methods. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96 (IROS 96)*, volume 1, pages 276–283, Osaka, Japan, Nov. 1996.

[111] Y. Nesterov. Squared functional systems and optimization problems. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 405–440. Kluwer Academic Publishers, 2000.

[112] Y. Nesterov and M. Todd. Self–scaled barriers and interior–point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[113] P. Van Overschee and B. De Moor. *Subspace Identification of Linear Systems: Theory, Implementation, Applications.* Kluwer Academic Publishers, 1996.

[114] G. Pannocchia and J. B. Rawlings. Robustness of MPC and disturbance models for multivariable ill–conditioned processes. Technical Report 2001–02, University of Wisconsin–Madison, Madison, WI 53706 (U.S.A), May 2001.

[115] A. Papachristodoulou and S. Prajna. A tutorial on sum of squares techniques for systems analysis. In *Proceedings of the American Control Conference 2005*, volume 4, pages 2686–2700, Portland, Oregon, USA, June 2005.

[116] T. W. Parks and C. S. Burrus. *Digital Filter Design.* John Wiley & Sons, New York, USA, 1987.

[117] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, USA, 2000.

[118] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[119] S. Prajna. POD model reduction with stability guarantee. In *Proceedings of 42nd IEEE Conference on Decision and Control (CDC 2003)*, Hawaii, USA, December 2003.

[120] S. J. Qina and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[121] S. S. Ravindran. A reduced–order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, November 2000.

[122] J. B. Rawlings and J. G. Ekerdt. *Chemical Reactor Analysis and Design Fundamentals*. Nob Hill Publishing, Madison, Wisconsin, 2002.

[123] J. B. Rawlings, E. S. Meadows, and K. Muske. Nonlinear model predictive control: a tutorial and survey. In *Proceedings of the IFAC Symposium on Advanced Control of Chemical Processes*, pages 203–214, Kyoto, Japan, 1994.

[124] Bruce Reznick. Some concrete aspects of Hilbert's 17th problem. In *In Contemporary Mathematics*, pages 251–272. American Mathematical Society, 2000.

[125] J. Richalet, A. Rault, J. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

[126] N. L. Ricker. Model predictive control: State of the art. In Y. Arkun and W. H. Ray, editors, *Chemical Process Control (CPC IV). Forth International Conference on Chemical Process Control*, pages 271–296, Amsterdan, 1991. Elsevier.

[127] U. T. Ringertz. Eigenvalues in optimal structural design. In A.R Conn, L. T. Biegler, T. F. Coleman, and F. Santosa, editors, *Proceedings of an IMA Workshop on Large–Scale Optimizatioin*, pages 135–149, 1997.

[128] J. A. Rossiter. *Model-Based Predictive Control. A practical Approach*. CRC Press, Boca Raton, Florida, USA, 2003.

[129] R. Ruotolo and C. Surace. Using SVD to detect damage detection in structures with different operational conditions. *Journal of Sound and Vibration*, 226(3):425–439, 1999.

[130] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. I. Teil. Entwicklung willkürlicher funktionen nach system vorgeschriebener. *Math. Ann.*, 63:433–476, 1907.

[131] P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45:1649–1659, 1999.

[132] L. F. Shampine and M. E. Hosea. Analysis and implementation of TR–BDF2. *Applied Numerical Mathematics*, 20:21–37, 1996.

[133] H. Shang, J. Forbes, and M. Guay. Model predictive control for quasilinear hyperbolic distributed parameter systems. *Industrial and Engineering Chemistry Research*, 43(16):2140–2149, 2004.

[134] L. Sirovich. Turbulence and the dynamics of coherent structures. Part I: coherent structures. Part II: symmetries and transformations. Part III: dynamics and scaling. *Quart. Appl. Math.*, 45:561–590, 1987.

[135] L. Sirovich and M. Kirby. Low–dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, 1987.

[136] L. Sirovich, B. W. Knight, and J. D. Rodriguez. Optimal low-dimensional dynamical approximations. *Quarterly of Applied Mathematics*, 48(3):535–548, September 1990.

[137] I. Y. Smets, D. Dochain, and J. F. Van Impe. Optimal temperature control of a steady–state exothermic plug–flow reactor. *AIChE Journal*, 48 (2):279–286, 2002.

[138] I. Y. Smets and J. F. Van Impe. Optimal control of tubular chemical reactors: performance assessment under transient and diffuse conditions. In *Proceedings of the 10th Mediterranean Conference on Control and Automation – MED2002*, Lisbon, Portugal, 2002.

[139] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, December 1993.

[140] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[141] J. J. Sylvester. A new proof that a general quadric may be reduced to its canonical form (that is, a linear function of squares) by means of a real orthogonal substitution. *Messenger of Mathematics*, 19:1–5, 1889.

[142] J. J. Sylvester. On the reduction of a bilinear quantic of the nth order to the form of a sum of n products by a double orthogonal substitution. *Messenger of Mathematics*, 19:42–46, 1889.

[143] J. J. Sylvester. Sur la réduction biorthogonale d'une forme linéo-linéaire à sa forme cannonique. *Comptes Rendus de l'Academie Sciences, Paris*, 108:651–653, 1889.

[144] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Series in Computational and Physycal Processes in Mechanics and Thermal Sciences. Taylor & Francis, USA, second edition, 1997.

[145] P. Tatjewski and M. Ławryńczuk. Soft computing in model–based predictive control. *Int. J. Appl. Math. Comput. Sci.*, 16(1):7–26, 2006.

[146] A. Theodoropoulou, R. A Adomaitis, and E. Zafiriou. Model reduction for optimization of rapid thermal chemical vapor deposition systems. *IEEE Transactions on Semiconductor Manufacturing*, 11(1):85–98, February 1998.

[147] E.F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics. A practical Introduction*. Springer–Verlag, Heidelberg, Germany, second edition, 1999.

[148] F. van Belzen. Model reduction through proper orthogonal decompositions for multidimensional systems. In *Book of abstracts of the 26th Benelux meeting on Systems and Control*, page 152, Lommel, Belgium, March 2007.

[149] F. van Belzen. Analysis and approximation of multi-way arrays through singular value decompositions. In *Book of abstracts of the 27th Benelux meeting on Systems and Control*, page 103, Heeze, The Netherlands, March 2008.

[150] F. van Belzen, S. Weiland, and J. de Graaf. Singular value decompositions and low rank approximations of multi-linear functionals. In *Proceedings of the 46th IEEE conference on Decision and Control*, pages 3751–3756, New Orleans, USA, December 2007.

[151] S. Volkwein and A. Hepberger. Impedance identification by POD model reduction techniques (impedanz-identifikation mittels POD modellreduktion). *at - Automatisierungstechnik*, 56(8):437–446, August 2008.

[152] A. Webb. *Statistical Pattern Recognition*. John Wiley & Sons, LTD, second edition, July 2002.

[153] S. Weiland and F. van Belzen. Model reduction through multilinear singular value decompositions. In *Proceedings of the European Conference on Computational Dynamics ECCOMAS CFD 2006*, Egmond aan Zee, The Netherlands, September 2006.

[154] H. Weyl. Das asymptotische verteilungsgesetz der eigenwert linearer partieller differentialgleichungen (mit einer anwendung auf der theorie der hohlraumstrahlung). *Math. Ann.*, 71:441–479, 1912.

[155] W. Y. Yang, W. C., T. S. Chung, and J. Morris. *Applied Numerical Methods using Matlab*. Wiley–Interscience, 2005.

[156] A. Yousefi, B. Lohmann, J. Lienemann, and J. G. Korvink. Nonlinear heat tranfer modelling and reduction. In *Proceedings of the 12th IEEE Mediterranean Conference on Control and Automation (MED '04)*, Kusadasi, Turkey, June 2004.

Curriculum Vitae

Oscar Mauricio Agudelo Mañozca was born in Cali, Colombia, in January 1974. In 1997, he received the degree of Electronics Engineer from the Univerisidad Autónoma de Occidente, Cali, Colombia, and in 2004, he received the master degree in Industrial Control Engineering from the Universidad de Ibagué in cooperation with the Katholieke Universiteit Leuven and the Universiteit Gent, Ibagué, Colombia. His master thesis was entitled "Control of a helicopter laboratory process using linear and fuzzy techniques". From 1997 to 2004, he worked at the Universidad Autónoma de Occidente as a full time teacher of control and automation. In October 2004, he moved to Leuven in order to start his predoctoctal stage in the Katholieke Universiteit Leuven. During this period he worked alongside three companies, IPCOS (from Belgium), Protomation (from Netherlands) and Stamicarbon (from Netherlands) in a research project entitled "Control of the synthesis section of a urea plant using MPC control techniques". In 2005, he started the Ph.D program in the SCD/SISTA research division of the Electrical Engineering Department of the K.U.Leuven, under the supervision of Prof. Dr. Ir. Bart De Moor, Prof. Dr. Ir. Jairo Espinosa and Prof. Dr. Ir. Joos Vandewalle.

Publications by the author

Journal Papers

- O. M. Agudelo, M. Baes, J. J. Espinosa, M. Diehl and B. De Moor. Positive polynomial constraints for POD-based model predictive controllers. *IEEE Transactions on Automatic Control*, 54(5):988–999, May 2009.
- O. M. Agudelo, J. J. Espinosa and J. Vandewalle. Control of a helicopter laboratory process using fuzzy techniques. Internal Report 06-74, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006. Accepted for publication in *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*.
- O. M. Agudelo, J. J. Espinosa and B. De Moor. Acceleration of POD models with polynomial nonlinearities. Internal Report 09-81, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2009. Submitted to *IEEE Transactions on Control Systems Technology*.

International Conference Papers

- O. M. Agudelo, J. J. Espinosa and J. Vandewalle. Control of the synthesis section of a urea plant by means of an MPC controller. In W. Marquardt, and C. Pantelides, editors, *Proceedings of the 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, volume 21 of *Computer-Aided Chemical Engineering*, pages 1305–1310. Elsevier, 2006.
- O. M. Agudelo, J. J. Espinosa and B. De Moor. Application of POD and predictive control techniques to the control of the temperature profile of an one-dimensional bar. In *Proceedings of the XII Latin-American Congress on Automatic Control (CLCA 2006)*, pages 127–132, Salvador, Brazil, October 2006.

- O. M. Agudelo, J. J. Espinosa and B. De Moor. Control of a tubular chemical reactor by means of POD and predictive control techniques. In *Proceedings of the European Control Conference 2007 (ECC'07)*, pages 1046–1053, Kos, Greece, July 2007.
- O. M. Agudelo, J. J. Espinosa and B. De Moor. POD-based predictive controller with temperature constraints for a tubular reactor. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, pages 3537–3542, New Orleans, USA, December 2007.
- O. M. Agudelo, J. J. Espinosa and B. De Moor. Algorithm for reducing the number of constraints of POD-based predictive controllers. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC 2008)*, pages 4743–4748, Cancun, Mexico, December 2008.
- O. M. Agudelo, J. J. Espinosa and B. De Moor. Acceleration of nonlinear POD models: a neural network approach. In *Proceedings of the European Control Conference 2009 (ECC'09)*, pages 1547–1552, Budapest, Hungary, August 2009.

Published Abstracts

- O. M. Agudelo, J. J. Espinosa and B. De Moor. Control of a tubular reactor using POD and MPC. In *Book of Abstracts of the 26th Benelux Meeting on Systems and Control*, page 171, Lommel, Belgium, March 2007.

Internal Reports

- O. M. Agudelo, J. J. Espinosa and J. Vandewalle. Control of the synthesis section of a urea Plant using MPC control techniques. Internal Report 05-157, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.
- O. M. Agudelo, J. J. Espinosa and J. Vandewalle. Application of predictive control techniques to the synthesis section of a urea plant. Internal Report 05-213, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.