Model Predictive Control of Dissipative Distributed Parameter Systems

by

Liu Liu

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering
University of Alberta

Abstract

Distributed parameter systems (DPSs) are distinguished by the fact that the states, controls, and outputs may depend on spatial position. The certain class of dissipative DPSs includes many underlying chemical and mechanical spatiotemporal phenomena such as chemical reactions, convection and diffusion, flexible structures and certain wave propagation problems, all of which can be described by partial differential equations (PDEs). In the past decade, considerable work has concentrated on the construction of a general framework of reduced-order control synthesis for PDEs systems arising from the modeling of DPSs on the basis of low-order ODEs models which are derived by spectral decomposition schemes. Among those control synthesis, model predictive control (MPC) is a popular and widely used method because of its ability to account for input and state constraints. However, these works did not address completely the problem of state constraints in the predictive controller design for either the PDE systems with non-self-adjoint operators or the PDE system describing flexible structures. Furthermore, almost all the existing MPC designs for DPSs are developed in an implicit form and implemented in an on-line way, which leads to the numerically-determined control actions and relatively large computational effort.

This thesis presents a MPC scheme for the parabolic PDE system where a convective term is included in the operator to describe the convective heat and mass transfer which makes the operator non-self-adjoint as well as a MPC scheme for the flexible structural system described by a fourth-order PDE, and an explicit/multi-parametric MPC scheme for dissipative PDE systems. First, a MPC scheme which accounts for the input and state constraints is proposed for the parabolic PDEs system describing the axial dispersion chemical reactor. Subsequently, an approach is proposed to approximate the infinite-dimensional representation of Euler-Bernoulli beam system by a reduced-order finite-dimensional model, and the proposed MPC scheme is implemented on the reduced-order beam system. Following this, an explicit MPC scheme, which is solved off-line, is proposed to stabilize the certain class of dissipative PDE systems as well as guarantee the input and state constraints.

Acknowledgements

First and foremost, I would like to express my deepest thanks to my two supervisors, Prof. Stevan Dubljevic and Prof. Biao Huang. Their patience, support, and immense knowledge were key motivations through my graduate studies. Without their assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished.

At the beginning I know almost nothing about distributed parameter systems, it was Prof. Dubljevic who taught me step-by-step. I am truly thankful for his selfless help and dedication to both my personal and academic development. I cannot think of a better supervisor to have. Prof. Huang is a guiding beacon, who has been always there to listen and give advice. He has always found the time to propose consistently excellent improvements. I owe a great debt of gratitude to Prof. Dubljevic and Prof. Huang.

I would also like to express my deepest gratitude to my parents for supporting me spiritually throughout my life.

Last but not the least, I am grateful to my fellow friends in computer process control group who supported me a lot: Yu Yang, Mojtaba Izadi, Javad Abdollahi, Azzam Hazim, Xiaodong Xu, Felicia Yapari, Yaojie Lu, Yuan Yuan, Jing Zhang, Zhankun Xi, Tianbo Liu, Shuning Li, and Ming Ma.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Transport-reaction process is a typical representative of a distributed-parameter system as it usually possesses the characteristic of considerable spatial variations caused by the potential presence of convection and diffusion phenomena. First-principle modeling of the transport-reaction process within finite spatial domain usually leads to the system of parabolic PDEs. The major characteristic of parabolic PDEs systems is that their spatial differential operators are featured by a spectrum that can be partitioned into a slow part which includes a finite number of eigenvalues that are close to the imaginary axis and a fast complement which consists of an infinite number of eigenvalues that are located far-left in the complex plane [1]. In addition, the Euler-Bernoulli beam system described by a fourth-order PDE is another representative of the dissipative distributed parameter system. Although the fourth-oder PDE describing the flexible beam system is not parabolic, its spectrum shares the same feature with the parabolic PDE system. In the past decade, there are many existing results on the framework of predictive controller for distributed-parameter systems. However, these results did not address completely the problem of state constraints in the controller design for PDEs systems with non-self-adjoint operators and non-symmetric eigenfunctions. Furthermore, control practitioners will often face the scenarios in which ac-

tuators and sensors have their limits due to physical properties or practical characteristics associated with actuators/sensors implementation, or the system state is required not to be in excess of specified limit values (for example, requiring the concentration of a certain product to be maintained above some desired purity condition). Motivated by this consideration, in this thesis, we present a model predictive control synthesis for the transport-reaction system which has a convection term and for the flexible Euler-Bernoulli beam system which involves a fourth order spatial operator. In addition, motivated by the consideration of overcoming the drawback of regular model predictive control, an explicit/multi-parametric model predictive control formulation is proposed for the certain class of dissipative distributed parameter systems as well.

1.2 Background

Control problem of various classes of distributed parameter systems, arising from the modeling of transport-reaction processes has attracted a lot of attention in the past decade. The traditional approach for control of parabolic PDEs systems utilizes spatial discretization techniques to obtain systems of ordinary differential equations (ODEs), which are subsequently utilized as the foundation of the finite-dimensional controllers design; see [2, 3]. This approach has a significant drawback that the number of states which must be preserved to obtain a system of ODEs in order to yield the required order of approximation, might be quite large, which leads to a high dimensional controller realization and complex controller design. In the past, considerable work has concentrated on the construction of a general framework of reduced-order control synthesis for parabolic PDEs systems and other PDEs systems arising from the modeling of distributed-parameter systems on the basis of low-order ODEs models which are derived by the combination of the concept of inertial

manifolds [4] and spectral method. A number of researchers have explored many problems related to control of a system described by PDE, such as dynamic optimization, output feedback controller design, nonlinear and robust control of the PDE system [5–7]. Besides the above results which are concerned with the order reduction, stabilization, and tracking problem of a parabolic PDE system, notable research has been carried out on the development of methodologies on optimal control for the distributed-parameter systems [1, 8]. Apart from the issue of optimal controller design for distributed-parameter systems with fixed sensors or actuators [9], significant research has been devoted to the optimal location of sensors and actuators [10] and how to switch among actuators under an optimal control policy [11, 12].

Model predictive control (MPC) synthesis is a popular and widely used method in the control of process systems because of its ability to account for manipulated input and state variable constraints. The optimal control sequence over a finite horizon is computed by solving an online open-loop controller performance optimization problem. At each sampling instant, the controller performance is presented by the deviations of input and predicted state from their set points. Only the first control move of the sequence is regarded as the optimal control input and then injected into the system. By repeating the performance optimization problem with new predicted state values at each instant, an online implicit model predictive controller is obtained. In the past, numerous research studies have concentrated on the general framework of model predictive control synthesis for PDE systems describing distributed parameter systems, such as MPC with internal model control structure on the PDE system [13] and characteristic-based MPC on the hyperbolic PDE system [14]. Besides the above MPC frameworks, which are derived with traditional order reduction techniques, the issue of development of model predictive controllers which are constructed on the basis

of reduced-order models has been explored by Dubljevic, Christofides, Alonso within the framework of distributed [15] and boundary applied actuation [16], predictive output and full state feedback control [17], multivariable predictive control of PDE [18], and robust multi-model predictive control [19]. In all of the above results, the models of interest have a feature that the spatial operator is self-adjoint, leading to symmetric spatial operators and eigenfunctions. It is necessary to consider a non-self-adjoint operator arising from the convective term which describes convective heat and mass transfer in a transport-reaction process.

In another line of work, many mechanical systems, such as robots with flexible arms and aircraft appendages can be modeled by distributed parameter systems. In the past, the control problem of flexible systems is mainly focused on the boundary control method and control related issues, such as controllability [20] and boundary control [21] of the wave equation, boundary control of a Euler-Bernoulli beam [22], basis property and stability of Euler-Bernoulli beam [23]. The flexible Euler-Bernoulli beam can be modeled by a fourth-order PDE system which is neither parabolic nor hyperbolic. It is necessary to decompose the fourth-order spatial operator by spectral method and develop a reduced-order predictive controller for the beam system.

Apart from the issues of closed-loop stability, reference tracking, and constraint satisfaction for regular MPC of distributed parameter systems, significant research has been devoted to the modern issues of predictive controller design for distributed parameter systems, such as economic MPC of parabolic PDE systems [24] and piece-wise constant predictive control to satisfy predefined performance criteria [25]. In all of the above results, the MPC laws applied on distributed parameter systems have a feature that the MPC formulations are solved online and the optimal control action values are in an implicit form. Online opti-

mization leads to the major shortcomings that the computational effort for regular MPC is relatively large and we are not aware of any knowledge on the control actions which are determined numerically. The disadvantages of regular MPC can be overcome by explicit MPC [26–30] as the solution to explicit MPC is derived as an explicit function of the state variable with its corresponding critical regions, and the implementation of explicit MPC is then completed in an off-line way. The optimization problem involved in explicit MPC is solved as a set of off-line multi-parametric programming problems in which only the current control variables, current states and constraints are considered [31]. In other words, the single-stage multi-parametric programming problem is decomposed into a set of smaller optimization subproblems by dynamic programming techniques [32] and each optimization subproblem will then be solved by multi-parametric programming techniques [33]. The computational effort is reduced by disassembling the optimization problem into smaller optimization subproblems and by implementing the controller in an off-line way.

1.3 Thesis outline and contributions

This thesis is organized as follows:

In Chapter 2, the algorithm of model predictive control is proposed for the axial dispersion chemical reactor system. We consider a scenario where a convective (first-order partial derivative) term is included in the spatial operator to describe the convective heat and mass transfer, which makes the operator non-self-adjoint. The eigenvalue problem of the operator is then solved in a biorthogonal form with non-symmetric eigenfunctions. In order to explore the best level of approximation for the infinite-dimensional system, different discretization methods are discussed and explored in this chapter. The synthesis of modal model predictive control is formulated to control the temperature and concentration

in the axial dispersion chemical reactor with input and state constraints. The modal model representation of the PDEs system is derived using Galerkin's method and spectral decomposition technique. Two discretization methods are provided to derive different types of discrete-time model representation. In order to obtain the discrete-time model representation, a transfer function of the PDEs system is required. Then the discrete-time model is utilized in the design of a modal model predictive controller. Input and state constraints are accounted for in the discrete-time predictive controller.

In Chapter 3, the algorithm of model predictive control proposed in Chapter 2 is implemented on the flexible Euler-Bernoulli beam system. Since the Euler-Bernoulli equation is neither hyperbolic nor parabolic, the method to solve the eigenvalue problem of the Euler-Bernoulli equation is different with those to solve hyperbolic or parabolic equations. The eigenvalue problem of a fourth-order spatial operator is addressed and the result is subsequently used for decomposing the two-dimensional spatial operator arising from the flexible Euler-Bernoulli beam system with structural damping. After applying Galerkin's method and spectral decomposition technique, the standard discretization method and Tustin's discretization method with Cayley transform are utilized to derive the discrete-time model representations, respectively. The transfer function of the beam system is also obtained to derive Tustin's discrete-time model representation. Two MPC formulations, in which the fast dynamics are only involved in the PDE state constraints equations, are developed on the basis of two discrete-time system representations.

In Chapter 4, the algorithm of explicit/multi-parametric model predictive control is proposed for the certain class of dissipative distributed parameter system described by PDE system. A finite-dimensional modal representation of the infinite-dimensional system is derived through Galerkin's method and modal decomposition techniques. The algorithm

of explicit/multi-parametric MPC is constructed in a way that the objective function is only concerned with low-order model dynamics, while the state constraints account for both low-order and higher-order model dynamics. Finally, performance of the proposed explicit MPC formulation is demonstrated through simulation studies on the transport-reaction system and the flexible Euler-Bernoulli beam system.

Chapter 5 summarizes the main results of this thesis and discusses future research directions.

Chapter 2

Model Predictive Control of Axial Dispersion Chemical Reactor*

2.1 Introduction

This chapter discusses the development of model predictive control algorithm which accounts for the input and state constraints applied to the parabolic partial differential equations (PDEs) system describing the axial dispersion chemical reactor. Spatially varying terms arising from the nonlinear PDEs model are accounted in model development. Finite-dimensional modal representation capturing the dominant dynamics of the PDEs system is derived for controller design through Galerkin's method and modal decomposition technique. Tustin's discretization and Cayley transform are used to obtain infinite-dimensional discrete-time modal dynamic representations which are used in subsequent constrained controller design. The proposed discrete-time constrained model predictive control synthesis is constructed in a way that the objective function is only based on the low-order modal representation of the PDEs system, while higher-order modes are utilized only in the constraints of the PDEs state. Finally, the MPC formulations are successfully applied, via simulation results, to the PDEs system with input and state constraints.

*This chapter is a revised version of "L. Liu, B. Huang, S. Dubljevic, Model predictive control of axial dispersion chemical reactor, Journal of Process Control. 24(11) (2014) 1671-1690."

2.2 Axial dispersion chemical reactor

In this chapter, we consider the axial dispersion chemical reactor system. A first-order exothermic reaction is occurring in a non-adiabatic tubular chemical reactor. Fluid which contains reactant A at temperature T_f and constant concentration C_f flows into the tubular reactor of length L and radius R from the left end of the reactor. A first-order chemical reaction, $A \to B$, is taking place in the reactor, with constant reaction enthalpy ΔH. The reactor is cooled by a cooling jacket in which the coolant is at temperature T_w; see Fig. 2.1.

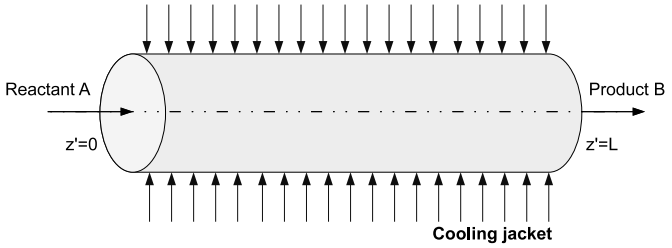


Figure 2.1: Schematic description of the axial dispersion chemical reactor

The first-principle model for such a system is given in [34],

$$\begin{aligned}
\rho_f C_{pf} \frac{\partial T(z',t')}{\partial t'} =& -\rho_f C_{pf} v \frac{\partial T(z',t')}{\partial z'} + k \frac{\partial^2 T(z',t')}{\partial z'^2} + (-\Delta H) k_0 e^{-E_a/R_g T(z',t')} C_A(z',t') \\
& - \frac{2U}{R}\big(T(z',t') - T_w(t')\big) \\
\frac{\partial C_A(z',t')}{\partial t'} =& -v \frac{\partial C_A(z',t')}{\partial z'} + D_A \frac{\partial^2 C_A(z',t')}{\partial z'^2} - k_0 e^{-E_a/R_g T(z',t')} C_A(z',t')
\end{aligned} \tag{2.1}$$

with the following boundary and initial conditions:

$$\begin{aligned}
k \frac{\partial T(0,t')}{\partial z'} &= \rho_f C_{pf} v\big(T(0,t') - T_f(t')\big) \\
\frac{\partial C_A(0,t')}{\partial z'} &= \frac{v}{D_A}\big(C_A(0,t') - C_f\big) \\
\frac{\partial T(L,t')}{\partial z'} &= 0, \qquad \frac{\partial C_A(L,t')}{\partial z'} = 0
\end{aligned} \tag{2.2}$$

$$T(z',0) = T_0(z'), \qquad C_A(z',t') = C_{A0}(z') \tag{2.3}$$

and subject to the following constraints:

$$\begin{aligned} T^{min} &\leq \int_0^L r'(z')T(z',t')dz' \leq T^{max} \\ T_w^{min} &\leq T_w(t) \leq T_w^{max} \\ T_f^{min} &\leq T_f(t) \leq T_f^{max} \end{aligned} \tag{2.4}$$

where $T(z',t')$ denotes the temperature of reacting fluid in the reactor and $C_A(z',t')$ denotes the concentration of reactant A in the reactor; $z' \in [0,L]$ and $t \in [0,\infty)$ denote the spatial coordinate and the time, respectively; E_a, R_g and k_0 represent the activation energy, the gas constant, and the pre-exponential constant; U denotes the overall heat transfer coefficient and is assumed to be constant; the heat capacity, density, thermal conductivity, mass diffusivity and velocity of the reaction mixture are represented by C_{pf}, ρ_f, k, D_A and v, respectively and are fixed in constants. T^{max}, T_w^{max}, T_f^{max}, and T^{min}, T_w^{min}, T_f^{min} are all real numbers denoting the upper and lower constraints related to the temperature in the reactor, the temperature of the reactor wall, and the temperature of the inlet fluid, respectively. $r'(z') \in L_2[0,L]$, a square-integrable function, represents the state constraints distribution and illustrates where the temperature constraints are enforced within the spatial domain $[0,L]$.

In order to simplify the system representation we put the model in a dimensionless form by defining:

$$\begin{aligned} &t = \frac{t'v}{L}; \qquad z = \frac{z'}{L}; \qquad Pe_1 = \frac{\rho_f C_{pf} v L}{k}; \qquad Pe_2 = \frac{vL}{D_A} \\ &x_1(z,t) = \frac{T - T_r}{T_r}; \qquad x_2(z,t) = \frac{(C_A - C_{Ar})}{C_{Ar}} \\ &u_w = \frac{T_w - T_r}{T_r}; \qquad u_i = \frac{T_f - T_r}{T_r}; \qquad C_i = \frac{C_f - C_{Ar}}{C_{Ar}} \end{aligned}$$

$$x_{10} = \frac{T_0 - T_r}{T_r}; \qquad x_{20} = \frac{C_{A0} - C_{Ar}}{C_{Ar}}$$
$$B_1 = \frac{(-\Delta H)k_0 e^{-E_a/R_g T_r}}{\rho_f C_{pf} T_r}\frac{L}{v}; \qquad B_2 = \frac{k_0 L e^{E_a/R_g T_r}}{v}; \qquad \gamma = \frac{E_a}{R_g T_r}; \qquad \beta = \frac{2UL}{R\rho_f C_{pf} v}$$
$$\mathcal{X}^{min} = \frac{T^{min} - T_r}{T_r}; \qquad u_w^{min} = \frac{T_w^{min} - T_r}{T_r}; \qquad u_i^{min} = \frac{T_f^{min} - T_r}{T_r}$$
$$\mathcal{X}^{max} = \frac{T^{max} - T_r}{T_r}; \qquad u_w^{max} = \frac{T_w^{max} - T_r}{T_r}; \qquad u_i^{max} = \frac{T_f^{max} - T_r}{T_r}$$

where T_r and C_{Ar} are the reference temperature and concentration, respectively. Then the system of Eqs. (2.1)-(2.3) can be written in the following form:

$$\frac{\partial x_1(z,t)}{\partial t} = -\frac{\partial x_1(z,t)}{\partial z} + \frac{1}{Pe_1}\frac{\partial^2 x_1(z,t)}{\partial z^2} + B_1 e^{\gamma x_1(z,t)/(1+x_1(z,t))}(1 + x_2(z,t)) - \beta\big(x_1(z,t) - u_w(t)\big)$$
$$\frac{\partial x_2(z,t)}{\partial z} = -\frac{\partial x_2(z,t)}{\partial z} + \frac{1}{Pe_2}\frac{\partial^2 x_2(z,t)}{\partial z^2} - B_2 e^{\gamma x_1(z,t)/(1+x_1(z,t))}(1 + x_2(z,t)) \tag{2.5}$$

subject to the following boundary and initial conditions:

$$\frac{\partial x_1(0,t)}{\partial z} = Pe_1\big(x_1(0,t) - u_i(t)\big); \qquad \frac{\partial x_2(0,t)}{\partial z} = Pe_2\big(x_2(0,t) - C_i\big)$$
$$\frac{\partial x_1(1,t)}{\partial z} = 0; \qquad \frac{\partial x_2(1,t)}{\partial z} = 0 \tag{2.6}$$

$$x_1(z,0) = x_{10}(z); \qquad x_2(z,0) = x_{20}(z) \tag{2.7}$$

as well as subject to the following input and state constraints:

$$u_w^{min} \le u_w(t) \le u_w^{max},\ u_i^{min} \le u_i(t) \le u_i^{max} \tag{2.8}$$

$$\mathcal{X}^{min} \le \int_0^1 r(z)x(z,t)dz \le \mathcal{X}^{max} \tag{2.9}$$

where $x_i(z,t)$ for $i = 1, 2$ represents the PDE state; $z \in [0,1]$ and $t \in [0,\infty)$ denote the spatial coordinate and the time, respectively. In Eq. (2.9), $r(z)$ denotes the dimensionless state constraints distribution function. The terms $\partial x_i(z,t)/\partial z$ and $\partial^2 x_i(z,t)/\partial z^2$ represent the first-order and second-order spatial derivatives of $x_i(z,t)$ for $i = 1, 2$, and $x_{i0}(z)$ is a sufficiently smooth function of z.

2.3 Eigenvalue problem

Since a nonlinear term exists in Eq. (2.5), it is essential to linearize this system about some steady state of interest, $x_{1s}(z)$, $x_{2s}(z)$, u_{ws}, u_{0s}, satisfying

$$
\begin{aligned}
0 &= -\frac{\partial x_{1s}}{\partial z} + \frac{1}{Pe_1}\frac{\partial^2 x_{1s}}{\partial z^2} + B_1 e^{\gamma x_{1s}/(1+x_{1s})}\big(1+x_{2s}\big) - \beta\big(x_{1s}-u_{ws}\big) \\
0 &= -\frac{\partial x_{2s}}{\partial z} + \frac{1}{Pe_2}\frac{\partial^2 x_{2s}}{\partial z^2} - B_2 e^{\gamma x_{1s}/(1+x_{1s})}\big(1+x_{2s}\big) \\
&\frac{\partial x_{1s}(z)}{\partial z}\bigg|_{z=0} = Pe_1\big(x_{1s}(0)-u_{is}\big); \qquad \frac{\partial x_{2s}(z)}{\partial z}\bigg|_{z=0} = Pe_2\big(x_{2s}(0)-C_i\big) \\
&\frac{\partial x_{1s}(z)}{\partial z}\bigg|_{z=1} = 0; \qquad \frac{\partial x_{2s}(z)}{\partial z}\bigg|_{z=1} = 0
\end{aligned}
\tag{2.10}
$$

After linearization, we obtain the linearized equation in $\hat{x}_i(z,t) = x_i(z,t) - x_{is}(z)$ for $i = 1, 2$, $\hat{u}_w(t) = u_w(t) - u_{ws}$, $\hat{u}_i(t) = u_i(t) - u_{is}$

$$
\begin{aligned}
\frac{\partial \hat{x}_1(z,t)}{\partial t} &= -\frac{\partial \hat{x}_1(z,t)}{\partial z} + \frac{1}{Pe_1}\frac{\partial^2 \hat{x}_1(z,t)}{\partial z^2} + J_{11}(z)\hat{x}_1(z,t) + J_{12}(z)\hat{x}_2(z,t) + \beta\hat{u}_w(t) \\
\frac{\partial \hat{x}_2(z,t)}{\partial t} &= -\frac{\partial \hat{x}_2(z,t)}{\partial z} + \frac{1}{Pe_2}\frac{\partial^2 \hat{x}_2(z,t)}{\partial z^2} + J_{21}(z)\hat{x}_1(z,t) + J_{22}(z)\hat{x}_2(z,t) \\
\frac{\partial \hat{x}_1(0,t)}{\partial z} &= Pe_1\big(\hat{x}_1(0,t) - \hat{u}_i(t)\big); \qquad \frac{\partial \hat{x}_2(0,t)}{\partial z} = Pe_2\hat{x}_2(0,t) \\
\frac{\partial \hat{x}_1(1,t)}{\partial z} &= 0; \qquad \frac{\partial \hat{x}_2(1,t)}{\partial z} = 0
\end{aligned}
\tag{2.11}
$$

where $J_{ij}(z)$ is the Jacobian of the nonlinear term evaluated at $x_{is}(z)$ for $i = 1, 2, j = 1, 2$

$$
\begin{bmatrix} J_{11}(z) & J_{12}(z) \\ J_{21}(z) & J_{22}(z) \end{bmatrix} = \begin{bmatrix} \frac{B_1\gamma}{(1+x_{1s})^2}(1+x_{2s})e^{\gamma x_{1s}/(1+x_{1s})} - \beta & B_1 e^{\gamma x_{1s}/(1+x_{1s})} \\ -\frac{B_2\gamma}{(1+x_{1s})^2}(1+x_{2s})e^{\gamma x_{1s}/(1+x_{1s})} & -B_2 e^{\gamma x_{1s}/(1+x_{1s})} \end{bmatrix}
\tag{2.12}
$$

The above boundary condition is nonhomogeneous due to the presence of $u_i(t)$. In order to transform the boundary control problem into a distributed control problem, we make the boundary condition in Eq. (2.11) homogeneous by inserting the nonhomogeneous part into the differential equation with a Dirac delta function [34]. Then, we suppress the ($\hat{}$)

notation and obtain the following equivalent equations:

$$\frac{\partial}{\partial t}\begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 & 0 \\ 0 & \mathcal{A}_2 \end{bmatrix}\begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} + \begin{bmatrix} J_{11}(z) & J_{12}(z) \\ J_{21}(z) & J_{22}(z) \end{bmatrix}\begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} + \begin{bmatrix} \beta\ u_w(t) + \delta(z)u_i(t) \\ 0 \end{bmatrix} \tag{2.13}$$

subject to

$$\begin{aligned} \frac{\partial}{\partial z}\begin{bmatrix} x_1(0,t) \\ x_2(0,t) \end{bmatrix} &= \begin{bmatrix} Pe_1 & 0 \\ 0 & Pe_2 \end{bmatrix}\begin{bmatrix} x_1(0,t) \\ x_2(0,t) \end{bmatrix} \\ \frac{\partial}{\partial z}\begin{bmatrix} x_1(1,t) \\ x_2(1,t) \end{bmatrix} &= 0 \end{aligned} \tag{2.14}$$

where the operator $\mathcal{A}_i$ is defined as:

$$\mathcal{A}_i\phi_i(z) = -\frac{d\phi_i(z)}{dz} + \frac{1}{Pe_i}\frac{d^2\phi_i(z)}{dz^2} \tag{2.15}$$

for $i = 1, 2$, and $\phi_i(z)$ denotes a smooth function defined on the spatial interval $[0, 1]$, with its dense domain:

$$\begin{aligned} D(\mathcal{A}_i) = \{\phi_i(z) \in L_2[0,1] :\ & \phi_i(z),\ \frac{d\phi_i(z)}{dz},\ \textit{abs. cont.}, \\ & \frac{d^2\phi_i(z)}{dz} \in L_2[0,1], \\ & \frac{d\phi_i(0)}{dz} = Pe\ \phi_i(0), \quad \frac{d\phi_i(1)}{dz} = 0\} \end{aligned} \tag{2.16}$$

In order to simplify the mathematical model representation, we reformulate the PDEs of Eq. (2.13) in the form of an infinite-dimensional system defined in the space $\mathcal{H} = L_2[0, 1]$, with the following inner product and norm:

$$(\gamma_1, \gamma_2) = \int_0^1 \gamma_1(z)\gamma_2(z)dz, \qquad \|\gamma_1\|_2 = (\gamma_1, \gamma_1)^{1/2} \tag{2.17}$$

where γ_1, γ_2 are both elements of $L_2[0, 1]$.

The eigenvalue problem of the operator $\mathcal{A}_i$ is in the form of:

$$-\frac{d\phi_{in}}{dz} + \frac{1}{Pe_i}\frac{d^2\phi_{in}}{dz^2} = \lambda_{in}\phi_{in} \tag{2.18}$$

subject to

$$\begin{aligned} \frac{d\phi_{in}(0)}{dz} &= Pe_i\ \phi_{in}(0) \\ \frac{d\phi_{in}(1)}{dz} &= 0 \end{aligned} \tag{2.19}$$

for $i = 1, 2; n = 1, \cdots, \infty$. λ_{in} represents an eigenvalue and ϕ_{in} represents the eigenfunction associated with λ_{in}.

Since the operator $\mathcal{A}_i$ is a non-self-adjoint linear operator, Eq. (2.18) should be put into Sturm-Liouville form:

$$\frac{1}{Pe_i}e^{Pe_i z}\frac{d}{dz}(e^{-Pe_i z}\frac{d\phi_{in}(z)}{dz}) - \lambda_{in}\phi_{in}(z) = 0 \tag{2.20}$$

with the orthonormality relation between the system eigenfunction $\phi_{in}(z)$ and the adjoint eigenfunction $\phi^*_{in}(z)$:

$$\langle \phi_{in}(z), \phi^*_{im}(z) \rangle = \delta_{mn} \tag{2.21}$$

where δ_{mn} is Kronecker delta function, and the adjoint eigenfunction is given as:

$$\phi^*_{in}(z) = e^{-Pe_i z}\phi_{in}(z) \tag{2.22}$$

Now the eigenvalues and eigenfunctions of the operator $\mathcal{A}$ can be solved analytically with the solution given by:

$$\begin{aligned} \lambda_{in} &= -\frac{\alpha^2_{in}}{Pe_i} - \frac{Pe_i}{4} \\ \phi_{in}(z) &= A_{in}e^{Pe_i z/2}\left(\cos(\alpha_{in}z) + \frac{Pe_i}{2\alpha_{in}}\sin(\alpha_{in}z) \right) \end{aligned} \tag{2.23}$$

where,

$$tan\ \alpha_{in} = \frac{Pe_i\alpha_{in}}{\alpha^2_{in} - (Pe_i/2)^2}$$

$$A_{in} = \left[\int_0^1 \left(\cos(\alpha_{in} z) + \frac{Pe_i}{2\alpha_{in}} \sin(\alpha_{in} z) \right)^2 dz \right]^{-1/2} \tag{2.24}$$

for $i = 1, 2; n = 1, \cdots, \infty$.

Remark 1 *The Péclet numbers Pe_1 and Pe_2 are defined to be the ratio of the rate of convection to the rate of diffusion for heat transport and mass transport, respectively. The value of the Péclet number determines the dominant transport phenomena between convection and diffusion. In this work, the process with $Pe_1 = Pe_2 = Pe$ is considered, where Pe is a constant and belongs to the range $(0.1, 10)$. In this range, this choice of Péclet numbers means that the importance of convection and diffusion are comparable in both heat and mass transport. However, if $Pe_1 \neq Pe_2$, we have to solve two eigenvalue problems numerically in order to obtain a unique set of eigenvalues.*

Since $Pe_1 = Pe_2 = Pe$ is chosen in this work, the notations λ_n, ϕ_n and ϕ_n^* will be used instead of λ_{in}, ϕ_{in} and ϕ_{in}^* in the following sections.

Next, Galerkin's method is applied to the PDEs system of Eq. (2.13). Assume that the solution to PDEs is in the following form:

$$x_i(z,t) = \sum_{n=1}^{\infty} a_{in}(t)\phi_n(z), \quad i = 1, 2 \tag{2.25}$$

Projecting Eq. (2.13) on the adjoint eigenfunction domain, one can obtain the following modal representation:

$$\frac{d}{dt}\begin{bmatrix} a_{1n}(t) \\ a_{2n}(t) \end{bmatrix} = \lambda_n \begin{bmatrix} a_{1n}(t) \\ a_{2n}(t) \end{bmatrix} + \begin{bmatrix} b_n & c_n \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_w(t) \\ u_i(t) \end{bmatrix} + \begin{bmatrix} \sum_{m=1}^{\infty} a_{1m}(t) f_{nm}^{11} + \sum_{m=1}^{\infty} a_{2m}(t) f_{nm}^{12} \\ \sum_{m=1}^{\infty} a_{1m}(t) f_{nm}^{21} + \sum_{m=1}^{\infty} a_{2m}(t) f_{nm}^{22} \end{bmatrix} \tag{2.26}$$

where,

$$f_{nm}^{ij} = \int_0^1 J_{ij}(z)\phi_n^*(z)\phi_m(z)dz, \quad i = 1, 2,\ j = 1, 2 \tag{2.27}$$

$$b_n = \beta \int_0^1 \phi_n^*(z) dz \tag{2.28}$$

$$c_n = \phi_n^*(0) \tag{2.29}$$

For the purpose of controller design in the following sections, two measurements are set at the output (right end) of the reactor, which are described as:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} x_1(1,t) \\ x_2(1,t) \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{\infty} a_{1n}(t)\phi_n(z=1) \\ \sum_{n=1}^{\infty} a_{2n}(t)\phi_n(z=1) \end{bmatrix} \tag{2.30}$$

Rewrite Eq. (2.26) as the following form:

$$\begin{cases} \dot{a}(t) = Aa(t) + Bu(t) \\ y(t) = Ca(t) \end{cases} \tag{2.31}$$

where,

$$a(t) = \begin{bmatrix} a_{11}(t) & a_{21}(t) & a_{12}(t) & a_{22}(t) & \cdots & a_{1n}(t) & a_{2n}(t) & \cdots \end{bmatrix}^T$$

$$A = \begin{bmatrix} \lambda_1 + f_{11}^{11} & f_{11}^{12} & f_{12}^{11} & f_{12}^{12} & \cdots & f_{1n}^{11} & f_{1n}^{12} & \cdots \\ f_{11}^{21} & \lambda_1 + f_{11}^{22} & f_{12}^{21} & f_{12}^{22} & \cdots & f_{1n}^{21} & f_{1n}^{22} & \cdots \\ f_{21}^{11} & f_{21}^{12} & \lambda_2 + f_{22}^{11} & f_{22}^{12} & \cdots & f_{2n}^{11} & f_{2n}^{12} & \cdots \\ f_{21}^{21} & f_{21}^{22} & f_{22}^{21} & \lambda_2 + f_{22}^{22} & \cdots & f_{2n}^{21} & f_{2n}^{22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ f_{n1}^{11} & f_{n1}^{12} & f_{n2}^{11} & f_{n2}^{12} & \cdots & \lambda_n + f_{nn}^{11} & f_{nn}^{12} & \cdots \\ f_{n1}^{21} & f_{n1}^{22} & f_{n2}^{21} & f_{n2}^{22} & \cdots & f_{nn}^{21} & \lambda_n + f_{nn}^{22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 & 0 & b_2 & 0 & \cdots & b_n & 0 & \cdots \\ c_1 & 0 & c_2 & 0 & \cdots & c_n & 0 & \cdots \end{bmatrix}^T$$

$$C = \begin{bmatrix} \phi_1(1) & 0 & \phi_2(1) & 0 & \cdots & \phi_n(1) & 0 & \cdots \\ 0 & \phi_1(1) & 0 & \phi_2(1) & \cdots & 0 & \phi_n(1) & \cdots \end{bmatrix} \tag{2.32}$$

The model representation in Eq. (2.31) needs to be transformed into an equivalent discrete-time representation in order to be utilized for MPC construction. Appropriate discrete-time model representation must be adopted to decouple slow dynamics and fast dynamics.

2.4 Modal decomposition

To this end, the technique of modal decomposition is applied on the system representation (infinite-dimensional) of Eq. (2.31) in order to obtain an approximate system representation (finite-dimensional). Spectral projection operators $\mathcal{P}_s$ and $\mathcal{P}_f$ are defined, such that the space $\mathcal{H}$ can be partitioned into two subspaces, $\mathcal{H}_s$ and $\mathcal{H}_f$, which are given as $\mathcal{H}_s = span\{\phi_1^*(z), \phi_2^*(z), ...\phi_q^*(z)\}$ and $\mathcal{H}_f = span\{\phi_{q+1}^*(z), \phi_{q+2}^*(z), ...\}$ (q can be an arbitrary positive integer greater than 1 because all $\lambda < 0$ for this system). For $i = 1, 2$, the eigen-coefficients $a_i(t)$ of the PDE system can be decomposed as:

$$\begin{aligned} a_{is}(t) = P_s a_i(t) = \langle \phi_s^*(z), x_i(z,t) \rangle = \left\langle \phi_s^*(z), \sum_{n=1}^{\infty} a_{in}(t)\phi_n(z) \right\rangle \\ a_{if}(t) = P_f a_i(t) = \langle \phi_f^*(z), x_i(z,t) \rangle = \left\langle \phi_f^*(z), \sum_{n=1}^{\infty} a_{in}(t)\phi_n(z) \right\rangle \end{aligned} \tag{2.33}$$

Thus, we get two subsystems, $a_s(t)-$ and $a_f(t)-$ subsystem, which are referred to as slow subsystem and fast subsystem, respectively. The slow subsystem is a finite-dimensional subsystem which captures the dominant dynamic characteristic of the PDE system, whereas the fast subsystem is an infinite-dimensional subsystem. Thus, the continuous-time model dynamic given by Eq. (2.31) can be written as:

$$\begin{bmatrix} \dot{a}_s(t) \\ \dot{a}_f(t) \end{bmatrix} = \begin{bmatrix} A_s & A_{12} \\ A_{21} & A_f \end{bmatrix} \begin{bmatrix} a_s(t) \\ a_f(t) \end{bmatrix} + \begin{bmatrix} B_s \\ B_f \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} C_s & C_f \end{bmatrix} \begin{bmatrix} a_s(t) \\ a_f(t) \end{bmatrix} \tag{2.34}$$

2.5 Discrete-time system representation

There are two important issues arising from transformation of Eq. (2.34) to a discrete counterpart. First is the question of transforming the continuous-time infinite-dimensional system to discrete-time infinite-dimensional system representation, and the second one is the degree of approximation applied on the discrete-time model in order to obtain a model suitable for the MPC realization. In the ensuing section, two discretization methods are explored and the obtained finite-dimensional approximations are discussed.

2.5.1 Standard discretization

In the first method, the fast subsystem is truncated by a $2l^{th}$ order approximation. Transform the slow subsystem and the approximation of the fast subsystem ($2l^{th}$ order approximation) into an appropriate discrete-time equivalent of the continuous-time dynamics:

$$\begin{bmatrix} a_s(k+1) \\ a_f(k+1) \end{bmatrix} = \begin{bmatrix} A_{ds} & A_{d12} \\ A_{d21} & A_{df} \end{bmatrix} \begin{bmatrix} a_s(k) \\ a_f(k) \end{bmatrix} + \begin{bmatrix} B_{ds} \\ B_{df} \end{bmatrix} u(k)$$

$$y(k) = C_{ds} a_s(k) + C_{df} a_f(k) \tag{2.35}$$

where,

$$\begin{bmatrix} A_{ds} & A_{d12} \\ A_{d21} & A_{df} \end{bmatrix} = A_d, \quad \begin{bmatrix} B_{ds} \\ B_{df} \end{bmatrix} = B_d, \quad \begin{bmatrix} C_{ds} & C_{df} \end{bmatrix} = C_d, \tag{2.36}$$

and A_d, B_d, C_d are calculated in the following way:

$$\begin{bmatrix} A_d & B_d \\ C_d & 0 \end{bmatrix} = \begin{bmatrix} e^{A_{2l}\Delta t} & \int_0^{\Delta t} e^{A_{2l}\tau} d\tau B_{2l} \\ C_{2l} & 0 \end{bmatrix} \tag{2.37}$$

where Δt denotes the discretization time interval; A_{2l}, B_{2l}, and C_{2l} are the $2l^{th}$ order approximations of A, B, and C in Eq. (2.32).

Since the fast and slow dynamics are coupled by the off-diagonal elements of matrix A_d, it is important to decompose this coupling. The above system can always be diagonalized if the eigenvalues of the matrix A_d belong to the point spectrum of A_d. In another words, there exists a transformation matrix V such that

$$V^{-1}A_dV = \begin{bmatrix} \check{A}_{ds} & 0 \\ 0 & \check{A}_{df} \end{bmatrix}$$

Then, we do the following state and matrix transformation with V:

$$\begin{bmatrix} a_s(k) \\ a_f(k) \end{bmatrix} = V \begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix}, \quad \begin{bmatrix} \check{B}_s \\ \check{B}_f \end{bmatrix} = V^{-1} \begin{bmatrix} B_s \\ B_f \end{bmatrix}, \quad \begin{bmatrix} \check{C}_s & \check{C}_f \end{bmatrix} = \begin{bmatrix} C_s & C_f \end{bmatrix} V \tag{2.38}$$

Suppressing the ($\check{}$) notation, one may obtain the following equivalent discrete-time system:

$$\begin{aligned} z_s(k+1) &= A_{ds}z_s(k) + B_{ds}u(k) \\ z_f(k+1) &= A_{df}z_f(k) + B_{df}u(k) \\ y(k) &= C_{ds}z_s(k) + C_{df}z_f(k) \end{aligned} \tag{2.39}$$

where A_{ds}, A_{df}, B_{ds}, B_{df}, C_{ds}, and C_{df} are $(2q \times 2q)$, $((2l-2q) \times (2l-2q))$, $(2q \times 2)$, $((2l-2q) \times 2)$, $(2 \times 2q)$, and $(2 \times (2l-2q))$ matrices, respectively. This is a diagonal structure where the slow modes and fast modes are not coupled with each other and the only coupling is through the control injection.

2.5.2 Discretization by Cayley transform

In the second method, the infinite-dimensional system of Eq. (2.34) is transformed directly using Tustin's approximation and Cayley transform to the following discrete-time equiva-

lent:

$$\begin{bmatrix} a_s(k) \\ a_f(k) \end{bmatrix} = \begin{bmatrix} \bar{A}_{ds} & \bar{A}_{d12} \\ \bar{A}_{d21} & \bar{A}_{df} \end{bmatrix} \begin{bmatrix} a_s(k-1) \\ a_f(k-1) \end{bmatrix} + \begin{bmatrix} \bar{B}_{ds} \\ \bar{B}_{df} \end{bmatrix} u(k)$$

$$y(k) = \bar{C}_{ds} a_s(k-1) + \bar{C}_{df} a_f(k-1) + \bar{D}_d u(k) \tag{2.40}$$

where,

$$\begin{bmatrix} \bar{A}_{ds} & \bar{A}_{d12} \\ \bar{A}_{d21} & \bar{A}_{df} \end{bmatrix} = \bar{A}_d, \quad \begin{bmatrix} \bar{B}_{ds} \\ \bar{B}_{df} \end{bmatrix} = \bar{B}_d, \quad \begin{bmatrix} \bar{C}_{ds} & \bar{C}_{df} \end{bmatrix} = \bar{C}_d \tag{2.41}$$

In order to get the formulation of $\bar{A}_d$, $\bar{B}_d$, $\bar{C}_d$, and $\bar{D}_d$, we introduce a unique extension (by density and continuity) of the operator A, the operator A_{-1}, known as the Yosida extension of $A \in \mathscr{L}(X; X_{-1})$ where X is the Hilbert space and $dom(A) \subset X$. Take $\alpha \in \rho(A)$, where $\rho(A)$ is the nonempty resolvent set of A, and define $\|x\|_{X_1} = \|(\alpha - A)x\|_X$ for each $x \in dom(A)$. Then X_1 is a Hilbert space with respect to the norm $\|x\|_{X_1}$ and $A \in \mathscr{L}(X_1; X)$. The Hilbert space X_{-1} is defined as the completion of X with respect to the norm $\|x\|_{X_{-1}} = \|(\alpha - A)^{-1}x\|_X$ [35]. On the basis of all of the above definitions, $\bar{A}_d$, $\bar{B}_d$, $\bar{C}_d$, and $\bar{D}_d$ are given by the Cayley transform of the infinite-dimensional system:

$$\begin{bmatrix} \bar{A}_d & \bar{B}_d \\ \bar{C}_d & \bar{D}_d \end{bmatrix} = \begin{bmatrix} (\delta + A)(\delta - A)^{-1} & \sqrt{2\delta}(\delta - A_{-1})^{-1}B \\ \sqrt{2\delta}C(\delta - A)^{-1} & \mathscr{G}(\delta) \end{bmatrix} \tag{2.42}$$

where $\delta = 2/\Delta t$ and $\Delta t > 0$ denotes the discretization time interval. $\mathscr{G}(\delta)$ is defined as the transfer function of the PDE system evaluated at δ.

The unbounded operators A, B and C of the continuous-time system are mapped into bounded operators A_d, B_d and C_d in the discrete-time counterpart through Cayley transform. And it turns out that control properties, such as controllability, are the same for both systems [36].

As for the transfer function $\mathscr{G}(s)$ of the PDE system, two methods are used to obtain

it in this work. In the first method, a closed-form expression for $\mathscr{G}(s)$ can be obtained by taking Laplace transform of the PDEs in Eq. (2.13).

Let us first take Laplace transform of the PDEs, which gives the following boundary value problem for $x_i(z,s)$, where s is regarded as a parameter:

$$\frac{d}{dz}\begin{bmatrix} x_1(z,s) \\ \frac{dx_1(z,s)}{dz} \\ x_2(z,s) \\ \frac{dx_2(z,s)}{dz} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ Pe_1\big(s-J_{11}(z)\big) & Pe_1 & -Pe_1J_{12}(z) & \\ 0 & 0 & 0 & 1 \\ -Pe_2J_{21} & 0 & Pe_2\big(s-J_{22}\big) & Pe_2 \end{bmatrix} \begin{bmatrix} x_1(z,s) \\ \frac{dx_1(z,s)}{dz} \\ x_2(z,s) \\ \frac{dx_2(z,s)}{dz} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -Pe_1\beta & -Pe_1\delta(z) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_w(s) \\ u_i(s) \end{bmatrix} \tag{2.43}$$

subject to

$$\begin{aligned} &\frac{dx_1(0,s)}{dz} = Pe_1x_1(0,s), \qquad \frac{dx_2(0,s)}{dz} = Pe_2x_2(0,s) \\ &\frac{x_1(1,s)}{dz} = \frac{x_2(1,s)}{dz} = 0 \end{aligned} \tag{2.44}$$

which has the general solution:

$$\begin{bmatrix} x_1(z,s) \\ \frac{dx_1(z,s)}{dz} \\ x_2(z,s) \\ \frac{dx_2(z,s)}{dz} \end{bmatrix} = e^{\mathbf{\Gamma}z} \begin{bmatrix} x_1(0,s) \\ Pe_1x_1(0,s) \\ x_2(0,s) \\ Pe_2x_2(0,s) \end{bmatrix} + \int_0^z e^{\mathbf{\Gamma}(z-\xi)} \begin{bmatrix} 0 & 0 \\ -Pe_1\beta & -Pe_1\delta(\xi) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_w(s) \\ u_i(s) \end{bmatrix} d\xi \tag{2.45}$$

where the matrix

$$\mathbf{\Gamma} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ Pe_1\big(s-J_{11}(z)\big) & Pe_1 & -Pe_1J_{12}(z) & \\ 0 & 0 & 0 & 1 \\ -Pe_2J_{21} & 0 & Pe_2\big(s-J_{22}\big) & Pe_2 \end{bmatrix}$$

We cannot treat the matrix exponential of $\mathbf{\Gamma}$ directly because of the existence of a spatial variable $J_{ij}(z)$. J_{ij}^a, which is defined as a spatial average of $J_{ij}(z)$, should be used instead of $J_{ij}(z)$ for computing the matrix exponential of $\mathbf{\Gamma}$. For the sake of simplicity, θ_{ij} and κ_{ij} are used to denote the entries in the i^{th} row and j_{th} column of matrices $e^{\Gamma z}$ and $e^{\Gamma(z-\xi)}$, respectively.

According to the boundary condition of Eq. (2.44) and the solution of Eq. (2.45), we have

$$\begin{bmatrix} 0 \\ \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{dx_1(1,s)}{dz} \\ \\ \frac{dx_2(1,s)}{dz} \end{bmatrix}$$

$$= \begin{bmatrix} (\theta_{21}+\theta_{22}Pe_1)x_1(0,s)+(\theta_{23}+\theta_{24}Pe_2)x_2(0,s)-Pe_1\beta\int_0^z \kappa_{22}d\xi u_w(s)-Pe_1\int_0^z \delta(\xi)\kappa_{22}d\xi u_i(s)\big|_{z=1} \\ (\theta_{41}+\theta_{42}Pe_1)x_1(0,s)+(\theta_{43}+\theta_{44}Pe_2)x_2(0,s)-Pe_1\beta\int_0^z \kappa_{42}d\xi u_w(s)-Pe_1\int_0^z \delta(\xi)\kappa_{42}d\xi u_i(s)\big|_{z=1} \end{bmatrix} \tag{2.46}$$

Thus,

$$\begin{aligned} x_1(0,s) = & \left.\frac{(\theta_{43}+\theta_{44}Pe_2)Pe_1\beta\int_0^z \kappa_{22}d\xi-(\theta_{23}+\theta_{24}Pe_2)Pe_1\beta\int_0^z \kappa_{42}d\xi}{(\theta_{21}+\theta_{22}Pe_1)(\theta_{43}+\theta_{44}Pe_2)-(\theta_{23}+\theta_{24}Pe_2)(\theta_{41}+\theta_{42}Pe_1)}\right|_{z=1} u_w(s) \\ & + \left.\frac{(\theta_{43}+\theta_{44}Pe_2)Pe_1\int_0^z \delta(\xi)\kappa_{22}d\xi-(\theta_{23}+\theta_{24}Pe_2)Pe_1\int_0^z \delta(\xi)\kappa_{42}d\xi}{(\theta_{21}+\theta_{22}Pe_1)(\theta_{43}+\theta_{44}Pe_2)-(\theta_{23}+\theta_{24}Pe_2)(\theta_{41}+\theta_{42}Pe_1)}\right|_{z=1} u_i(s) \\ x_2(0,s) = & \left.\frac{(\theta_{41}+\theta_{42}Pe_1)Pe_1\beta\int_0^z \kappa_{22}d\xi-(\theta_{21}+\theta_{22}Pe_1)Pe_1\beta\int_0^z \kappa_{42}d\xi}{(\theta_{41}+\theta_{42}Pe_1)(\theta_{23}+\theta_{24}Pe_2)-(\theta_{43}+\theta_{44}Pe_2)(\theta_{21}+\theta_{22}Pe_1)}\right|_{z=1} u_w(s) \\ & + \left.\frac{(\theta_{41}+\theta_{42}Pe_1)Pe_1\int_0^z \delta(\xi)\kappa_{22}d\xi-(\theta_{21}+\theta_{22}Pe_1)Pe_1\int_0^z \delta(\xi)\kappa_{42}d\xi}{(\theta_{41}+\theta_{42}Pe_1)(\theta_{23}+\theta_{24}Pe_2)-(\theta_{43}+\theta_{44}Pe_2)(\theta_{21}+\theta_{22}Pe_1)}\right|_{z=1} u_i(s) \end{aligned} \tag{2.47}$$

For the sake of simplicity, henceforth $x_1(0,s)$ and $x_2(0,s)$ are represented in the following form:

$$x_1(0,s) = p_{1w}u_w(s) + p_{1i}u_i(s)$$

$$x_2(0,s) = p_{2w}u_w(s) + p_{2i}u_i(s)$$

Since the measurement is located at the output (right end) of the reactor, we obtain

$$\begin{aligned} y_1(s) =& x_1(1,s) \\ =& (\theta_{11}+\theta_{12}Pe_1)x_1(0,s)+(\theta_{13}+\theta_{14}Pe_2)x_2(0,s)-Pe_1\beta\int_0^z \kappa_{12}d\xi u_w(s)-Pe_1\int_0^z \delta(\xi)\kappa_{12}d\xi u_i(s) \\ =& \big[(\theta_{11}+\theta_{12}Pe_1)p_{1w}+(\theta_{13}+\theta_{14}Pe_2)p_{2w}-Pe_1\beta\int_0^z \kappa_{12}d\xi\big]u_w(s) \\ & + \big[(\theta_{11}+\theta_{12}Pe_1)p_{1i}+(\theta_{13}+\theta_{14}Pe_2)p_{2i}-Pe_1\int_0^z \delta(\xi)\kappa_{12}d\xi\big]u_i(s) \\ y_2(s) =& x_2(1,s) \\ =& (\theta_{31}+\theta_{32}Pe_1)x_1(0,s)+(\theta_{33}+\theta_{34}Pe_2)x_2(0,s)-Pe_1\beta\int_0^z \kappa_{32}d\xi u_w(s)-Pe_1\int_0^z \delta(\xi)\kappa_{32}d\xi u_i(s) \\ =& \big[(\theta_{31}+\theta_{32}Pe_1)p_{1w}+(\theta_{33}+\theta_{34}Pe_2)p_{2w}-Pe_1\beta\int_0^z \kappa_{32}d\xi\big]u_w(s) \\ & + \big[(\theta_{31}+\theta_{32}Pe_1)p_{1i}+(\theta_{33}+\theta_{34}Pe_2)p_{2i}-Pe_1\int_0^z \delta(\xi)\kappa_{32}d\xi\big]u_i(s) \end{aligned} \tag{2.48}$$

Now, the transfer function of the PDE system:

$$\mathscr{G}(s) = \begin{bmatrix} \mathscr{G}_{1w}(s) & \mathscr{G}_{1i}(s) \\ \mathscr{G}_{2w}(s) & \mathscr{G}_{2i}(s) \end{bmatrix} \tag{2.49}$$

can be represented in the following form:

$$\begin{aligned}
\mathscr{G}_{1w}(s) &= (\theta_{11} + \theta_{12} Pe_1) p_{1w} + (\theta_{13} + \theta_{14} Pe_2) p_{2w} - Pe_1 \beta \int_0^z \kappa_{12} d\xi \\
\mathscr{G}_{1i}(s) &= (\theta_{11} + \theta_{12} Pe_1) p_{1i} + (\theta_{13} + \theta_{14} Pe_2) p_{2i} - Pe_1 \int_0^z \delta(\xi) \kappa_{12} d\xi \\
\mathscr{G}_{2w}(s) &= (\theta_{31} + \theta_{32} Pe_1) p_{1w} + (\theta_{33} + \theta_{34} Pe_2) p_{2w} - Pe_1 \beta \int_0^z \kappa_{32} d\xi \\
\mathscr{G}_{2i}(s) &= (\theta_{31} + \theta_{32} Pe_1) p_{1i} + (\theta_{33} + \theta_{34} Pe_2) p_{2i} - Pe_1 \int_0^z \delta(\xi) \kappa_{32} d\xi
\end{aligned} \tag{2.50}$$

Then, the matrix $\bar{D}_d$ in Eq. (4.6) can be expressed as:

$$\bar{D}_d = \begin{bmatrix} \mathscr{G}_{1w}(\delta) & \mathscr{G}_{1i}(\delta) \\ \mathscr{G}_{2w}(\delta) & \mathscr{G}_{2i}(\delta) \end{bmatrix}$$

The second method to obtain the transfer function of the PDE is given as following:

$$\mathscr{G}(s) = \sum_{n=1}^{\infty} \frac{1}{s - \zeta_n} C \Psi_n (\overline{B * \Psi_n^*})^T \qquad \text{for } \zeta_n \in \rho(A) \tag{2.51}$$

where A, B and C are the operators in Eq. (2.32). ζ_n denotes an eigenvalue of the operator A; Ψ_n and Ψ_n^* denote the corresponding eigenfunction and adjoint eigenfunction of ζ_n, respectively.

Remark 2 *The difficulty of the method given by Eq.* (2.51) *is that the eigenvalue and eigenfunction of the operator A, i.e., ζ_n and Ψ_n, are not identical with λ_n and ϕ_n in Eq.* (2.23) *due to the existence of nonlinear terms $J_{ij}(z)x(z,t)$, $i = 1, 2; j = 1, 2$ in Eq.* (2.13). *Similarly, the adjoint eigenfunction of the operator A, Ψ_n^*, is never the same as ϕ_n^*.*

Since the matrix $\bar{A}_d$ is diagonalizable with the transformation matrix $\bar{V}$ such that

$$\bar{V}^{-1}\bar{A}_d\bar{V} = \begin{bmatrix} \breve{\bar{A}}_{ds} & 0 \\ 0 & \breve{\bar{A}}_{df} \end{bmatrix}$$

we do the following state and matrix transformation with $\bar{V}$:

$$\begin{bmatrix} \bar{z}_s(k) \\ \bar{z}_f(k) \end{bmatrix} = \bar{V}^{-1} \begin{bmatrix} a_s(k) \\ a_f(k) \end{bmatrix}, \quad \begin{bmatrix} \breve{\bar{B}}_s \\ \breve{\bar{B}}_f \end{bmatrix} = \bar{V}^{-1} \begin{bmatrix} \bar{B}_s \\ \bar{B}_f \end{bmatrix}, \quad \begin{bmatrix} \breve{\bar{C}}_s & \breve{\bar{C}}_f \end{bmatrix} = \begin{bmatrix} \bar{C}_s & \bar{C}_f \end{bmatrix} \bar{V}, \quad \breve{\bar{D}}_d = \bar{D} \tag{2.52}$$

Suppressing the ($\breve{}$) notation, the discrete time system of Eq. (2.40) can be rewritten as following:

$$\begin{aligned} \bar{z}_s(k) &= \bar{A}_{ds}\bar{z}_s(k-1) + \bar{B}_{ds}u(k) \\ \bar{z}_f(k) &= \bar{A}_{df}\bar{z}_f(k-1) + \bar{B}_{df}u(k) \\ y(k) &= \bar{C}_{ds}\bar{z}_s(k-1) + \bar{C}_{df}\bar{z}_f(k-1) + \bar{D}_d u(k) \end{aligned} \tag{2.53}$$

Now we have two approximations of the system representation in Eqs. (2.31)-(2.32), given by Eq. (2.39) and Eq. (2.53), respectively. The major difference between these two representations is that the input contributes directly to the output in the second approximation while the output in the first approximation is only associated with the state.

2.6 Modal model predictive control

In this section, two types of linear time-invariant discrete-time model dynamics as developed in previous sections are utilized in the formulation of the model predictive control and they are given in the standard exact discretization form:

$$\begin{cases} z(k+1) = A_d z(k) + B_d u(k) \\ y(k) = C_d z(k) \end{cases} \tag{2.54}$$

and the Tustin's discretization form:

$$\begin{cases} \bar{z}(k) = \bar{A}_d \bar{z}(k-1) + \bar{B}_d u(k) \\ y(k) = \bar{C}_d \bar{z}(k-1) + \bar{D}_d u(k) \end{cases} \tag{2.55}$$

The regulator is founded as the solution to an optimization problem such that the following open-loop performance objective function on an infinite horizon is minimized at the sampling time k:

$$\begin{aligned} \min_{u^N} \sum_{j=0}^{\infty} \Big(& y(k+j|k)^T Q y(k+j|k) + u(k+j|k)^T R u(k+j|k) \\ & + \Delta u(k+j|k)^T S \Delta u(k+j|k) \Big) \end{aligned} \tag{2.56}$$

where Q and S are symmetric positive semidefinite penalty matrices, and R is a symmetric positive definite matrix. $y(k+j|k)$ and $u(k+j|k)$ represent the output and input variable at future time $k+j$ predicted at current time k, and the term $\Delta u(k+j|k) = u(k+j|k) - u(k+i-1|k)$ denotes the change of an input vector at time $k+j$. The vector u^N contains the sequences $u(k|k), u(k+1|k), ..., u(k+N-1|k)$ in which the first element $u(k|k)$ is the future control action to be injected to the plant. At time $k+N$, the input vector $u(k+j|k)$ is set to zero and kept at zero for all $j \geq N$ in the calculation of the open-loop quadratic objective function value [37]. In the following formulations, we set the matrix $S = 0$.

First, the receding horizon regulator of Eq. (2.56) is applied on the discrete-time dynamic model of Eq. (2.54) which is developed by the standard discretization method. Since the slow and fast modes are coupled only by the input injection, the following modal model predictive control algorithm is based on an approximate state representation $z_s(k)$ which is constructed only with slow modes:

$$\min_{u^N} \quad \sum_{j=0}^{N-1} \Big(z_s(k+j|k)^T C_{ds}^T Q C_{ds} z_s(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big)$$

$$+ z_s(k+N|k)^T \bar{Q}_1 z_s(k+N|k) \tag{2.57}$$

$$\begin{aligned} s.t. \quad & z_s(k+j+1|k) = A_{ds} z_s(k+j|k) + B_{ds} u(k+j|k) \\ & u^{min} \leq u(k+j|k) \leq u^{max} \\ & \mathcal{X}^{min} \leq \int_0^1 r_s(z) x_s(z, k+j|k) dz \leq \mathcal{X}^{max} \end{aligned} \tag{2.58}$$

where u^N is a vector of future control inputs calculated on the finite horizon N, Q is a positive semidefinite matrix, R is a positive definite matrix, and $\bar{Q}_1$ denotes the terminal state penalty matrix. $r_s(z)$ represents the state constraint distribution function. The above constrained optimization problem can be considered as a quadratic programming problem. Given any initial condition $z_s(0)$, the above formulation gives a feasible solution. Although this formulation has a low-order characteristic, a main disadvantage is the fact that the MPC formulation of Eqs. (2.57)-(2.58), when performed on the full state system, can only satisfy input constraints but not guarantee the PDEs state constraints, as the fast modes $z_f(k)$ are not involved either in the objective function or in the PDEs state constraints equation. Hence, the contribution of fast modes dynamics must be accounted for in some manner by the MPC algorithm in order to satisfy the requirement of PDEs state constraints.

To this end, the algorithm of Eqs. (2.57)-(2.58) can be reformulated by explicitly incorporating the evolution of the fast subsystem into the PDEs state constraints equation [38]. The control move at time k, under the MPC law of this scenario, is calculated by solving the following minimization problem:

$$\begin{aligned} \min_{u^N} \quad & \sum_{j=0}^{N-1} \Big(z_s(k+j|k)^T C_{ds}^T Q C_{ds} z_s(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big) \\ & + z_s(k+N|k)^T \bar{Q}_1 z_s(k+N|k) \end{aligned} \tag{2.59}$$

$$s.t. \quad z_s(k+j+1|k) = A_{ds} z_s(k+j|k) + B_{ds} u(k+j|k)$$

$$z_f(k+j+1|k) = A_{df} z_f(k+j|k) + B_{df} u(k+j|k)$$

$$u^{min} \leq u(k+j|k) \leq u^{max}$$

$$\mathcal{X}^{min} \leq \int_0^1 r_s(z) x_s(z, k+j|k) dz + \int_0^1 r_f(z) x_f(z, k+j|k) dz \leq \mathcal{X}^{max} \tag{2.60}$$

In the predictive control law formulation Eqs. (2.59)-(2.60), the constrained optimization program is concerned with only slow dynamics; however, the PDEs state constraints equation is expressed by the formulation that includes two contributions. One contribution involves slow dynamics and the other complementary contribution is based on fast dynamics. In this way, the PDEs state constraints, which are applied in the construction of a control move, account for the fast subsystem in the MPC formulation. It is necessary to emphasize that although the fast dynamics are involved in the PDEs state constraints equation, they are not involved in the objective function, which keeps relatively low computational complexity.

Next, the receding horizon regulator of Eq. (2.56) is applied on the discrete-time dynamic model of Eq. (2.55) which is developed through Cayley transform. First, a state representation that is constructed only by slow modes, which is the same as Eqs. (2.57)-(2.58) in some sense, is utilized. In this case, the MPC formulation is given as:

$$\begin{aligned} \min_{u^N} \quad & \sum_{j=0}^{N-1} \Big((\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j))^T Q (\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j)) \\ & + u(k+j|k)^T R u(k+j|k) \Big) + \bar{z}_s(k+N|k)^T \bar{Q}_2 \bar{z}_s(k+N|k) \end{aligned} \tag{2.61}$$

$$\begin{aligned} s.t. \quad & \bar{z}_s(k+j|k) = \bar{A}_{ds} \bar{z}_s(k+j-1|k) + \bar{B}_{ds} u(k+j|k) \\ & u^{min} \leq u(k+j|k) \leq u^{max} \\ & \mathcal{X}^{min} \leq \int_0^1 r_s(z) x_s(z, k+j|k) dz \leq \mathcal{X}^{max} \end{aligned} \tag{2.62}$$

where u^N, N, Q, R, and $r_s(z)$ denote the same meanings as in Eqs. (2.57)-(2.58); $\bar{Q}_2$ denotes the terminal state penalty matrix. The constrained optimization problem in Eq. (2.61) can be considered as a quadratic programming for u^N in a similar way as [37].

Due to the different type of discretization, the above quadratic programming is given by

$$\min_{u^N} \quad (u^N)^T \bar{H} u^N + 2(u^N)^T \bar{G} z_s(k-1) \tag{2.63}$$

where the matrices $\bar{H}$ and $\bar{G}$ are given as following:

$$\bar{H} = \begin{bmatrix} \bar{B}_{ds}^T \bar{Q}_2 \bar{B}_{ds} + \bar{D}_d^T Q \bar{D}_d + R & \cdots & \bar{B}_{ds}^T \bar{A}_{ds}^{T(N-1)} \bar{Q}_2 \bar{B}_{ds} + \bar{B}_{ds}^T \bar{A}_{ds}^{T(N-2)} \bar{C}_{ds}^T Q \bar{D}_d \\ \bar{B}_{ds}^T \bar{Q}_2 \bar{A}_{ds} \bar{B}_{ds} + \bar{D}_d^T Q \bar{C}_{ds} \bar{B}_{ds} & \cdots & \bar{B}_{ds}^T \bar{A}_{ds}^{T(N-2)} \bar{Q}_2 \bar{B}_{ds} + \bar{B}_{ds}^T \bar{A}_{ds}^{T(N-3)} \bar{C}_{ds}^T Q \bar{D}_d \\ \vdots & \vdots & \vdots \\ \bar{B}_{ds}^T \bar{Q}_2 \bar{A}_{ds}^{(N-1)} \bar{B}_{ds} + \bar{D}_d^T Q \bar{C}_{ds} \bar{A}_{ds}^{(N-2)} \bar{B}_{ds} & \cdots & \bar{B}_{ds}^T \bar{Q}_2 \bar{B}_{ds} + \bar{D}_d^T Q \bar{D}_d + R \end{bmatrix} \tag{2.64}$$

$$\bar{G} = \begin{bmatrix} \bar{B}_{ds}^T \bar{Q}_2 \bar{A}_{ds} + \bar{D}_d^T Q \bar{C}_{ds} \\ \bar{B}_{ds}^T \bar{Q}_2 \bar{A}_{ds}^2 + \bar{D}_d^T Q \bar{C}_{ds} \bar{A}_{ds} \\ \vdots \\ \bar{B}_{ds}^T \bar{Q}_2 \bar{A}_{ds}^N + \bar{D}_d^T Q \bar{C}_{ds} \bar{A}_{ds}^{(N-1)} \end{bmatrix} \tag{2.65}$$

and $\bar{Q}_2$ is defined as the infinite sum:

$$\bar{Q}_2 = \sum_{i=0}^{\infty} \bar{A}_{ds}^{Ti} \bar{C}_{ds}^T Q \bar{C}_{ds} \bar{A}_{ds}^i \tag{2.66}$$

The MPC formulation of Eqs. (2.61)-(2.62) has the same disadvantage as in Eqs. (2.57)-(2.58) in the sense that when performed on the full state system it can only satisfy input constraints but not guarantee the PDEs state constraints. Therefore, the evolution of the fast subsystem should be incorporated into the state constraints equation. In this case, the MPC algorithm is given as the following constrained optimization problem:

$$\begin{aligned} \min_{u^N} \quad & \sum_{j=0}^{N-1} \Big((\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j))^T Q (\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j)) \\ & + u(k+j|k)^T R u(k+j|k) \Big) + \bar{z}_s(k+N|k)^T \bar{Q}_2 \bar{z}_s(k+N|k) \end{aligned} \tag{2.67}$$

$$s.t. \quad \bar{z}_s(k+j|k) = \bar{A}_{ds} \bar{z}_s(k+j-1|k) + \bar{B}_{ds} u(k+j|k)$$

$$\bar{z}_f(k+j|k) = \bar{A}_{df}\bar{z}_f(k+j-1|k) + \bar{B}_{df}u(k+j|k)$$

$$u^{min} \leq u(k+j|k) \leq u^{max}$$

$$\mathcal{X}^{min} \leq \int_0^1 r_s(z)x_s(z,k+j|k)dz + \int_0^1 r_f(z)x_f(z,k+j|k)dz \leq \mathcal{X}^{max} \tag{2.68}$$

In the above MPC formulations, the state constraints equations are all expressed with state constraints distribution functions which will be replaced later by specified output matrices from the discrete-time model representations in the ensuing simulation section. When implemented on the full state system, the MPC formulations of both Eqs. (2.59)-(2.60) and Eqs. (2.67)-(2.68) can satisfy the input constraints and also guarantee the PDEs state constraints. The major difference between the two MPC formulations is that the discrete-time model representations utilized for the construction of a predictive controller are obtained through different discretization methods. The discretization method with Cayley transform introduces the input injection directly into the state constraints equation together with the matrix $\bar{D}_d$. The direct contribution from input to state constraints equation leads to some consequence which will be shown in the following section.

2.7 Simulation studies

In this section, the performance of the two strategies of MPC formulations developed in the previous section is demonstrated and compared through computer simulations. We consider the parabolic PDE system given by Eqs. (2.5)-(2.9), with $Pe_1 = Pe_2 = 7$, $\beta = 2$, $B_1 = 0.25$, $B_2 = 0.1$, $\gamma = 7.5$. The operating steady state, $x_{is}(z)$, $i = 1, 2$, is verified to be a stable one with these values of parameters. The state constraints distribution function, $r(z)$, is given as $r(z) = \delta(z - z_c)$ for $z \in [0, 1]$ and $z_c = 1$, which means that the state constraints are only to be actualized at the single point of the right boundary. The control objective is to enforce the two control inputs $u_w(t)$ and $u_i(t)$ and the PDE state at a desired point subject

to the input and state constraints of Eq. (2.8) and (2.9). For this simulation, the first two eigenvalues are considered to be the dominant ones, which means that the slow subsystem contains only the first two modes. Thus we can derive the following two types of state space representations that describe the transient evolution of the first l pairs of modes through a modal decomposition technique:

$$\begin{aligned} z_s(k+1) &= A_{ds} z_s(k) + B_{ds} u(k) \\ z_f(k+1) &= A_{df} z_f(k) + B_{df} u(k) \\ y(k) &= C_{ds} z_s(k) + C_{df} z_f(k) \end{aligned} \tag{2.69}$$

and

$$\begin{aligned} \bar{z}_s(k) &= \bar{A}_{ds} \bar{z}_s(k-1) + \bar{B}_{ds} u(k) \\ \bar{z}_f(k) &= \bar{A}_{df} \bar{z}_f(k-1) + \bar{B}_{df} u(k) \\ y(k) &= \bar{C}_{ds} \bar{z}_s(k-1) + \bar{C}_{df} \bar{z}_f(k-1) + \bar{D}_d u(k) \end{aligned} \tag{2.70}$$

where $z_s(k) = [z_{11}(k)\, z_{21}(k)\, z_{12}(k)\, z_{22}(k)]^T$, $z_f(k) = [z_{13}(k)\, z_{23}(k) \cdots z_{1l}(k)\, z_{2l}(k)]^T$, $\bar{z}_s(k) = [\bar{z}_{11}(k)\, \bar{z}_{21}(k)\, \bar{z}_{12}(k)\, \bar{z}_{22}(k)]^T$, $\bar{z}_f(k) = [\bar{z}_{13}(k)\, \bar{z}_{23}(k) \cdots \bar{z}_{1l}(k)\, \bar{z}_{2l}(k)]^T$; $z_{in}(k), \bar{z}_{in}(k) \in \mathbb{R}$ are the n^{th} modes corresponding to the i^{th} PDE, $u(k) = [u_w(k)\, u_i(k)]$. We now implement the first two MPC formulations proposed in the previous section on the discrete-time system of Eqs. (2.69). The initial condition of PDEs is considered to be $x_{10}(z) = x_{20}(z) = 0.02\,\phi_1(z)$ in all ensuing simulations, which implies that $z_{in}(0) = V^{-1}\langle \phi_n^*(z), x_{i0}(z) \rangle$ and $\bar{z}_{in}(0) = \bar{V}^{-1}\langle \phi_n^*(z), x_{i0}(z) \rangle$. l is chosen to be 15 (further increases of the value l led to identical results). In the first case, the slow subsystem is utilized as the foundation of a predictive controller design, while the fast subsystem is not involved in the MPC development. For

this scenario, the MPC law is given by Eqs. (2.71)-(2.72):

$$\min_{u^N} \quad \sum_{j=0}^{N-1} \Big(z_s(k+j|k)^T C_{ds}^T Q C_{ds} z_s(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big)$$
$$+ z_s(k+N|k)^T \bar{Q}_1 z_s(k+N|k) \tag{2.71}$$

$$s.t. \quad z_s(k+j+1|k) = A_{ds} z_s(k+j|k) + B_{ds} u(k+j|k)$$
$$u^{min} \leq u(k+j|k) \leq u^{max}$$
$$\mathcal{X}^{min} \leq C_{ds} z_s(k) \leq \mathcal{X}^{max} \tag{2.72}$$

where $Q = I$, $R = 0.0001I$, and the regulator horizon $N = 100$. The input and state constraints are given as $u_w^{min} = u_i^{min} = -1.8$, $u_w^{max} = u_i^{max} = 1$, $\mathcal{X}_1^{min} = \mathcal{X}_2^{min} = -0.1$, $\mathcal{X}_1^{max} = \mathcal{X}_2^{max} = 0.7$. In all ensuing simulation studies, we keep the initial condition, input and state constraints identical with the above values. The resulting constrained optimization problem becomes a quadratic program which can be solved through the MATLAB subroutine QuadProg. Then we implement the optimal control input on the continuous full-state plant model. Fig. 2.3 shows the states profiles of the closed-loop system under the performance of the MPC law given by Eqs. (2.71)-(2.72). The profile of the corresponding constrained control inputs are shown in Fig. 2.2(b) (solid line). The profile of constrained state at $z = 1$ is shown in Fig. 2.2(a) (solid line), it is obvious that the upper limit of the PDEs state constraint is violated for a while. The violation of PDEs state constraints is the consequence of neglecting fast modes dynamics in the construction of the PDEs state constraints equation. In order to involve fast modes dynamics in the PDEs state constraints equation, we utilize the modified MPC formulation with the open-loop quadratic cost function and input and state constraints equations given as the following:

$$\min_{u^N} \quad \sum_{j=0}^{N-1} \Big(z_s(k+j|k)^T C_{ds}^T Q C_{ds} z_s(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big)$$

$$+ z_s(k+N|k)^T \bar{Q}_1 z_s(k+N|k) \tag{2.73}$$

$$\begin{aligned} s.t. \quad & z_s(k+j+1|k) = A_{ds} z_s(k+j|k) + B_{ds} u(k+j|k) \\ & z_f(k+j+1|k) = A_{df} z_f(k+j|k) + B_{df} u(k+j|k) \\ & u^{min} \leq u(k+j|k) \leq u^{max} \\ & \mathcal{X}^{min} \leq C_{ds} z_s(k) + C_{df} z_f(k) \leq \mathcal{X}^{max} \end{aligned} \tag{2.74}$$

The MPC tuning parameters are kept the same as those in Eqs. (2.71)-(2.72). Results are shown in Fig. 2.5(a) (solid line) and Fig. 2.6 where it is demonstrated that the PDEs state constraints are guaranteed for all time under the performance of the MPC law given by the formulations of Eqs. (2.73)-(2.74). The profile of corresponding constrained control inputs is given in Fig. 2.5(b) (solid line). It is demonstrated that under the MPC law given by Eqs. (2.73)-(2.74), the input constraints are active.

Next we implement the latter two MPC formulations proposed in the previous section on the discrete time system of Eqs. (2.70). Only slow subsystem is utilized as the foundation of a predictive controller design in this scenario. For this case, the control law is given by Eqs. (2.75)-(2.76):

$$\begin{aligned} \min_{u^N} \quad & \sum_{j=0}^{N-1} \Big((\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j))^T Q (\bar{C}_{ds} \bar{z}_s(k+j-1|k) + \bar{D}_{ds} u(k+j)) \\ & + u(k+j|k)^T R u(k+j|k) \Big) + \bar{z}_s(k+N|k)^T \bar{Q}_2 \bar{z}_s(k+N|k) \end{aligned} \tag{2.75}$$

$$\begin{aligned} s.t. \quad & \bar{z}_s(k+j|k) = \bar{A}_{ds} \bar{z}_s(k+j-1|k) + \bar{B}_{ds} u(k+j|k) \\ & u^{min} \leq u(k+j|k) \leq u^{max} \\ & \mathcal{X}^{min} \leq \bar{C}_{ds} \bar{z}_s(k-1) + \bar{D}_d u(k) \leq \mathcal{X}^{max} \end{aligned} \tag{2.76}$$

The values of Q, N, initial condition, input and state constraints are kept identical with those in Eqs. (2.71)-(2.72), while the value of R is $0.001I$ in this scenario. States evolution of the closed-loop system is shown in Fig. 2.4. The profile of the corresponding control inputs is shown in Fig. 2.2(b) (dashed line) and the constrained states are shown In Fig. 2.2(a) (dashed line). The upper limit of the PDEs state constraints is violated for a while. The violation of the state constraint is caused by the negligence of fast modes dynamics in the construction of the PDEs state constraints equation. In order to involve fast modes dynamics in the PDEs state constraints equation, we utilize the modified MPC formulation with the open-loop quadratic cost function and input and state constraints equations given as the following:

$$\begin{aligned} \min_{u^N} \quad & \sum_{j=0}^{N-1} \Big((\bar{C}_{ds}\bar{z}_s(k+j-1|k) + \bar{D}_{ds}u(k+j))^T Q(\bar{C}_{ds}\bar{z}_s(k+j-1|k) + \bar{D}_{ds}u(k+j)) \\ & + u(k+j|k)^T R u(k+j|k) \Big) + \bar{z}_s(k+N|k)^T \bar{Q}_2 \bar{z}_s(k+N|k) \end{aligned} \tag{2.77}$$

$$\begin{aligned} s.t. \quad & \bar{z}_s(k+j|k) = \bar{A}_{ds}\bar{z}_s(k+j-1|k) + \bar{B}_{ds}u(k+j|k) \\ & \bar{z}_f(k+j|k) = \bar{A}_{df}\bar{z}_f(k+j-1|k) + \bar{B}_{df}u(k+j|k) \\ & u^{min} \le u(k+j|k) \le u^{max} \\ & \mathcal{X}^{min} \le \bar{C}_{ds}\bar{z}_s(k-1) + \bar{C}_{df}\bar{z}_f(k-1) + \bar{D}_d u(k) \le \mathcal{X}^{max} \end{aligned} \tag{2.78}$$

The MPC tuning parameters, input and state constraints are kept identical as those in Eqs. (2.75)-(2.76). The results shown in Fig. 2.5(a) (dashed line) and Fig. 2.7 demonstrate that the state constraints are satisfied for all time under implementation of the predictive control formulation of Eqs. (2.77)-(2.78). The profile of corresponding control inputs is given in Fig. 2.5(b) (dashed line), which shows that the input constraints are also active.

As demonstrated clearly by Fig. 2.5(a), the PDEs state constraints can be guaranteed by accounting for the dynamics of fast modes in the states constraints equation for the MPC

formulations constructed on both standard and Tustin's discrete-time system. However, in order to decide which optimal control sequence is better for implementation on a continuous-time plant model, the performance of MPC laws constructed on standard and Tustin's discrete-time system is compared in Table 2.1.

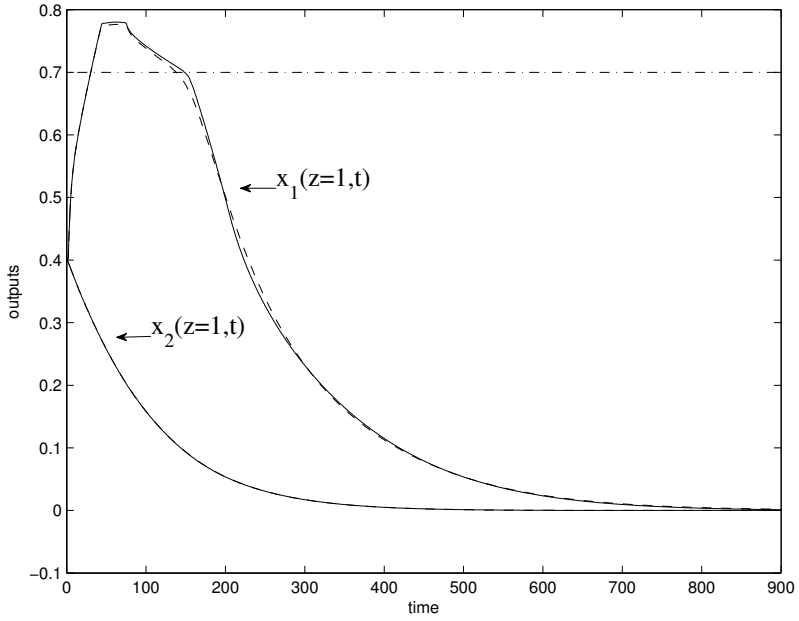
		Standard discrete		Tustin's discrete	
Plant model	Time step	Constraints	Stability	Constraints	Stability
Finite difference	1×10^{-5}	Satisfied	Stable	Satisfied	Stable
	$[1 \times 10^{-5}, 3 \times 10^{-3}]$	Satisfied	Stable	Satisfied	Stable
Galerkin	$[4 \times 10^{-3}, 6 \times 10^{-3}]$	Satisfied	Stable	Violated	Stable
	$[7 \times 10^{-3}, 1 \times 10^{-2}]$	Violated	Unstable	Violated	Unstable

Table 2.1: Performance of MPC laws constructed on standard and Tustin's discrete-time system when implemented on different continuous-time plant model

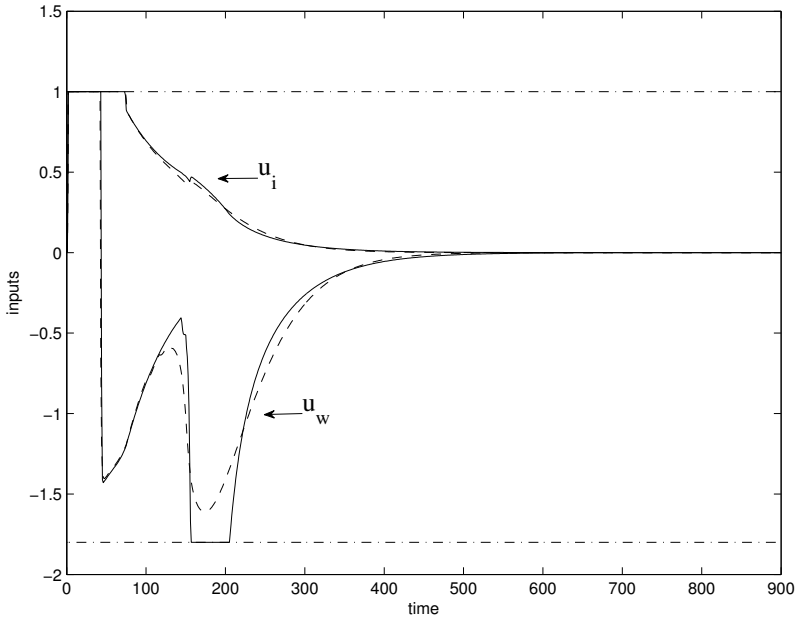
In Table 2.1, the MPC laws are built on the basis of discrete-time system obtained by standard discretization method and Tustin's discretization method, respectively. The discretization time step utilized in both methods is 1×10^{-3}. Then the MPC laws are implemented on the continuous-time plant model obtained by finite difference method and Galerkin's method, respectively. The time step for finite difference model is 1×10^{-5} while the time step for Galerkin model is varied from 1×10^{-5} to 1×10^{-2}. The constraints are satisfied all the time on finite difference model with the optimal control obtained by MPC laws constructed on either standard or Tustin's discrete-time system. When the continuous-time model obtained by Galerkin's method is utilized as the plant model, the performance of MPC laws depends on the value of the time step used for plant model approximation. When the time step of the plant model is between 1×10^{-5} and 3×10^{-3}, both the MPC

law built on standard discrete-time model and the MPC law built on Tustin's discrete-time model have a good performance. But when the time step is between 4×10^{-3} and 6×10^{-3}, only the MPC law built on standard discrete-time system can guarantee that the constraints will be satisfied. When the time step is equal or greater than 7×10^{-3}, both the MPC laws are numerically unstable. Therefore, the time step must be chosen carefully in order to avoid numeral instability. In summary, we conclude that it is better to choose a plant time step which is close to the discretization time step of controller. For control practitioners, a good choice is to choose the time step according to the internal clock.

In this work, the constrained optimization programs of the proposed predictive control laws are concerned with only slow dynamics in order to reduce computational effort. Table 2.2 gives the computation time of optimization programs which are concerned with low-order dynamics and full-order dynamics, respectively. The MATLAB subroutine QuadProg is used as the optimizer in all simulations. Constraints and MPC tuning parameters are kept identical in all optimization programs. The computation time of optimization program with objective function accounting for only slow dynamics (2^{nd}-order) is almost triple that of the optimization program with objective function accounting for full dynamics (15^{th}-order). The statement holds for both the predictive control law constructed on standard discrete-time model and that constructed on Tustin's discrete-time model. Although only 2 modes are taken into account for the low-order objective function, it makes a good control performance because the slow dynamics can capture the dominant dynamics of the PDEs system. Thus the benefit of the proposed model predictive control synthesis using low-order objective function and full-order constraints equation is that it makes a good control performance and reduces computational effort.

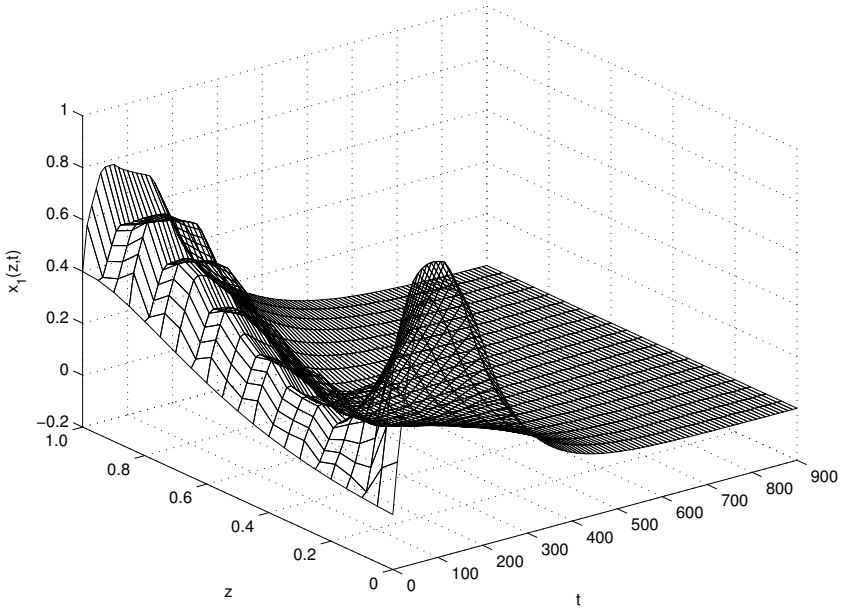


(a) States profile at $z = 1$ under the control law of Eqs. (2.71) and (2.72) (solid line); States profile at $z = 1$ under the control law of Eqs. (2.75) and (2.76) (dashed line); upper state constraint (dash-dot line).

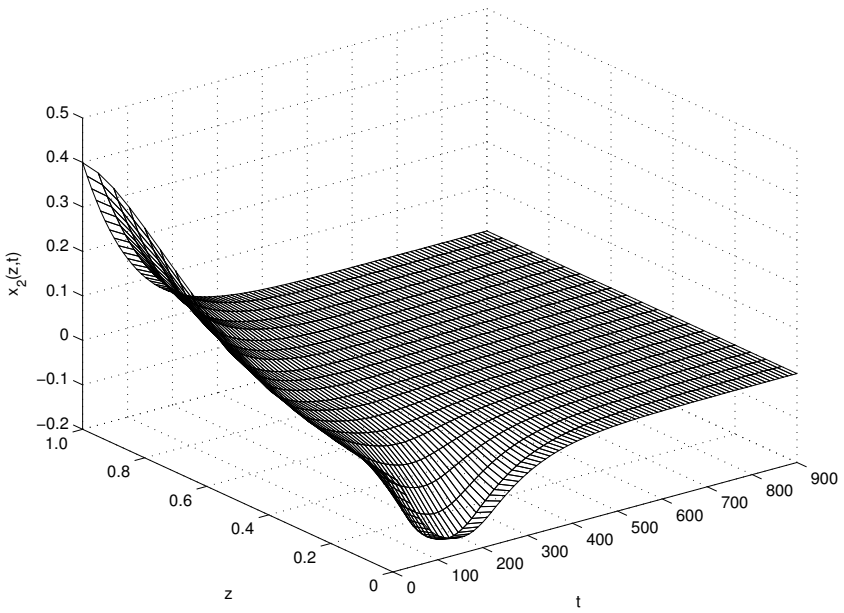


(b) Inputs profile computed by the control law of Eqs. (2.71) and (2.72) (solid line); Inputs profile computed by the control law of Eqs. (2.75) and (2.76) (dashed line); inputs constraints (dash-dot line).

Figure 2.2: Comparison between profiles of closed-loop system under the control law of Eqs. (2.71) and (2.72) constructed on standard discrete-time system with input and state constraints accounting for only slow modes and profiles of closed-loop system under the control law of Eqs. (2.75) and (2.76) constructed on Tustin's discrete-time system with input and state constraints accounting for only slow modes.

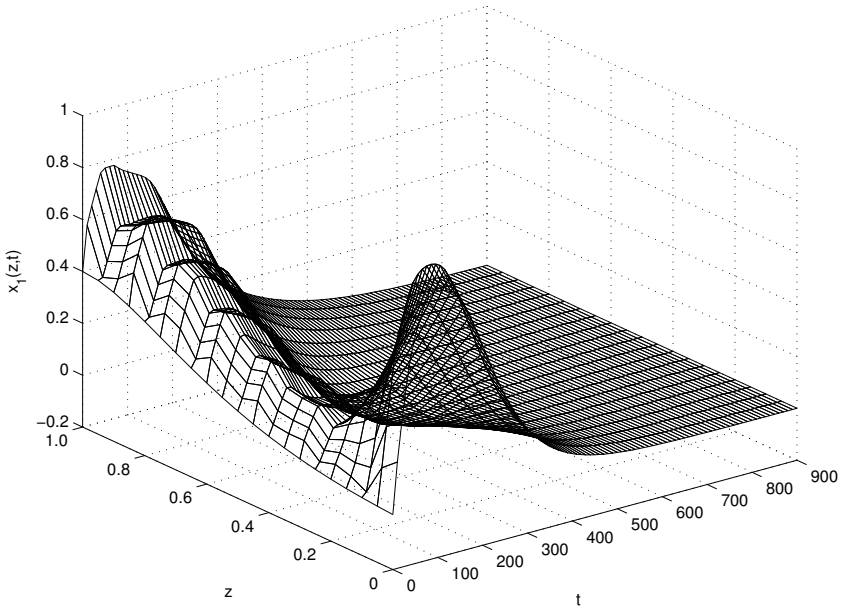


(a) Profile of $x_1(z,t)$ (temperature)

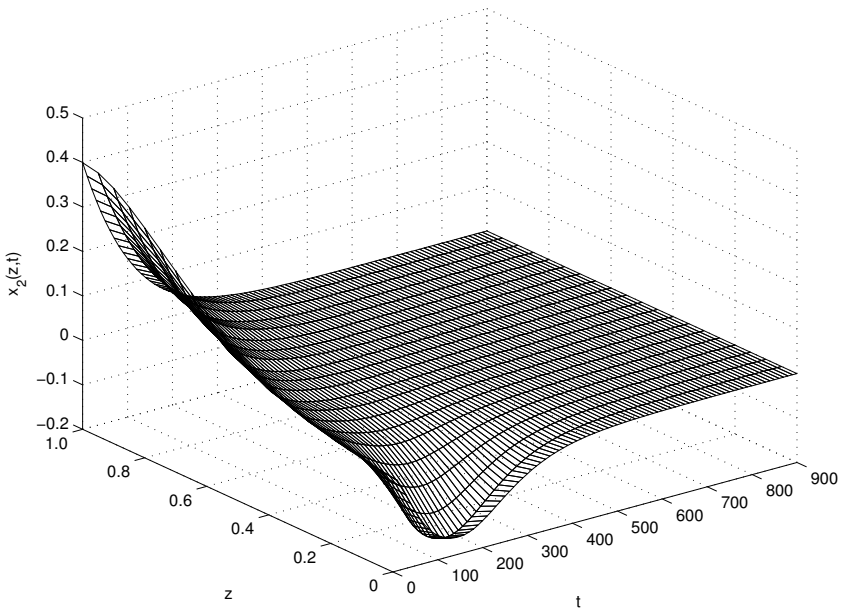


(b) Profile of $x_2(z,t)$ (concentration)

Figure 2.3: Closed-loop state profiles under the control law of Eqs. (2.71) and (2.72) constructed on standard discrete-time system with input and state constraints accounting for only slow modes.

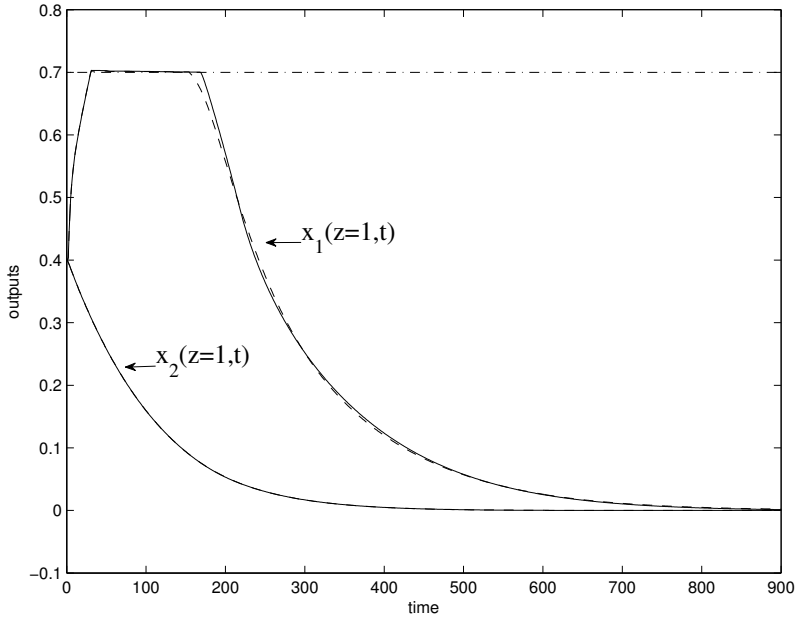


(a) Profile of $x_1(z,t)$ (temperature)

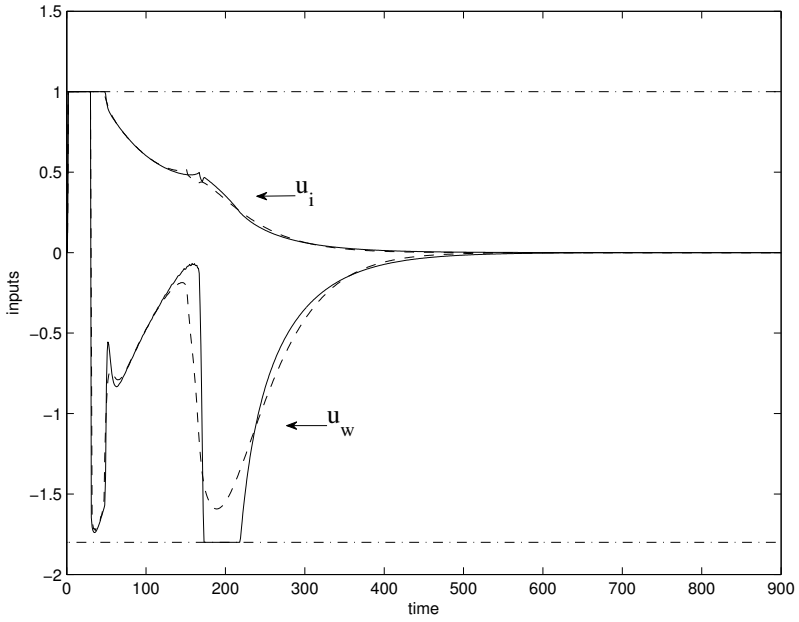


(b) Profile of $x_2(z,t)$ (concentration)

Figure 2.4: Closed-loop state profiles under the control law of Eqs. (2.75) and (2.76) constructed on Tustin's discrete-time system with input and state constraints accounting for only slow modes.

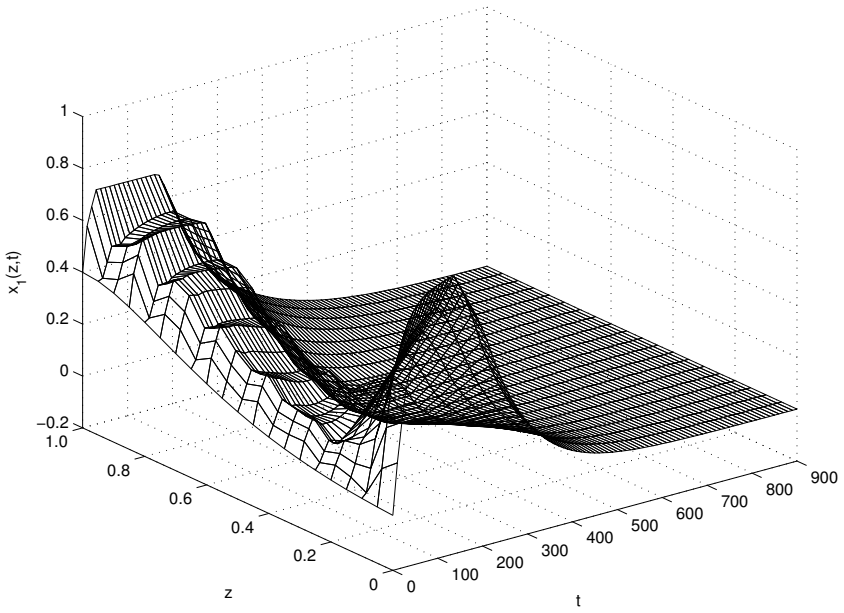


(a) States profile at $z = 1$ under the control law of Eqs. (2.73) and (2.74) (solid line); States profile at $z = 1$ under the control law of Eqs. (2.77) and (2.78) (dashed line); upper state constraint (dash-dot line).

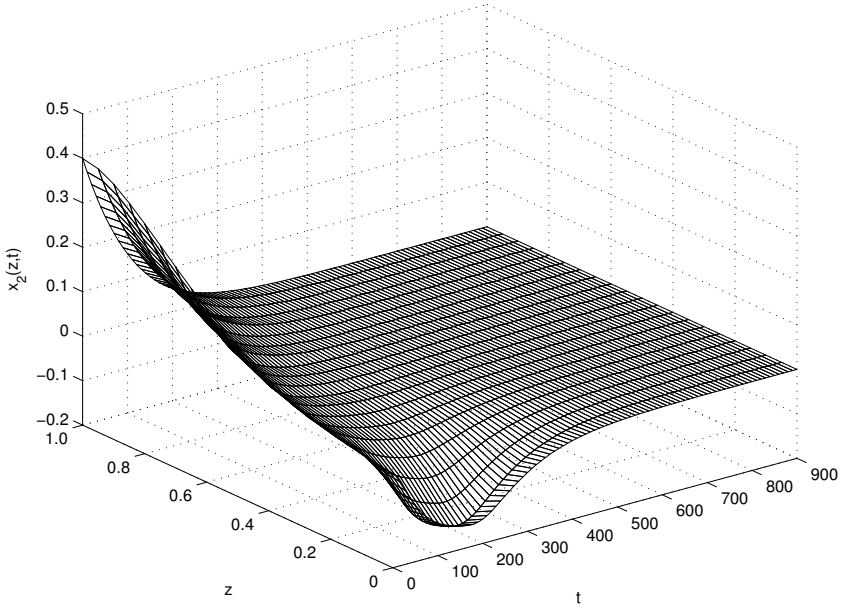


(b) Inputs profile computed by the control law of Eqs. (2.73) and (2.74) (solid line); Inputs profile computed by the control law of Eqs. (2.77) and (2.78) (dashed line); inputs constraints (dash-dot line).

Figure 2.5: Comparison between profiles of closed-loop system under the control law of Eqs. (2.73) and (2.74) constructed on standard discrete-time system with input and state constraints accounting for full modes and profiles of closed-loop system under the control law of Eqs. (2.77) and (2.78) constructed on Tustin's discrete-time system with input and state constraints accounting for full modes.

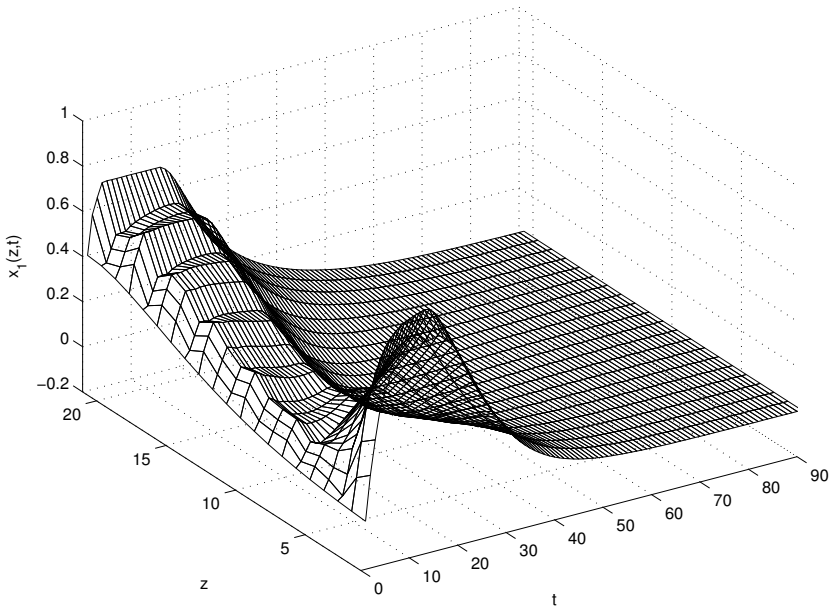


(a) Profile of $x_1(z,t)$ (temperature)

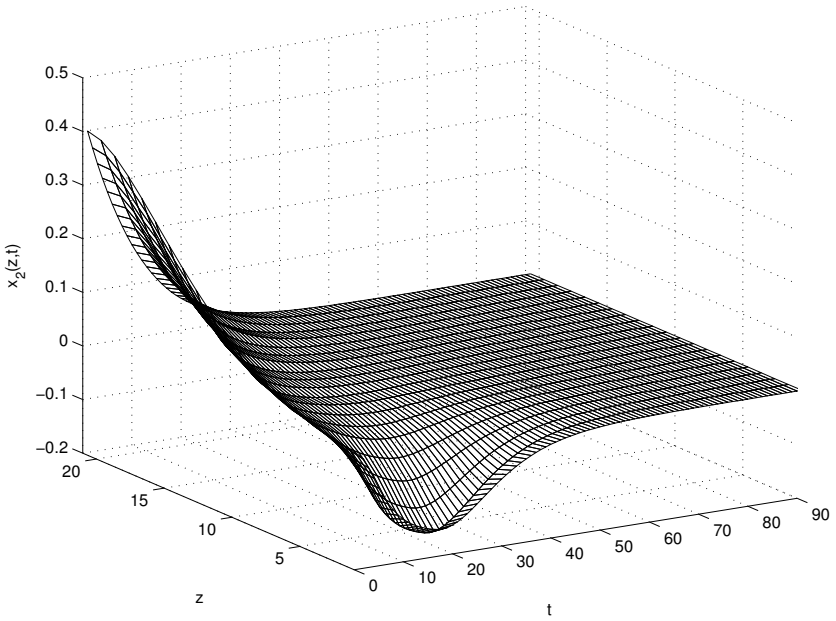


(b) Profile $x_2(z,t)$ (concentration)

Figure 2.6: Closed-loop state profiles under the control law of Eqs. (2.73) and (2.74) constructed on standard discrete-time system with input and state constraints accounting for full modes.



(a) Profile of $x_1(z,t)$ (temperature)



(b) Profile $x_2(z,t)$ (concentration)

Figure 2.7: Closed-loop state profiles under the control law of Eqs. (2.77) and (2.78) constructed on Tustin's discrete-time system with input and state constraints accounting for full modes.

Model used for MPC	Dynamics order	Computation time (seconds)
Standard discrete Galerkin	2^{nd}-order	154.18
	15^{th}-order	448.84
Tustin's discrete Galerkin	2^{nd}-order	147.01
	15^{th}-order	482.17

Table 2.2: Computation time

2.8 Conclusions

In this chapter, model predictive control algorithms are developed for the parabolic PDEs system with consideration of input and state constraints arising in the context of an axial dispersion chemical reactor process. We consider a dimensionless model described by nonlinear parabolic PDEs and solve an eigenvalue problem of the spatial operator which can be transformed into the Sturm-Liouville form. Galerkin's method and modal decomposition technique are adopted to obtain a finite-dimensional system that captures the dominant dynamics of the parabolic PDEs system, which is used as the basis for the low-order controller design subsequently. Two different discretization methods, standard and Tustin's discretization, are used to derive the discrete-time modal representations. The transfer function of the PDEs system is also derived in order to obtain The Tustin's discrete-time model with Cayley transform. Four model predictive control formulations, which differ in the way that the dynamics of fast modes are involved in the performance objective function and state constraints equation, are derived on the basis of finite-dimensional approximations. Finally, the MPC formulations are demonstrated and compared, via simulation studies, to achieve the control objectives. The performance of the predictive control laws are compared through the implementation of optimal control on different plant models. A comparison between computation time is also provided to demonstrate the benefit of using a low-order controller.

Chapter 3

Model Predictive Control of Euler-Bernoulli Beam with Structural Damping

3.1 Introduction

This chapter focuses on the development of model predictive control algorithm which accounts for the input and state constraints applied to the fourth-order partial differential equation (PDE) system describing the flexible Euler-Bernoulli beam system with structural damping. The eigenvalue problem of the fourth-order spatial operator is solved, and orthogonal basis are chosen for model decomposition according to the eigenvalue and eigenvector of the operator. The infinite-dimensional representation of the fourth-order PDE dynamics is approximated by a reduced-order finite-dimensional model through spectral decomposition technique. The continuous-time model representation is transformed into discrete-time equivalents by standard discretization method and Tustin's discretization method with Cayley transform, respectively. The transfer function of the beam system is also obtained to derive Tustin's discrete-time model representation. Two MPC formulations, in which the fast dynamics are only involved in the PDE state constraints , are developed on the basis of two discrete-time system representations. The effectiveness of the proposed MPC structure is successfully illustrated via simulation studies.

3.2 Euler-Bernoulli beam system

In this chapter, we consider a flexible Euler-Bernoulli beam of unit length, cross-sectional area a, mass density ρ and structural damping with damping coefficient ϵ. If we suppose the beam is simply supported, the out-of-plane displacement w of the beam is governed by the Euler-Bernoulli Beam Equation,

$$\rho a \frac{\partial^2 w}{\partial t^2} + \epsilon \frac{\partial^3 w}{\partial t \partial z^2} + EI \frac{\partial^4 w}{\partial z^4} = 0 \tag{3.1}$$

with the following boundary and initial conditions:

$$w(0,t) = 0 = w(1,t) = \frac{d^2 w(0,t)}{dz^2}$$

$$\frac{d^2 w(1,t)}{dz^2} = u(t) \tag{3.2}$$

$$w(z,0) = w_0(z) \tag{3.3}$$

where $w(z,t)$ is the vertical deflection of the beam at time t and at a distance z from one end, $0 \leq z \leq 1$; $u(t)$ is the control variable; E is the Young's modulus of elasticity, and I is the area moment of inertia of the beam's cross section.

For the sake of simplicity, the parameters $c = EI/(\rho a)$ and $\alpha = -\epsilon/(2\rho a)$ are introduced. Then the model in Eq. (3.1) can be rewritten in the form of:

$$\frac{\partial^2 w}{\partial t^2} - 2\alpha \frac{\partial^3 w}{\partial t \partial z^2} + c \frac{\partial^4 w}{\partial z^4} = 0 \tag{3.4}$$

with the following boundary and initial conditions:

$$w(0,t) = 0 = w(1,t) = \frac{d^2 w(0,t)}{dz^2}$$

$$\frac{d^2 w(1,t)}{dz^2} = u(t) \tag{3.5}$$

$$w(z,0) = w_0(z) \tag{3.6}$$

The Euler-Bernoulli equation is neither hyperbolic nor parabolic. Because the Euler-bernoulli equation does not have a finite speed of propagation, as do the hyperbolic equations, nor does it have smoother solution as time evolves, as do the parabolic equations [39]. Therefore, the method to solve the eigenvalue problem of the Euler-Bernoulli equation is different with those to solve hyperbolic or parabolic equations.

3.3 Eigenvalue problem

The boundary conditions in Eq. (3.2) is non-homogeneous because of the presence of $u(t)$. In order to transform the boundary control problem into a distributed control problem, we introduce a new state variable $v(z,t)$ which satisfies the corresponding homogeneous boundary conditions:

$$v(0,t) = 0 = v(1,t) = \frac{d^2v(0,t)}{dz^2} = \frac{d^2v(1,t)}{dz^2} \tag{3.7}$$

The relationship between $v(z,t)$ and $w(z,t)$ is:

$$w(z,t) = v(z,t) + b(z)u(t) \tag{3.8}$$

Substituting Eq. (3.8) into the boundary conditions in Eq. (3.2) and combining the results with Eq. (3.7), we can obtain $b(z)$ in the form of:

$$b(z) = \frac{1}{6}(z^3 - z) \tag{3.9}$$

and the following equivalent PDE model:

$$\frac{\partial^2 v}{\partial t^2} = -c\frac{\partial^4 v}{\partial z^4} + 2\alpha\frac{\partial^3 v}{\partial t \partial z^2} - b(z)\frac{d^2u(t)}{dt^2} + 2\alpha\frac{d^2b(z)}{dz^2}\frac{du(t)}{dt} \tag{3.10}$$

In order to formulate an abstract state space system on a Hilbert space, the term of second-order derivative respect to time must be transformed into a first-order derivative. To this

end, we introduce a state vector:

$$\begin{bmatrix} x_1(z,t) & x_2(z,t) \end{bmatrix}^T = \begin{bmatrix} v(z,t) & \frac{\partial w(z,t)}{\partial t} \end{bmatrix}^T$$

and a new control variable:

$$\tilde{u}(t) = \frac{du(t)}{dt}$$

Thus the system representation is transformed into the following equivalent:

$$\frac{\partial}{\partial t} \begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c\mathscr{A}_0 & 2\alpha\mathscr{A}_0^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} + \begin{bmatrix} -b(z) \\ 0 \end{bmatrix} \tilde{u}(t) \tag{3.11}$$

subject to

$$\begin{aligned} & x_1(0,t) = x_1(1,t) = 0 \\ & \frac{\partial^2 x_1(0,t)}{\partial z^2} = \frac{\partial^2 x_1(1,t)}{\partial z^2} = 0 \end{aligned} \tag{3.12}$$

where $\mathscr{A}_0$ is a positive, self adjoint operator on $L_2[0,1]$ defined as:

$$\mathscr{A}_0\phi(z) = \frac{d^4\phi(z)}{dz^4} \tag{3.13}$$

with its dense domain:

$$\begin{aligned} D(\mathscr{A}_0) = \{\phi(z) \in L_2[0,1] : & \frac{d\phi(z)}{dz}, \frac{d^2\phi(z)}{dz^2}, \frac{d^3\phi(z)}{dz^3}, \frac{d^4\phi(z)}{dz^4} \in L_2[0,1], \\ & \phi(z), \frac{d\phi(z)}{dz}, \frac{d^2\phi(z)}{dz^2}, \frac{d^3\phi(z)}{dz^3}, abs.\ cont., \\ & \phi(0) = \phi(1) = \frac{d^2\phi(0)}{dz^2} = \frac{d^2\phi(1)}{dz^2} = 0\} \end{aligned} \tag{3.14}$$

The operator $\mathscr{A}_0$ has the square root:

$$\mathscr{A}_0^{\frac{1}{2}}\phi(z) = \frac{d^2\phi(z)}{dz^2} \tag{3.15}$$

and its dense domain:

$$D(\mathscr{A}_0^{\frac{1}{2}}) = \{\phi(z) \in L_2[0,1] : \frac{d\phi(z)}{dz}, \frac{d^2\phi(z)}{dz^2} \in L_2[0,1],$$

$$\phi(z),\ \frac{d\phi(z)}{dz}, abs.\ cont.,$$

$$\phi(0) = \phi(1) = 0\} \tag{3.16}$$

We define the Hilbert space $Z = D(\mathscr{A}_0^{\frac{1}{2}}) \oplus L_2[0,1]$ with the inner product:

$$< \omega, \bar{\omega} >_Z = < \mathscr{A}_0^{\frac{1}{2}}\omega_1, \mathscr{A}_0^{\frac{1}{2}}\bar{\omega}_1 > + < \omega_2, \bar{\omega_2} > \tag{3.17}$$

where $\omega = \begin{bmatrix} \omega_1 \\ \\ \omega_2 \end{bmatrix}$ and $\bar{\omega} = \begin{bmatrix} \bar{\omega}_1 \\ \\ \bar{\omega}_2 \end{bmatrix}$, and $<,>$ is the usual inner product on $L_2[0,1]$.

The system operator associated with Eqs. (3.11) and (3.12) is now

$$\mathscr{A} = \begin{bmatrix} 0 & I \\ \\ -c\mathscr{A}_0 & 2\alpha\mathscr{A}_0^{\frac{1}{2}} \end{bmatrix} \tag{3.18}$$

with its dense domain:

$$D(\mathscr{A}) = D(\mathscr{A}_0) \oplus D(\mathscr{A}_0^{\frac{1}{2}})$$

The eigenvalue problem of the operator $\mathscr{A}$ is:

$$\mathscr{A} \begin{bmatrix} \phi_{1n}(z) \\ \\ \phi_{2n}(z) \end{bmatrix} = \lambda_n \begin{bmatrix} \phi_{1n}(z) \\ \\ \phi_{2n}(z) \end{bmatrix} \tag{3.19}$$

subject to

$$\phi_{1n}(0) = \phi_{1n}(1) = 0$$

$$\frac{d^2\phi_{1n}(0)}{dz^2} = \frac{d^2\phi_{1n}(1)}{dz^2} = 0 \tag{3.20}$$

for $n = 1, 2, \cdots, \infty$. λ_n denotes an eigenvalue; ϕ_{1n} and ϕ_{2n} denote the eigenfunctions associated with λ_n. To solve the eigenvalue problem of the operator $\mathscr{A}$ analytically, the determinant of $\mathscr{A} - \lambda_n I$ is set to zero.

$$0 = \left| \mathscr{A} - \lambda_n I \right|$$

$$
\begin{aligned}
&= \begin{vmatrix} -\lambda_n & I \\ -c\mathscr{A}_0 & 2\alpha\mathscr{A}_0^{\frac{1}{2}} - \lambda_n \end{vmatrix} \\
&= \lambda_n^2 I - 2\alpha\mathscr{A}_0^{\frac{1}{2}}\lambda_n + c\mathscr{A}_0 \\
&= \left(\mathscr{A}_0^{\frac{1}{2}} - \Big(\frac{\alpha\lambda_n}{c} + \frac{\sqrt{(\alpha^2-c)\lambda_n^2}}{c}\Big)I\right)\left(\mathscr{A}_0^{\frac{1}{2}} - \Big(\frac{\alpha\lambda_n}{c} - \frac{\sqrt{(\alpha^2-c)\lambda_n^2}}{c}\Big)I\right) \qquad (3.21)
\end{aligned}
$$

Therefore,

$$
\mathscr{A}_0^{\frac{1}{2}} - \Big(\frac{\alpha\lambda_n}{c} + \frac{\sqrt{(\alpha^2-c)\lambda_n^2}}{c}\Big)I = 0 \quad \text{or} \quad \mathscr{A}_0^{\frac{1}{2}} - \Big(\frac{\alpha\lambda_n}{c} - \frac{\sqrt{(\alpha^2-c)\lambda_n^2}}{c}\Big)I = 0
$$

Assume the eigenvalue of $\mathscr{A}_0^{\frac{1}{2}}$ is denoted by $\bar{\lambda}_n$, i.e. $\mathscr{A}_0^{\frac{1}{2}}\phi_n(z) = \bar{\lambda}_n\phi_n(z)$. We can easily get the spectrum of $\mathscr{A}_0^{\frac{1}{2}}$ which is $\bar{\lambda}_n = -n^2\pi^2$. Replacing $\mathscr{A}_0^{\frac{1}{2}}$ with $\bar{\lambda}_n$ in Eq. (3.21), we can obtain

$$
\lambda_n = \alpha\bar{\lambda}_n \pm \sqrt{(\alpha^2-c)\bar{\lambda}_n^2}
$$

Thus, the eigenvalues of operator $\mathscr{A}$ is

$$
\lambda_n = \begin{cases} -n^2\pi^2\big(\alpha \pm j\sqrt{c-\alpha^2}\,\big) & \text{if } \alpha^2 \le c \\ -n^2\pi^2\big(\alpha \pm \sqrt{\alpha^2-c}\,\big) & \text{if } \alpha^2 \ge c \end{cases} \qquad (3.22)
$$

Remark 3 *Usually we take $\alpha^2 \le c$ to avoid the linear growth of the wave, even though this growth (formally) does not affect the stability of the scheme.*

Therefore, in this chapter we consider the scenario where $\alpha^2 \le c$, i.e.,

$$
\lambda_n = -n^2\pi^2\big(\alpha \pm j\sqrt{c-\alpha^2}\,\big) \qquad (3.23)
$$

We assume the eigenvectors of the operator $\mathscr{A}$ is in the form of

$$
\Phi_n(z) = \begin{bmatrix} \phi_{1n}(z) \\ \phi_{2n}(z) \end{bmatrix}
$$

then the eigenvalue problem of operator $\mathscr{A}$

$$\mathscr{A}\Phi_n(z) = \lambda_n \Phi_n(z)$$

implies that

$$\begin{bmatrix} 0 & I \\ -\mathscr{A}_0 & 2\alpha\mathscr{A}_0^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \phi_{1n}(z) \\ \phi_{2n}(z) \end{bmatrix} = \lambda_n \begin{bmatrix} \phi_{1n}(z) \\ \phi_{2n}(z) \end{bmatrix} \tag{3.24}$$

Thus the eigenvalue problem is represented in the following form:

$$\begin{aligned} &\phi_{2n} = \lambda_n \phi_{1n} \\ &\frac{d^4\phi_{1n}(z)}{dz^4} - 2\alpha\lambda_n \frac{d^2\phi_{1n}(z)}{dz^2} + \lambda_n^2 \phi_{1n}(z) = 0 \end{aligned} \tag{3.25}$$

subject to the boundary conditions:

$$\begin{aligned} &\phi_{1n}(0) = \phi_{1n}(1) = 0 \\ &\frac{d^2\phi_{1n}(0)}{dz^2} = \frac{d^2\phi_{1n}(1)}{dz^2} = 0 \end{aligned} \tag{3.26}$$

We can get the solution to $\Phi_n(z)$, which is

$$\Phi_n(z) = \begin{bmatrix} sin(n\pi z) \\ \lambda_n sin(n\pi z) \end{bmatrix} \tag{3.27}$$

3.4 Modal decomposition

Since the eigenvector $\Phi_n(z)$ is two-dimensional and is nonorthogonal, there is a need to choose two orthonormal basis $\{e_n, f_n\}$ according to the eigenvector $\Phi_n(z)$, which is given as [40]:

$$e_n(z) = \frac{\sqrt{2}}{(n\pi)^2} \begin{bmatrix} sin(n\pi z) \\ 0 \end{bmatrix}$$

$$f_n(z) = \sqrt{2} \begin{bmatrix} 0 \\ sin(n\pi z) \end{bmatrix} \tag{3.28}$$

and $\{e_n, f_n\}$ satisfies the following orthonormal conditions:

$$< e_n, e_n >_Z = 1, \quad < f_n, f_n >_Z = 1, \quad < e_n, f_n >_Z = 0$$

where $<,>$ is the inner product on the Hilbert space Z.

Now we assume the solutions to x_1 and x_2 are in the form of:

$$\begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{\infty} a_{1n}(t)\psi_{1n}(z) \\ \sum_{n=1}^{\infty} a_{2n}(t)\psi_{2n}(z) \end{bmatrix} \tag{3.29}$$

where,

$$\psi_{1n}(z) = \frac{\sqrt{2}}{(n\pi)^2} sin(n\pi z)$$

$$\psi_{2n}(z) = \sqrt{2} sin(n\pi z) \tag{3.30}$$

are the nonzero elements of the basis $e_n(z)$ and $f_n(z)$, respectively. We also define the adjoint functions of $\psi^*_{1n}(z)$ and $\psi^*_{2n}(z)$ as following:

$$\psi^*_{1n}(z) = \sqrt{2}(n\pi)^2 sin(n\pi z)$$

$$\psi^*_{2n}(z) = \sqrt{2} sin(n\pi z) \tag{3.31}$$

which satisfies

$$< \psi_{1m}(z), \psi^*_{1n} >= \delta_{mn}$$

and

$$< \psi_{2m}(z), \psi^*_{2n} >= \delta_{mn}$$

where δ_{mn} is Kronecker delta function.

Now $x_1(z,t)$ and $x_2(z,t)$ can be represented in the following form:

$$\begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} = \begin{bmatrix} \psi_{11} & 0 & \psi_{12} & 0 & \cdots & \psi_{1n} & 0 & \cdots \\ 0 & \psi_{21} & 0 & \psi_{22} & \cdots & 0 & \psi_{2n} & \cdots \end{bmatrix} \begin{bmatrix} a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} \tag{3.32}$$

Also $b(z)$ can be decomposed with respect to the basis e_n

$$b(z) = \sum_{n=1}^{\infty} \beta_n \psi_{1n}(z)$$

The matrix

$$\begin{bmatrix} e_n & f_n \end{bmatrix} = \begin{bmatrix} \psi_{1n} & 0 \\ 0 & \psi_{2n} \end{bmatrix}$$

will be referred to as the transfer matrix T_n. Then we do state transformation and obtain the following model representation:

$$\frac{d}{dt} \begin{bmatrix} a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} = \begin{bmatrix} \Delta_1 & & & & & \\ & \Delta_2 & & & & \\ & & \ddots & & & \\ & & & \Delta_n & & \\ & & & & \ddots & \end{bmatrix} \begin{bmatrix} a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} + \begin{bmatrix} -\beta_1 \\ 0 \\ -\beta_2 \\ 0 \\ \vdots \\ -\beta_n \\ 0 \\ \vdots \end{bmatrix} \tilde{u}(t) \tag{3.33}$$

where,

$$\Delta_n = T_n^{-1} \mathscr{A} T_n = \begin{bmatrix} 0 & n^2\pi^2 \\ -cn^2\pi^2 & -2\alpha n^2\pi^2 \end{bmatrix}, \quad n = 1, 2, ..., \infty \tag{3.34}$$

and

$$\begin{aligned} \beta_n &= < b(z), \psi_{1n}^*(z) > \\ &= \int_0^1 \frac{z^3 - z}{6} \sqrt{2}(n\pi)^2 sin(n\pi z) dz \\ &= \sqrt{2}(-1)^n/(n\pi) \end{aligned} \tag{3.35}$$

For the sake of simplicity for controller design in the following sections, the control variable $u(t)$ is incorporated into the state variables. Therefore, the model representation of the system can be written as:

$$\frac{d}{dt}\begin{bmatrix} u(t) \\ a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & & & & & & \\ & \Delta_1 & & & & & \\ & & \Delta_2 & & & & \\ & & & \ddots & & & \\ & & & & \Delta_n & & \\ & & & & & \ddots & \end{bmatrix} \begin{bmatrix} u(t) \\ a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} + \begin{bmatrix} 1 \\ -\beta_1 \\ 0 \\ -\beta_2 \\ 0 \\ \vdots \\ -\beta_n \\ 0 \\ \vdots \end{bmatrix} \tilde{u}(t) \tag{3.36}$$

A measurement is set at the boundary of the beam, which is described as:

$$\begin{aligned} y(t) &= \left.\frac{\partial w(z,t)}{\partial z}\right|_{z=1} \\ &= \left.\frac{\partial}{\partial z}\Big(x_1(z,t) + b(z)u(t)\Big)\right|_{z=1} \\ &= \sum_{n=1}^{\infty} a_{1n}(t) \left.\frac{\partial \psi_{1n}(z)}{\partial z}\right|_{z=1} + \left.\frac{\partial b(z)}{\partial z}\right|_{z=1} u(t) \end{aligned}$$

$$= \begin{bmatrix} \frac{\partial b(z)}{\partial z}\Big|_{z=1} & \frac{\partial \psi_{11}(z)}{\partial z}\Big|_{z=1} & 0 & \frac{\partial \psi_{12}(z)}{\partial z}\Big|_{z=1} & 0 & \cdots & \frac{\partial \psi_{1n}(z)}{\partial z}\Big|_{z=1} & 0 & \cdots \end{bmatrix} \begin{bmatrix} u(t) \\ a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{3} & \beta_1 & 0 & \beta_2 & 0 & \cdots & \beta_n & 0 & \cdots \end{bmatrix} \begin{bmatrix} u(t) \\ a_{11}(t) \\ a_{21}(t) \\ a_{12}(t) \\ a_{22}(t) \\ \vdots \\ a_{1n}(t) \\ a_{2n}(t) \\ \vdots \end{bmatrix} \tag{3.37}$$

Thus we obtain the following infinite dimensional state space representation:

$$\begin{bmatrix} \dot{u}(t) \\ \dot{a}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} u(t) \\ a(t) \end{bmatrix} + \begin{bmatrix} 1 \\ B \end{bmatrix} \tilde{u}(t)$$

$$y(t) = \begin{bmatrix} \frac{1}{3} & C \end{bmatrix} \begin{bmatrix} u(t) \\ a(t) \end{bmatrix} \tag{3.38}$$

where,

$$
\begin{aligned}
a(t) &= \begin{bmatrix} a_{11}(t) & a_{21}(t) & a_{12}(t) & a_{22}(t) & \cdots & a_{1n}(t) & a_{2n}(t) & \cdots \end{bmatrix}^T \\
A &= \begin{bmatrix}
0 & \pi^2 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
-c\pi^2 & -2\alpha\pi^2 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & (2\pi)^2 & \cdots & 0 & 0 & \cdots \\
0 & 0 & -c(2\pi)^2 & -2\alpha(2\pi)^2 & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & 0 & (n\pi)^2 & \cdots \\
0 & 0 & 0 & 0 & \cdots & -c(n\pi)^2 & -2\alpha(n\pi)^2 & \cdots \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots
\end{bmatrix} \\
B &= \begin{bmatrix} -\frac{\sqrt{2}(-1)^1}{\pi} & 0 & -\frac{\sqrt{2}(-1)^2}{2\pi} & 0 & \cdots & -\frac{\sqrt{2}(-1)^n}{n\pi} & 0 & \cdots \end{bmatrix}^T \\
C &= \begin{bmatrix} \frac{\sqrt{2}(-1)^1}{\pi} & 0 & \frac{\sqrt{2}(-1)^2}{2\pi} & 0 & \cdots & \frac{\sqrt{2}(-1)^n}{n\pi} & 0 & \cdots \end{bmatrix}
\end{aligned}
\tag{3.39}
$$

Remark 4 *Now the continuous-time infinite-dimensional model representation of the flexible beam system is obtained. Since the first few eigenvalues and eigenvectors capture the dominant dynamic characteristic of the PDE system, the MPC controller will be constructed mainly on the basis of the first few modes. Therefore, appropriate model representation must be adopted to decouple the first few modes and the other modes.*

To decouple the slow and fast dynamics of the system, the technique of modal decomposition is applied on the infinite-dimensional system representation of Eq. (3.38) in order to obtain an approximate finite-dimensional system representation. We define two spectral projection operators $\mathcal{P}_s$ and $\mathcal{P}_f$ such that the Hilbert space $\mathcal{H}$ can be partitioned into two subspaces, $\mathcal{H}_s$ and $\mathcal{H}_f$. $\mathcal{H}_s$ is defined as $\mathcal{H}_s = span\{\Phi_1(z), \Phi_2(z), \cdots, \Phi_m(z)\}$ and $\mathcal{H}_f$ is defined as $\mathcal{H}_f = \{\Phi_{m+1}(z), \Phi_{m+2}(z), \cdots\}$ (m is an arbitrary positive integer greater than 1 because all

$\lambda_n < 0$ for this system). Thus, the eigen-coefficients $a(t)$ in Eq. (3.38) can be decomposed as:

$$a_s(t) = \mathcal{P}_s a(t) = \begin{bmatrix} a_{11}(t) & a_{21}(t) & a_{12}(t) & a_{22}(t) & \cdots & a_{1m}(t) & a_{2m}(t) \end{bmatrix}^T$$
$$a_f(t) = \mathcal{P}_f a(t) = \begin{bmatrix} a_{1(m+1)}(t) & a_{2(m+1)}(t) & a_{1(m+2)}(t) & a_{2(m+2)}(t) & \cdots \end{bmatrix}^T \tag{3.40}$$

The two subsystems, $a_s(t)-$ and $a_f(t)-$, are referred to as slow subsystem and fast subsystem, respectively. The slow subsystem is a finite-dimensional subsystem and it captures the dominant dynamic characteristic of the PDE system, while the fast subsystem is an infinite-dimensional subsystem. With the above definition of slow and fast subsystems, the continuous-time system representation in Eq. (3.38) can be rewritten as:

$$\begin{bmatrix} \dot{u}(t) \\ \dot{a}_s(t) \\ \dot{a}_f(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & A_s & 0 \\ 0 & 0 & A_f \end{bmatrix} \begin{bmatrix} u(t) \\ a_s(t) \\ a_f(t) \end{bmatrix} + \begin{bmatrix} 1 \\ B_s \\ B_f \end{bmatrix} \tilde{u}(t)$$
$$y = \begin{bmatrix} \frac{1}{3} & C_s & C_f \end{bmatrix} \begin{bmatrix} u(t) \\ a_s(t) \\ a_f(t) \end{bmatrix} \tag{3.41}$$

For the sake of simplicity, we incorporate the control variable u into the slow subsystem such that the model representation in Eq. (3.41) becomes:

$$\begin{bmatrix} \dot{z}_s(t) \\ \dot{z}_f(t) \end{bmatrix} = \begin{bmatrix} \tilde{A}_s & 0 \\ 0 & \tilde{A}_f \end{bmatrix} \begin{bmatrix} z_s(t) \\ z_f(t) \end{bmatrix} + \begin{bmatrix} \tilde{B}_s \\ \tilde{B}_f \end{bmatrix} \tilde{u}(t)$$
$$y = \begin{bmatrix} \tilde{C}_s & \tilde{C}_f \end{bmatrix} \begin{bmatrix} z_s(t) \\ z_f(t) \end{bmatrix} \tag{3.42}$$

where,

$$z_s(t) = \begin{bmatrix} u(t) \\ a_s(t) \end{bmatrix}, \qquad z_f(t) = a_f(t)$$

$$\tilde{A}_s = \begin{bmatrix} 0 & 0 \\ 0 & As \end{bmatrix}, \qquad \tilde{A}_f = A_f$$

$$\tilde{B}_s = \begin{bmatrix} 1 \\ B_s \end{bmatrix}, \qquad \tilde{B}_f = B_f$$

$$\tilde{C}_s = \begin{bmatrix} \frac{1}{3} & C_s \end{bmatrix}, \qquad \tilde{C}_f = C_f$$

3.5 Discrete-time system representation

The continuous-time slow and fast subsystem representations are obtained, however, a discrete-time controller is more acceptable to control practitioners because it is not only simple to be derived but also convenient to be implemented. Since most of the control systems used in industry are embedded control systems, it is quite common to implement discrete-time control strategies with just a digital-to-analog converter. Thus the discretization of a model before designing a controller is necessary. Therefore, the model representation in Eq. (3.42) needs to be transformed into an equivalent discrete-time representation in order to be utilized for MPC construction. Appropriate discrete-time model representation must be adopted to decouple slow dynamic and fast dynamics.

Since the continuous-time system representation in Eq. (3.42) is infinite-dimensional, there are two issues arising from transforming it to a discrete-time counterpart. The first issue is how to discretize an infinite-dimensional system, and the second issue is what is the appropriate degree of approximation applied on the discrete-time model in order to obtain a suitable model for the MPC realization. In order to solve the above two issues, two discretization methods are explored in the ensuing section.

3.5.1 Standard discretization

For standard discretization, the infinite-dimensional fast subsystem is replaced by a $2l$-dimensional approximation. Then the slow subsystem and the $2l$-dimensional approximation of the fast subsystem are transformed into an appropriate discrete-time equivalent:

$$\begin{bmatrix} z_s(k+1) \\ z_f(k+1) \end{bmatrix} = \begin{bmatrix} \tilde{A}_{ds} & 0 \\ 0 & \tilde{A}_{df} \end{bmatrix} \begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix} + \begin{bmatrix} \tilde{B}_{ds} \\ \tilde{B}_{df} \end{bmatrix} \tilde{u}(k)$$

$$y(k) = \begin{bmatrix} \tilde{C}_{ds} & \tilde{C}_{df} \end{bmatrix} \begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix} \tag{3.43}$$

where, $\tilde{A}_{ds}$, $\tilde{B}_{ds}$, $\tilde{C}_{ds}$, $\tilde{A}_{df}$, $\tilde{B}_{df}$, $\tilde{C}_{df}$ are calculated in the following way:

$$\begin{bmatrix} \tilde{A}_{ds} & \tilde{B}_{ds} \\ \tilde{C}_{ds} & 0 \end{bmatrix} = \begin{bmatrix} e^{\tilde{A}_s \Delta t} & \int_0^{\Delta t} e^{\tilde{A}_s \tau} d\tau \tilde{B}_s \\ \tilde{C}_s & 0 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{A}_{df} & \tilde{B}_{df} \\ \tilde{C}_{df} & 0 \end{bmatrix} = \begin{bmatrix} e^{\tilde{A}_f \Delta t} & \int_0^{\Delta t} e^{\tilde{A}_{f2l} \tau} d\tau \tilde{B}_{f2l} \\ \tilde{C}_{f2l} & 0 \end{bmatrix} \tag{3.44}$$

where Δt denotes the discretization time interval; $\tilde{A}_{f2l}$, $\tilde{B}_{f2l}$, and $\tilde{C}_{f2l}$ are the $2l^{th}$-order approximations of matrices $\tilde{A}_f$, $\tilde{B}_f$, and $\tilde{C}_f$ in Eq. (3.42).

Since the slow and fast dynamics are not coupled with each other, one may obtain the following equivalent discrete-time system representation:

$$\begin{aligned} z_s(k+1) &= \tilde{A}_{ds} z_s(k) + \tilde{B}_{ds} \tilde{u}(k) \\ z_f(k+1) &= \tilde{A}_{df} z_f(k) + \tilde{B}_{df} \tilde{u}(k) \\ y(k) &= \tilde{C}_{ds} z_s(k) + \tilde{C}_{df} z_f(k) \end{aligned} \tag{3.45}$$

where $\tilde{A}_{ds}$, $\tilde{A}_{df}$, $\tilde{B}_{ds}$, $\tilde{B}_{df}$, $\tilde{C}_{ds}$, $\tilde{C}_{df}$ are $(2m \times 2m)$, $(2l \times 2l)$, $(2m \times 1)$, $(2l \times 1)$, $(1 \times 2m)$, and $(1 \times 2l)$ matrices, respectively.

3.5.2 Discretization by Cayley Transform

In the second discretization method, the infinite-dimensional system of Eq. (3.42) is directly transformed to the following discrete-time equivalent using Tustin's approximation and Cayley transform:

$$\begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix} = \begin{bmatrix} \bar{A}_{ds} & 0 \\ 0 & \bar{A}_{df} \end{bmatrix} \begin{bmatrix} z_s(k-1) \\ z_f(k-1) \end{bmatrix} + \begin{bmatrix} \bar{B}_{ds} \\ \bar{B}_{df} \end{bmatrix} \tilde{u}(k)$$

$$y(k) = \begin{bmatrix} \bar{C}_{ds} & \bar{C}_{df} \end{bmatrix} \begin{bmatrix} z_s(k-1) \\ z_f(k-1) \end{bmatrix} + \bar{D}_d \tilde{u}(k) \tag{3.46}$$

where,

$$\begin{bmatrix} \bar{A}_{ds} & 0 \\ 0 & \bar{A}_{df} \end{bmatrix} = \bar{A}_d, \quad \begin{bmatrix} \bar{B}_{ds} \\ \bar{B}_{df} \end{bmatrix} = \bar{B}_d, \quad \begin{bmatrix} \bar{C}_{ds} & \bar{C}_{df} \end{bmatrix} = \bar{C}_d$$

and $\bar{A}_d$, $\bar{B}_d$, $\bar{C}_d$, and $\bar{D}_d$ are given by the Cayley transform of the infinite-dimensional system:

$$\begin{bmatrix} \bar{A}_d & \bar{B}_d \\ \bar{C}_d & \bar{D}_d \end{bmatrix} = \begin{bmatrix} (\delta + A)(\delta - A)^{-1} & \sqrt{2\delta}(\delta - A_{-1})^{-1}B \\ \sqrt{2\delta}C(\delta - A)^{-1} & \mathcal{G}(\delta) \end{bmatrix} \tag{3.47}$$

where $\delta = 2/\Delta t$ and $\Delta t > 0$ denotes the discretization time interval; A_{-1} is the Yosida extension of A; $\mathcal{G}(\delta)$ is defined as the transfer function of the PDE system evaluated at δ.

The unbounded operators A, B and C of the continuous-time system are mapped into bounded operators $\bar{A}_d$, $\bar{B}_d$ and $\bar{C}_d$ in the discrete-time counterpart through Cayley transform. And it turns out that control properties, such as controllability, are the same for both systems [36].

The expression of the transfer function $\mathcal{G}(\delta)$ can be obtained in the following way:

$\mathcal{G}(s) = C(sI - A)^{-1}B$

$$
\begin{aligned}
&= \begin{bmatrix} \frac{1}{3} & \beta_1 & 0 & \beta_2 & 0 & \cdots & \beta_n & 0 & \cdots \end{bmatrix}
\begin{bmatrix}
s^{-1} & & & & & & & & \\
& (sI-\Delta_1)^{-1} & & & & & & & \\
& & (sI-\Delta_2)^{-1} & & & & & & \\
& & & \ddots & & & & & \\
& & & & (sI-\Delta_n)^{-1} & & & & \\
& & & & & \ddots & & &
\end{bmatrix}
\begin{bmatrix} 1 \\ -\beta_1 \\ 0 \\ -\beta_2 \\ 0 \\ \vdots \\ -\beta_n \\ 0 \\ \vdots \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{3} & \beta_1 & 0 & \beta_2 & 0 & \cdots & \beta_n & 0 & \cdots \end{bmatrix}
\begin{bmatrix}
\frac{1}{s} & & & & & & & & \\
& \frac{s+2\alpha\pi^2}{|sI-\Delta_1|} & \frac{\pi^2}{|sI-\Delta_1|} & & & & & & \\
& \frac{-\pi^2}{|sI-\Delta_1|} & \frac{s}{|sI-\Delta_1|} & & & & & & \\
& & & \frac{s+8\alpha\pi^2}{|sI-\Delta_2|} & \frac{4\pi^2}{|sI-\Delta_2|} & & & & \\
& & & \frac{-4\pi^2}{|sI-\Delta_2|} & \frac{s}{|sI-\Delta_2|} & & & & \\
& & & & & \ddots & & & \\
& & & & & & \frac{s+2\alpha n^2\pi^2}{|sI-\Delta_n|} & \frac{n^2\pi^2}{|sI-\Delta_n|} & \\
& & & & & & \frac{-n^2\pi^2}{|sI-\Delta_n|} & \frac{s}{|sI-\Delta_n|} & \\
& & & & & & & & \ddots
\end{bmatrix}
\begin{bmatrix} 1 \\ -\beta_1 \\ 0 \\ -\beta_2 \\ 0 \\ \vdots \\ -\beta_n \\ 0 \\ \vdots \end{bmatrix} \\
&= \frac{1}{3s} - \beta_1^2 \frac{s+2\alpha\pi^2}{|sI-\Delta_1|} - \beta_2^2 \frac{s+8\alpha\pi^2}{|sI-\Delta_2|} - \cdots - \beta_n^2 \frac{s+2\alpha n^2\pi^2}{|sI-\Delta_n|} - \cdots \\
&= \frac{1}{3s} - \sum_{n=1}^{\infty} \beta_n^2 \frac{s+2\alpha n^2\pi^2}{|sI-\Delta_n|} \\
&= \frac{1}{3s} - \sum_{n=1}^{\infty} \frac{2(s+2\alpha n^2\pi^2)}{n^2\pi^2(s^2+2\alpha n^2\pi^2 s+n^4\pi^4)} \\
&= \sum_{n=1}^{\infty} \frac{2n^2\pi^2}{s(s^2+2\alpha n^2\pi^2 s+n^4\pi^4)}
\end{aligned} \tag{3.48}
$$

Since in Eq. (3.46) the slow and fast dynamics are not coupled with each other, the discrete-time system representation can be rewritten as the following:

$$
\begin{aligned}
z_s(k) &= \bar{A}_{ds} z_s(k-1) + \bar{B}_{ds} \tilde{u}(k) \\
z_f(k) &= \bar{A}_{df} z_f(k-1) + \bar{B}_{df} \bar{u}(k) \\
y(k) &= \bar{C}_{ds} z_s(k-1) + \bar{C}_{df} z_f(k-1) + \bar{D}_d \tilde{u}(k)
\end{aligned} \tag{3.49}
$$

Now two discrete-time approximations of the system representation in Eq. (3.38) are obtained and given by Eq. (3.45) and Eq. (3.49), respectively. The major difference between these two representations is that the control variable contributes directly to the output variable in the second representation, whereas in the first representation the output variable is

only contributed by the state variable.

3.6 Modal model predictive control

In this section, modal model predictive control formulations are constructed on the basis of those two linear time-invariant discrete-time model dynamics which are developed in previous sections. The two discrete-time system representations are given in the standard exact discretization form:

$$\begin{cases} z_s(k+1) = \tilde{A}_{ds} z_s(k) + \tilde{B}_{ds} \tilde{u}(k) \\ z_f(k+1) = \tilde{A}_{df} z_f(k) + \tilde{B}_{df} \tilde{u}(k) \\ y(k) = \tilde{C}_{ds} z_s(k) + \tilde{C}_{df} z_f(k) \end{cases} \tag{3.50}$$

and the Tustin's discretization form:

$$\begin{cases} z_s(k) = \bar{A}_{ds} z_s(k-1) + \bar{B}_{ds} \tilde{u}(k) \\ z_f(k) = \bar{A}_{df} z_f(k-1) + \bar{B}_{df} \bar{u}(k) \\ y(k) = \bar{C}_{ds} z_s(k-1) + \bar{C}_{df} z_f(k-1) + \bar{D}_d \tilde{u}(k) \end{cases} \tag{3.51}$$

The regulator is constructed as the solution to an optimization problem such that the following open-loop performance objective function on an infinite horizon is minimized at the sampling time k:

$$\min_{u^N} \sum_{j=0}^{\infty} \Big[y(k+j|k)^T Q y(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big] \tag{3.52}$$

where Q is a symmetric positive semidefinite penalty matrix, and R is a symmetric positive definite matrix. $y(k+j|k)$ and $u(k+j|k)$ represent the output and input variable at future time $k+j$ predicted at current time k. The vector u^N contains the sequences $u(k|k), u(k+1|k), ..., u(k+N-1|k)$ in which the first element $u(k|k)$ is the future control action to be injected to the plant. At time $k+N$, the input vector $u(k+j|k)$ is set to

zero and kept at zero for all $j \geq N$ in the calculation of the open-loop quadratic objective function value [37].

First, we apply the receding horizon regulator of Eq. (3.52) on the discrete-time dynamic model of Eq. (3.50) which is developed by the standard discretization method.

Remark 5 *It is proved in [41] that the MPC formulation constructed only by slow dynamics cannot guarantee the PDE state constraints because the fast dynamics are not involved either in the objective function or in the PDE state constraints equation. Therefore, in this case the contribution of fast dynamics are accounted for by the MPC algorithm in order to satisfy the requirement of PDE state constraints.*

The evolution of fast subsystem is explicitly incorporated into the PDE state constraints equation. Thus the control move at time k, under the MPC law of this scenario, is calculated by solving the following constrained minimization problem:

$$\begin{aligned} \min_{\tilde{u}^N} \quad & \sum_{j=0}^{N-1} \Big[z_s(k+j|k)^T \tilde{C}_{ds}^T Q \tilde{C}_{ds} z_s(k+j|k) + \tilde{u}(k+j|k)^T R \tilde{u}(k+j|k) \Big] \\ & + z_s(k+N|k)^T \bar{Q} z_s(k+N|k) \end{aligned} \tag{3.53}$$

$$\begin{aligned} s.t. \quad & z_s(k+j+1|k) = \tilde{A}_{ds} z_s(k+j|k) + \tilde{B}_{ds} \tilde{u}(k+j|k) \\ & z_f(k+j+1|k) = \tilde{A}_{df} z_f(k+j|k) + \tilde{B}_{df} \tilde{u}(k+j|k) \\ & \tilde{u}^{min} \leq \tilde{u}(k+j|k) \leq \tilde{u}^{max} \\ & \mathcal{X}^{min} \leq \int_0^1 r_s(z) x_{2s}(z, k+j|k) dz + \int_0^1 r_f(z) x_{2f}(z, k+j|k) dz \leq \mathcal{X}^{max} \end{aligned} \tag{3.54}$$

where $\tilde{u}^N$ is a vector of future control inputs calculated on the finite horizon N, Q is a positive semidefinite matrix, R is a positive definite matrix, and $\bar{Q}$ is the terminal state penalty matrix which is the solution of the following discrete Lyapunov equation.

$$\bar{Q} = \tilde{C}_{ds}^T Q \tilde{C}_{ds} + \tilde{A}_{ds}^T \bar{Q} \tilde{A}_{ds}$$

$r_s(z)$ represents the state constraint distribution function associated with slow dynamics, and $r_f(z)$ represents the state constraint distribution function associated with fast dynamics. The above constrained optimization problem can be considered as a quadratic programming problem. Given any initial condition $z_s(0)$, the above formulation gives a feasible solution.

Remark 6 *In the predictive control law of Eqs.* (3.53)-(3.54)*, the optimization objective function is concerned with only slow dynamics; however, the PDE state constraints equation includes two contributions. One involves slow dynamics and the other complementary contribution involves fast dynamics. In this way, the fast dynamics are incorporated into the PDE state constraints, which play a part in the construction of a optimal control move. It is necessary to emphasize that although the fast dynamics are involved in the PDEs state constraints equation, they are not involved in the objective function, which keeps relatively low computational complexity.*

Next, the receding horizon regulator of Eq. (3.52) is applied on the discrete-time dynamic model of Eq. (3.51) which is developed with Cayley transform. In this case, the predictive control law is given as the following constrained minimization problem:

$$\begin{aligned}\min_{\tilde{u}^N} \quad & \sum_{j=0}^{N-1} \Big[(\bar{C}_{ds} z_s(k+j-1|k) + \bar{D}_{ds}\tilde{u}(k+j|k))^T Q (\bar{C}_{ds} z_s(k+j-1|k) + \bar{D}_{ds}\tilde{u}(k+j|k)) \\ & + \tilde{u}(k+j|k)^T R \tilde{u}(k+j|k) \Big] + z_s(k+N|k)^T \bar{Q} z_s(k+N|k) \end{aligned} \tag{3.55}$$

$$\begin{aligned} s.t. \quad & z_s(k+j|k) = \bar{A}_{ds} z_s(k+j-1|k) + \bar{B}_{ds}\tilde{u}(k+j|k) \\ & z_f(k+j|k) = \bar{A}_{df} z_f(k+j-1|k) + \bar{B}_{df}\tilde{u}(k+j|k) \\ & \tilde{u}^{min} \le \tilde{u}(k+j|k) \le \tilde{u}^{max} \\ & \mathcal{X}^{min} \le \int_0^1 r_s(z) x_{2s}(z, k+j|k) dz + \int_0^1 r_f(z) x_{2f}(z, k+j|k) dz \le \mathcal{X}^{max} \end{aligned} \tag{3.56}$$

where $\tilde{u}^N$, N, Q, R, $\bar{Q}$, $r_s(z)$, and $r_f(z)$ represent the same meanings as in Eqs. (3.53)-(3.54). The constrained optimization problem in Eq. (3.55) can be considered as a quadratic programming for $\tilde{u}^N$. Due to the different type of discretization, the above quadratic programming is given by

$$\min_{\tilde{u}^N} \quad (\tilde{u}^N)^T \bar{H} \tilde{u}^N + 2(\tilde{u}^N)^T \bar{G} z_s(k-1) \tag{3.57}$$

where the matrices $\bar{H}$ and $\bar{G}$ are derived in the same way as in Chapter 2.

The state constraints distribution functions in the above PDE state constraints equations will be replaced later by particular matrices derived from the discrete-time system representations in the ensuing simulation section. Both the predictive control laws of Eqs. (3.53)-(3.54) and Eqs. (3.55)-(3.56) can satisfy the input constraints and also guarantee the PDE state constraints. The main difference between these two predictive control laws is that they are founded on different type of discrete-time system representation. The discrete-time model derived by Tustin's approximation and Cayley transformation introduces the input injection directly into the state constraints equation together with the matrix $\bar{D}_d$.

3.7 Simulation studies

In this section, the performance of the two MPC formulations developed in the previous section is demonstrated and compared through computer simulations. The PDE system given by Eqs. (3.4)-(3.6) is considered in this section, with $\alpha = 0.05$ and $c = 1$. The operating steady state $w_s(z)$ with these values of parameters is verified to be a stable one. The state constraints distribution function, $r(z)$, is given as $r(z) = \delta(z - z_c)$ for $z \in [0, 1]$ and $z_c = 1$, which means that state constraints are only to be actualized at the single point of the right boundary. The control objective is to enforce the first order derivative of control move, i.e., $\dot{u}(t)$, and the PDE state at the desired point subject to the input and

state constraints. For this simulation, the first 15 pairs of modes are used to approximate the plant model and the first 5 pairs of modes are considered to be the dominant ones, which means that the slow subsystem contains only the first 5 pairs of modes and the fast subsystem includes 10 pairs of modes. By incorporating the control variable $u(t)$ into the slow subsystem, one can obtain the following two discrete-time model representations that describe the transient evolution of the system dynamics:

$$\begin{cases} z_s(k+1) = \tilde{A}_{ds} z_s(k) + \tilde{B}_{ds} \tilde{u}(k) \\ z_f(k+1) = \tilde{A}_{df} z_f(k) + \tilde{B}_{df} \tilde{u}(k) \\ y(k) = \tilde{C}_{ds} z_s(k) + \tilde{C}_{df} z_f(k) \end{cases} \tag{3.58}$$

and

$$\begin{cases} z_s(k) = \bar{A}_{ds} z_s(k-1) + \bar{B}_{ds} \tilde{u}(k) \\ z_f(k) = \bar{A}_{df} z_f(k-1) + \bar{B}_{df} \tilde{u}(k) \\ y(k) = \bar{C}_{ds} z_s(k-1) + \bar{C}_{df} z_f(k-1) + \bar{D}_d \tilde{u}(k) \end{cases} \tag{3.59}$$

where,

$$z_s(k) = \begin{bmatrix} u(k) & a_{11}(k) & a_{21}(k) & \cdots & a_{15}(k) & a_{25}(k) \end{bmatrix}^T$$

$$z_f(k) = \begin{bmatrix} a_{16}(k) & a_{26}(k) & \cdots & a_{115}(k) & a_{215}(k) \end{bmatrix}^T$$

Remark 7 *Although the operating steady state $w_s(z)$ is stable, the whole system is unstable because the input variable u is incorporated into the state vectors. In other words, the eigenvalues of the discrete-time system represented in Eq.* (3.58) *and Eq.* (3.59) *consist of the eigenvalue of PDE system which are stable and the unstable eigenvalue* 1 *which is associated with the input variable $u(k)$.*

Next we implement the two MPC formulations proposed in the previous section on the discrete-trim system represented in Eq. (3.58). The initial condition of the PDE state is

considered to be

$$w_0(z) = \begin{cases} 0.4z & 0 \le z \le 0.5 \\ -0.4z + 0.4 & 0.5 < z \le 1 \end{cases}$$

The slow subsystem is utilized in the objective function and the fast subsystem is only involved in the state constraints equation. For this case, the predictive control law is given by Eqs. (3.60)-(3.61):

$$\begin{aligned} \min_{\tilde{u}^N} \quad & \sum_{j=0}^{N-1} \Big[z_s(k+j|k)^T \tilde{C}_{ds}^T Q \tilde{C}_{ds} z_s(k+j|k) + \tilde{u}(k+j|k)^T R \tilde{u}(k+j|k) \Big] \\ & + z_s(k+N|k)^T \bar{Q} z_s(k+N|k) \end{aligned} \tag{3.60}$$

$$\begin{aligned} s.t. \quad & z_s(k+j+1|k) = \tilde{A}_{ds} z_s(k+j|k) + \tilde{B}_{ds} \tilde{u}(k+j|k) \\ & z_f(k+j+1|k) = \tilde{A}_{df} z_f(k+j|k) + \tilde{B}_{df} \tilde{u}(k+j|k) \\ & \tilde{u}^{min} \le \tilde{u}(k+j|k) \le \tilde{u}^{max} \\ & \mathcal{X}^{min} \le \tilde{C}_{ds} z_s(k) + \tilde{C}_{df} z_f(k) \le \mathcal{X}^{max} \end{aligned} \tag{3.61}$$

where $Q = I$, $R = 0.01I$, and the regulator horizon $N = 50$. The input and state constraints are given as $\tilde{u}^{min} = -65$, $\tilde{u}^{max} = 95$, $\mathcal{X}^{min} = -0.60$, $\mathcal{X}^{max} = 0.4$. In all ensuing simulation studies, we keep the initial condition, input and state constraints identical with the above values. The resulting constrained optimization problem becomes a quadratic program which can be solved through the MATLAB subroutine QuadProg. Then we inject the optimal control move into the continuous full-state plant model. Fig. 3.1 shows the resulting profiles of the closed-loop system under the performance of the MPC law constructed on the basis of standard discrete-time system, which is given by Eqs. (3.60)-(3.61). The evolution of PDE state $w(z,t)$ (vertical displacement of beam) is shown in Fig. 3.1(a) and Fig. 3.1(b). The evolution of $\partial w(z,t)/\partial t$ (vertical velocity of beam) is shown in Fig. 3.2(a). The constrained

output $\partial w(1,t)/\partial z$ is shown in Fig. 3.2(b). The constrained control variable $\tilde{u}$, which is the first-order derivative of input u, is shown in Fig. 3.3(a). The input u is shown in Fig. 3.3(b).

Next we implement the second predictive control law proposed in the previous section on the discrete-time system represented in Eq. (3.59). The MPC formulation with the open-loop quadratic cost function and input and state constraints equations are given as the following:

$$\begin{aligned}\min_{\tilde{u}^N} \quad & \sum_{j=0}^{N-1} \Big[(\bar{C}_{ds}\bar{z}_s(k+j-1|k) + \bar{D}_{ds}\tilde{u}(k+j|k))^T Q(\bar{C}_{ds}\bar{z}_s(k+j-1|k) + \bar{D}_{ds}\tilde{u}(k+j|k)) \\ & + \tilde{u}(k+j|k)^T R\tilde{u}(k+j|k) \Big] + \bar{z}_s(k+N|k)^T \bar{Q}_2 \bar{z}_s(k+N|k) \end{aligned} \tag{3.62}$$

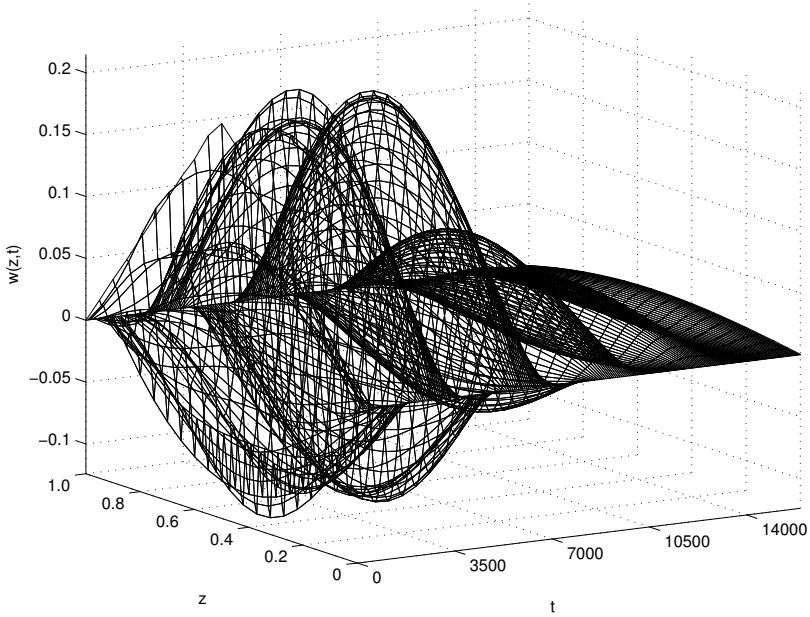
$$\begin{aligned} s.t. \quad & \bar{z}_s(k+j|k) = \bar{A}_{ds}\bar{z}_s(k+j-1|k) + \bar{B}_{ds}\tilde{u}(k+j|k) \\ & \bar{z}_f(k+j|k) = \bar{A}_{df}\bar{z}_f(k+j-1|k) + \bar{B}_{df}\tilde{u}(k+j|k) \\ & \tilde{u}^{min} \le \tilde{u}(k+j|k) \le \tilde{u}^{max} \\ & \mathcal{X}^{min} \le \bar{C}_{ds}\bar{z}_s(k-1) + \bar{C}_{df}\bar{z}_f(k-1) + \bar{D}_d\tilde{u}(k) \le \mathcal{X}^{max} \end{aligned} \tag{3.63}$$

The values of Q, R, initial condition, input and state constraints are kept identical with those in Eq. (3.60)-(3.61). But the values of horizon length N is changed to 80 in order to make sure there exist feasible solutions and active constraints. Fig. 3.4 shows the resulting profiles of the closed-loop system under the performance of the MPC law constructed on the basis of Cayley discrete-time system, which is given by Eqs. (3.62)-(3.63). Fig. 3.4(a) and Fig. 3.4(b) show the evolution of PDE state $w(z,t)$ (vertical displacement of beam). The evolution of $\partial w(z,t)/\partial t$ (vertical velocity of beam) is shown in Fig. 3.5(a). The constrained output $\partial w(1,t)/\partial z$ in Fig. 3.5(b) shows that the state constraints are satisfied for all time. The constrained control variable $\tilde{u}$, which is the first-order derivative of input u, is shown in Fig. 3.6(a). The input u is shown in Fig. 3.6(b).

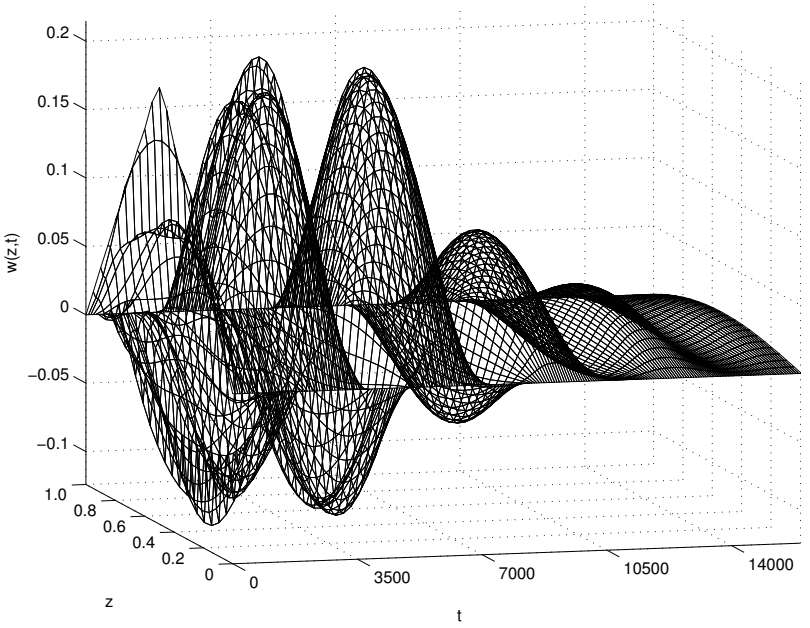
As demonstrated in Fig. 3.1 and Fig. 3.4, the input and state constraints are guaranteed by accounting for the fast dynamics in the state constraints equation for the predictive control laws founded on both standard and Cayley discrete-time system. But the MPC law based on Cayley discrete-time model needs a larger horizon to get feasible solutions.

3.8 Conclusions

In this chapter, model predictive control algorithms are developed for the Euler-Bernoulli beam system with consideration of input and state constraints. The Euler-Bernoulli beam model which describes the vertical motion of a thin horizontal beam with small displacements from rest is considered , and the eigenvalue problem of the beam system is solved. Since the Euler-Bernoulli equation is neither a hyperbolic PDE nor a parabolic PDE, a particular projection method and modal decomposition technique are adopted to obtain a finite-dimensional system that captures the dominant dynamics of the PDE system, which is used as the basis for the low-order controller design subsequently. Two different discretization methods, standard and Tustin's discretization, are used to derive the discrete-time modal representations. The transfer function of the PDE system is also derived in order to obtain the Tustin's discrete-time model representation with Cayley transform. Two model predictive control laws, in which the slow dynamics are involved in the performance objective function and the fast dynamics are only accounted for in the state constraints equation, are developed on the basis of two different discrete-time system representations. Finally, performance of the proposed MPC formulations are demonstrated and compared, via simulation studies, to achieve the control objectives.

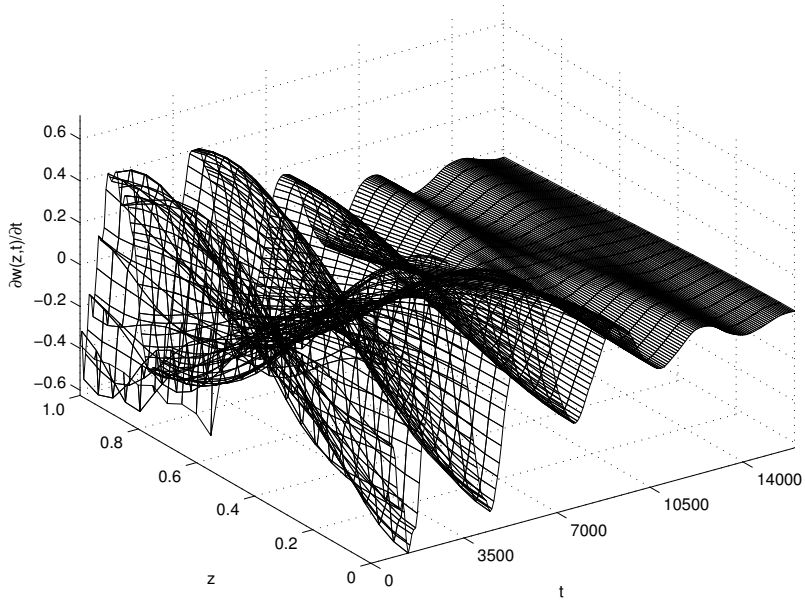


(a) Profile of displacement $w(z,t)$ (a).

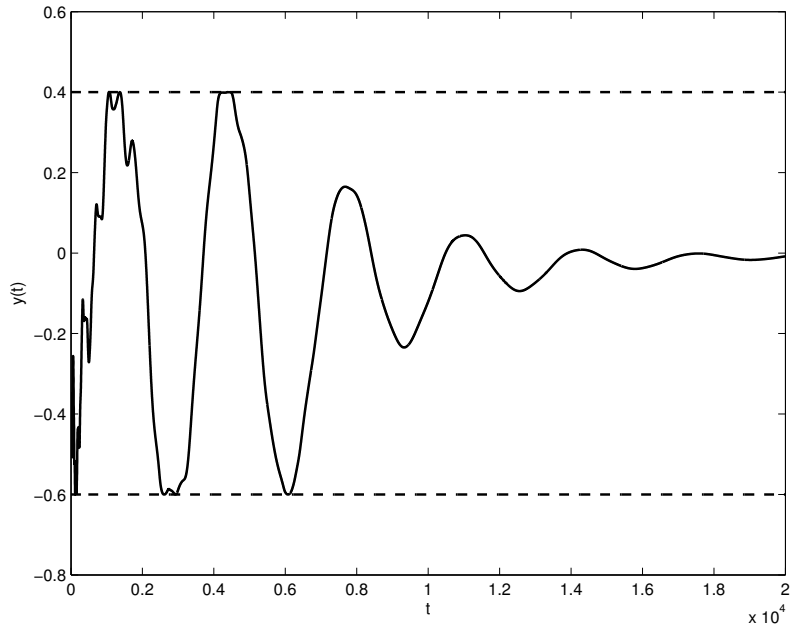


(b) Profile of displacement $w(z,t)$ (b).

Figure 3.1: Profiles of the displacement under the MPC law of Eqs. (3.60) and (3.61) constructed on standard discrete-time system with input and state constraints accounting for full modes.

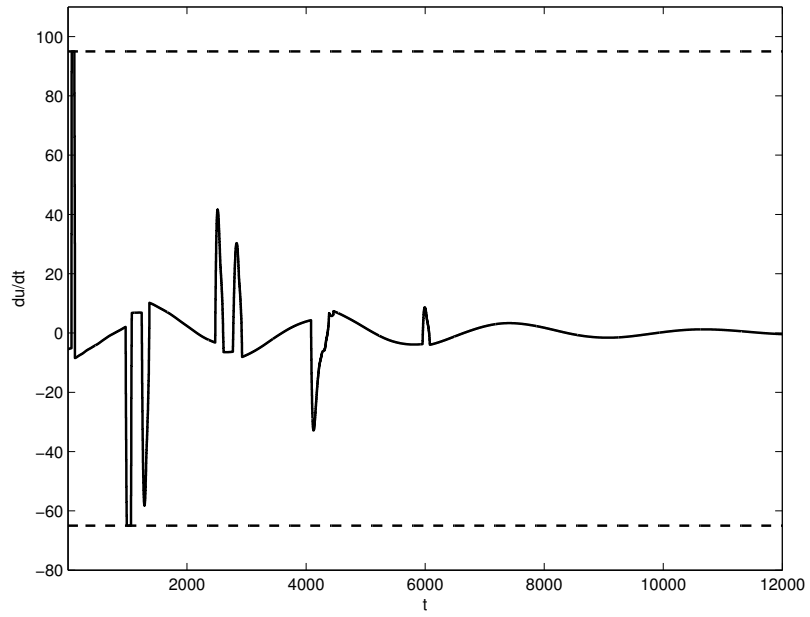


(a) Profile of velocity $\partial w(z,t)/\partial t$.

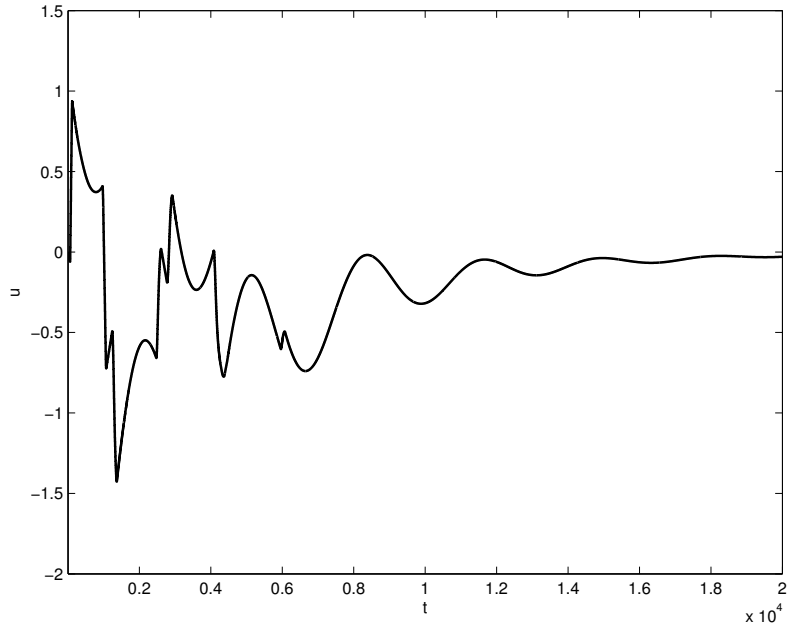


(b) Profile of constrained state at the boundary $\partial w(1,t)/\partial t$ (solid line); state constraints (dashed line).

Figure 3.2: Profiles of the velocity under the MPC law of Eqs. (3.60) and (3.61) constructed on standard discrete-time system with input and state constraints accounting for full modes.

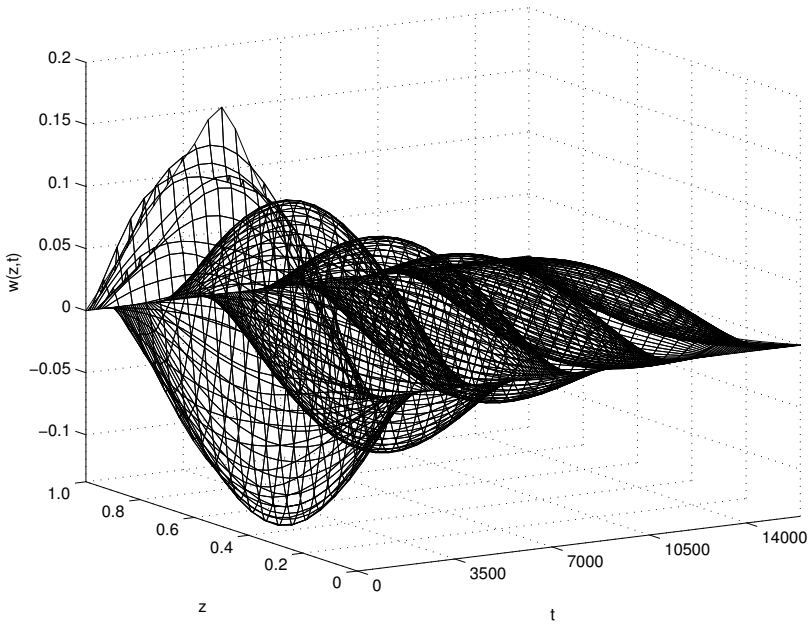


(a) Profile of constrained first-order derivative of input $\tilde{u}$ (solid line); constraints of $\tilde{u}$ (dashed line).

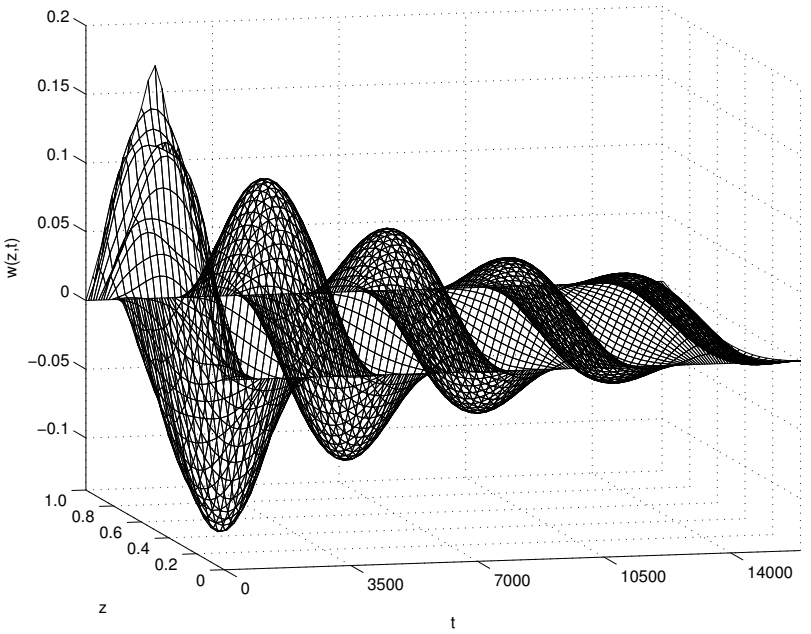


(b) Profile of boundary input u.

Figure 3.3: Profiles of the control variable computed by the MPC law of Eqs. (3.60) and (3.61) constructed on standard discrete-time system with input and state constraints accounting for full modes.

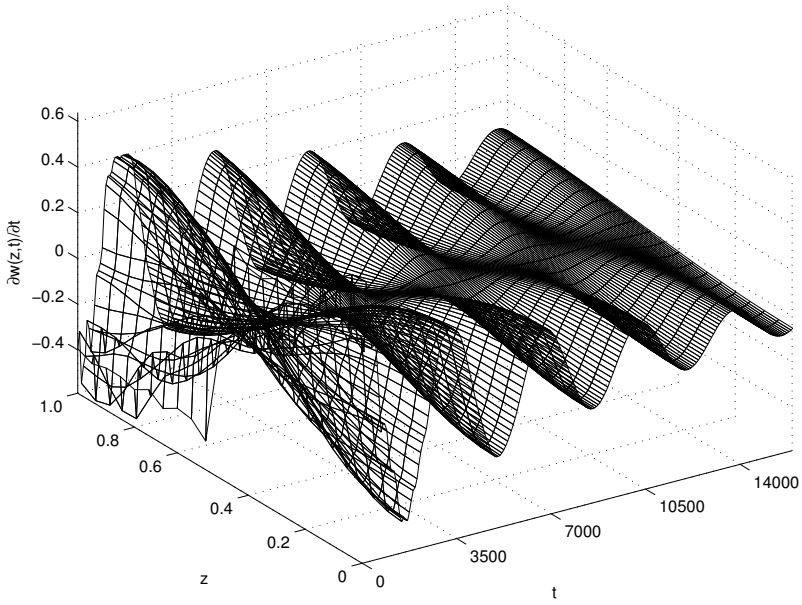


(a) Profile of displacement $w(z,t)$ (a).

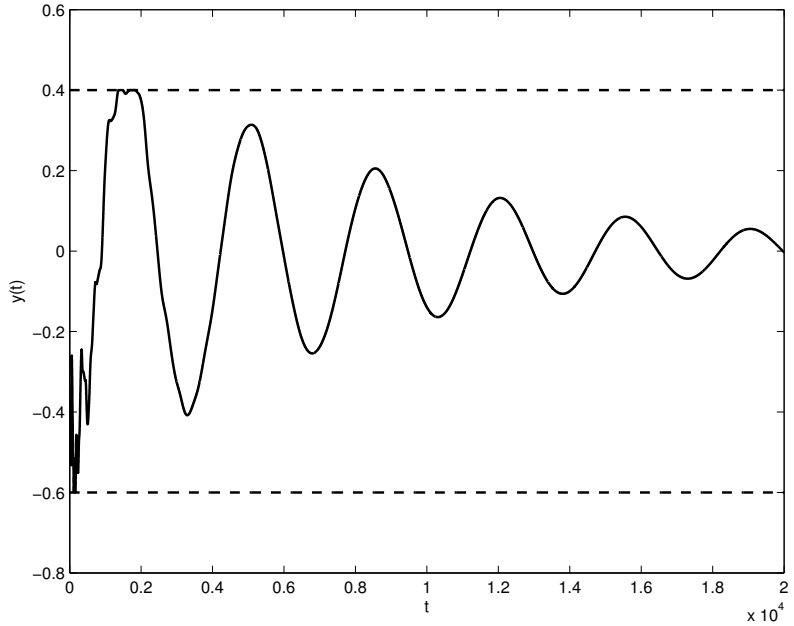


(b) Profile of displacement $w(z,t)$ (b).

Figure 3.4: Profiles of the displacement under the MPC law of Eqs. (3.62) and (3.63) constructed on Cayley discrete-time system with input and state constraints accounting for full modes.

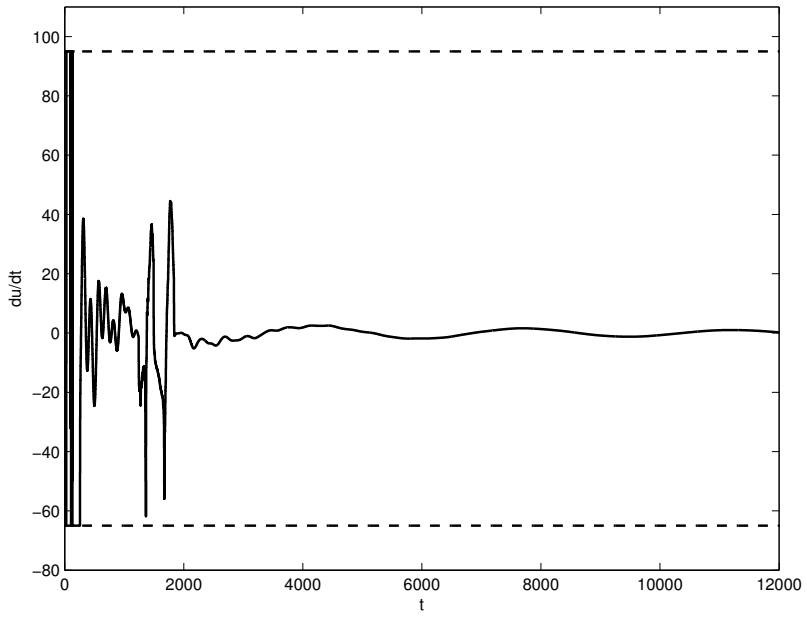


(a) Profile of velocity $\partial w(z,t)/\partial t$.

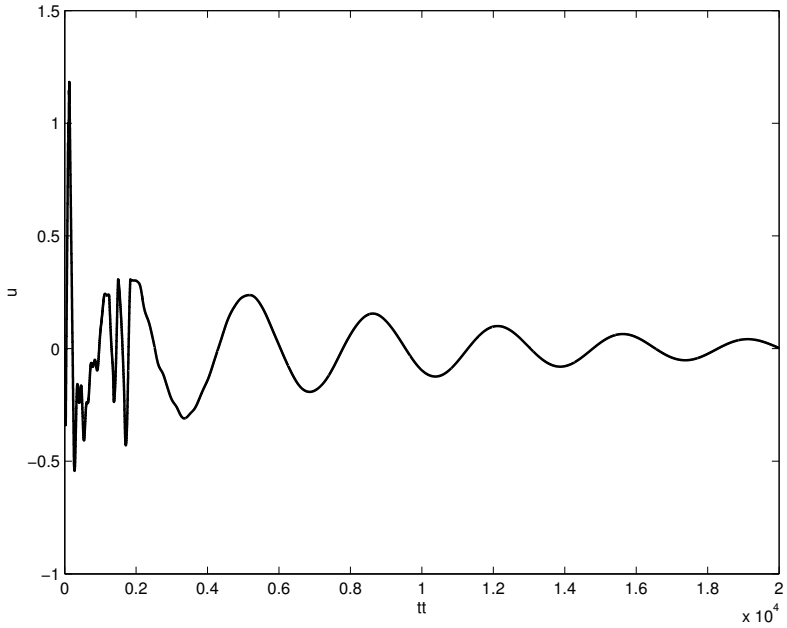


(b) Profile of constrained state at the boundary $\partial w(1,t)/\partial t$ (solid line); state constraints (dashed line).

Figure 3.5: Profiles of the velocity under the MPC law of Eqs. (3.62) and (3.63) constructed on Cayley discrete-time system with input and state constraints accounting for full modes.



(a) Profile of constrained first-order derivative of input $\tilde{u}$ (solid line); constraints of $\tilde{u}$ (dashed line).



(b) Profile of boundary input u.

Figure 3.6: Profiles of the control variable computed by the MPC law of Eqs. (3.62) and (3.63) constructed on Cayley discrete-time system with input and state constraints accounting for full modes.

Chapter 4

Explicit/Multi-Parametric Model Predictive Control of Dissipative Distributed Parameter Systems

4.1 Introduction

This chapter focuses on the development of an explicit/multi-parametric model predictive control algorithm to stabilize the discrete infinite-dimensional system arising from the discrete state space modeling of the certain class of dissipative distributed parameter systems, specifically, the parabolic partial differential equation (PDE) system and the flexible Euler-Bernoulli beam system. In particular, the class of dissipative distributed parameter system yields discrete modal representation which captures the dominant dynamics of the PDE system through Galerkin's method and modal decomposition techniques. The proposed explicit/multi-parametric model predictive control algorithm is constructed in a way that the objective function is concerned with only the low-order modes, while the state constraints involve both the low-order and higher-order modes. The explicit model predictive control problem is solved off-line by dynamic programming and multi-parametric quadratic programming techniques [42], and the solution is expressed as a piecewise affine function with its corresponding critical regions [26].

4.2 System description

In this work, we focus on the following time-invariant, infinite-dimensional system $\Sigma(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ with input u and output y:

$$\dot{x}(t) = \mathcal{A}x(t) + \mathcal{B}u(t), \quad x(0) = x_0$$
$$y(t) = \mathcal{C}x(t) + \mathcal{D}u(t) \tag{4.1}$$

subject to the following input and state constraints:

$$\mathcal{U}^{min} \leq u \leq \mathcal{U}^{max} \tag{4.2}$$

$$\mathcal{X}^{min} \leq (r_k, x) \leq \mathcal{X}^{max} \tag{4.3}$$

where $\mathcal{A} \in \mathcal{L}(X)$ is a generator of a strongly continuous semigroup on a separable complex Hilbert state space X, $\mathcal{B} \in \mathcal{L}(U, X)$ is a bounded linear operator from a Hilbert input space U to X, $\mathcal{C} \in \mathcal{L}(X, Y)$ is a bounded linear operator from X to a Hilbert output space Y, and $\mathcal{D} \in \mathcal{L}(U, Y)$ is a bounded operator from U to Y. $\mathcal{U}^{max}$ and $\mathcal{U}^{min}$ are real numbers denoting the upper and lower constraints of the input, respectively; $\mathcal{X}^{max}$ and $\mathcal{X}^{min}$ are real numbers representing the upper and lower constraints of the state, respectively. The square integrable function $r_k \in L_2[0, 1]$ is the k-th state constraint distribution function representing how the state constraints are enforced within the spatial interval [0,1]. The notation (r_k, x) represents the inner product of r_k and x. The inner product and norm of w_1, $w_2 \in L_2[0, 1]$ are given as:

$$(w_1,\, w_2) = \int_0^1 w_1(z)w_2(z)dz, \quad \|w_1\|_2 = (w_1,\, w_1)^{\frac{1}{2}} \tag{4.4}$$

Since the discrete-time controller is simpler for implementation and is more acceptable for practitioners, we transform the continuous-time infinite-dimensional system of Eq. (4.1) into

the following discrete-time infinite-dimensional $\Sigma_d(\mathcal{A}_d, \mathcal{B}_d, \mathcal{C}_d, \mathcal{D}_d)$:

$$x(k+1) = \mathcal{A}_d x(k) + \mathcal{B}_d u(k)$$
$$y(k) = \mathcal{C}_d x(k) + \mathcal{D}_d u(k) \tag{4.5}$$

where $\mathcal{A}_d \in \mathcal{L}(X)$, $\mathcal{B}_d \in \mathcal{L}(U, X)$, $\mathcal{C}_d \in \mathcal{L}(X, Y)$, and $\mathcal{D}_d \in \mathcal{L}(U, Y)$ with Hilbert spaces U and Y. The relationship between the $\Sigma(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ and $\Sigma_d(\mathcal{A}_d, \mathcal{B}_d, \mathcal{C}_d, \mathcal{D}_d)$ is given by:

$$\begin{bmatrix} \mathcal{A}_d & \mathcal{B}_d \\ \mathcal{C}_d & \mathcal{D}_d \end{bmatrix} = \begin{bmatrix} (\delta I + \mathcal{A})(\delta I - \mathcal{A})^{-1} & \sqrt{2\delta}(\delta I - \mathcal{A})^{-1}\mathcal{B} \\ \sqrt{2\delta}\mathcal{C}(\delta I - \mathcal{A})^{-1} & \mathcal{G}(\delta) \end{bmatrix} \tag{4.6}$$

where $\delta = 2/\Delta t$ and $\Delta t > 0$ denotes the discretization time interval. $\mathcal{G}(\delta)$ is defined as the transfer function of the PDE system evaluated at δ.

In order to decompose the system of Eq. (4.5) into a slow subsystem including all the unstable modes and a complement fast subsystem including only stable modes, we define two orthogonal projection operators, $\mathcal{P}_s$ and $\mathcal{P}_f$ such that $x_s(k) = \mathcal{P}_s x(k)$ and $x_f(k) = \mathcal{P}_f x(k)$, then the state $x(k)$ can be decomposed as

$$x(k) = x_s(k) + x_f(k) \tag{4.7}$$

Applying the separation operators $\mathcal{P}_s$ and $\mathcal{P}_f$ to the system $\Sigma_d(\mathcal{A}_d, \mathcal{B}_d, \mathcal{C}_d, \mathcal{D}_d)$, the above system can be written in the following decomposed form:

$$x_s(k+1) = \mathcal{A}_{ds} x_s(k) + \mathcal{B}_{ds} u(k)$$
$$x_f(k+1) = \mathcal{A}_{df} x_f(k) + \mathcal{B}_{df} u(k)$$
$$y(k) = \mathcal{C}_{ds} x_s(k) + \mathcal{C}_{df} x_f(k) + \mathcal{D}_d u(k) \tag{4.8}$$

Remark 8 *The separation can always be conducted when the slow modes and fast modes are not coupled, i.e., $\mathcal{A}_d$ is diagonal. Furthermore, $\mathcal{A}_d$ is diagonalizable if the operator $\mathcal{A}_d$ has a point spectrum, i.e., $\sigma(\mathcal{A}_d) = \{\lambda_{d1}, \lambda_{d2}, \cdots\}$.*

We assume the solution to $x(k)$ can be expressed by:

$$x(k) = \sum_{n=1}^{\infty} a_n(k)\phi_n(z) \tag{4.9}$$

where $\phi_n(z)$ is a smooth and orthogonal function on the spatial domain $[0, 1]$. Projecting the system of Eq. (4.8) on the $\phi_n(z)$ domain, we can obtain the following model representation for subsequent controller design:

$$\begin{aligned} a_s(k+1) &= A_{ds}a_s(k) + B_{ds}u(k) \\ a_f(k+1) &= A_{df}a_f(k) + B_{df}u(k) \\ x(k) &= x_s(k) + x_f(x) = C_{ds}a_s(k) + C_{df}a_f(k) \end{aligned} \tag{4.10}$$

4.3 Explicit/multi-parametric model predictive control

Despite significant efforts on the development of explicit/multi-parametric model predictive control for a discrete-time lumped parameter system (see, [26], [29], [31]), few results are available on the explicit model predictive control for the distributed parameter system. In this work, we will explore the synthesis of explicit MPC for the class of dissipative distributed parameter systems.

4.3.1 Preliminaries of multi-parametric programming

For the discrete-time system:

$$x(k+1) = Ax(k) + Bu(k)$$

The explicit/multi-parametric MPC problem is defined as the following multi-stage optimization problem:

$$V_N(x) = \min_{u^N} J(u^N, x)$$

$$
\begin{aligned}
&= \min_{u^N} \sum_{k=0}^{N-1} \left[x(k)^T Q x(k) + u(k)^T R u(k)\right] + x(N)^T \bar{Q} x(N) \\
s.t. \quad & x(k+1) = Ax(k) + Bu(k) \\
& \mathcal{U}^{min} \leq u(k) \leq \mathcal{U}^{max} \\
& \mathcal{X}^{min} \leq x(k) \leq \mathcal{X}^{max}
\end{aligned} \tag{4.11}
$$

where vector u^N contains the sequence of control inputs $\{u(0), u(1), \cdots, u(N-1)\}$; Q and $\bar{Q}$ are positive semidefinite matrices and R is a positive definite matrix of appropriate dimensions; N is the predictive horizon. The term $x(N)^T \bar{Q} x(N)$ is the quadratic terminal cost where $\bar{Q}$ is determined from the solution of the corresponding Lyapunov equation. The above multi-stage optimization problem can be decomposed into the following stage-wise optimization problems [31]:

$$
\begin{aligned}
V_k &= \min_{u(k)} J_k\big(u(k), x(k)\big) \\
&= \min_{u(k)} \sum_{i=k}^{N-1} \left[x(i)^T Q x(i) + u(i)^T R u(i)\right] + x(N)^T \bar{Q} x(N) \\
s.t. \quad & x(i+1) = Ax(i) + Bu(i) \\
& \mathcal{U}^{min} \leq u(i) \leq \mathcal{U}^{max} \\
& \mathcal{X}^{min} \leq x(i) \leq \mathcal{X}^{max}
\end{aligned} \tag{4.12}
$$

for all $k = N-1, N-2, \cdots, 0$. The problem of Eq. (4.12) is equivalent to the problem of Eq. (4.11). In problem of Eq. (4.11), the sequence of current and future control variables is considered in the optimization, however in the problem of Eq. (4.12) only the current control variable is considered in the optimization as we assume the future control variables have been obtained in the previous stages.

With the principle of dynamic programming, the problem of Eq. (4.12) at each stage can be solved as a multi-parametric programming problem by considering the current control

variable $u(k)$ as the optimization variable, and considering the current state variable and future control variables $\theta_k = \begin{bmatrix} x(k) & u(k+1) & \cdots & u(N-1) \end{bmatrix}^T$ as the vector of parameters. Substitution of the state predictions in the quadratic objective function and inequalities of Eq. (4.12) results in the following multi-parametric quadratic programming for $u(k)$ [43]:

$$\begin{aligned} V_k = \min_{u(k)} [\frac{1}{2} u(k)^T H u(k) + \theta_k^T F u(k)] + \frac{1}{2} \theta_k^T Y \theta_k \\ s.t. \quad Gu(k) \leq W + E\theta_k \end{aligned} \tag{4.13}$$

where, H, F, Y, G, W, E are obtained from Q, R, $\bar{Q}$ by straightforward algebraic manipulation. Since the optimization is only on the current control variable $u(k)$, the term $\frac{1}{2}\theta_k^T Y \theta_k$ in Eq. (4.13) is usually removed. Our goals are to solve the multi-parametric programming problem in Eq. (4.13) off-line, and to provide the optimal control move $u(k)$ as an explicit function of the parameters vector θ_k.

We consider a set of active constraints $\tilde{G}$, $\tilde{W}$, $\tilde{S}$, and assume that the rows of $\tilde{G}$ are linearly independent. Through solving the first-order Karush-Kuhn-Tucker (KKT) conditions for the multi-parametric programming problem, the critical regions of the optimal control variable are given by [26]:

$$\begin{cases} \big(GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{S} - S\big)\theta_k \leq W - GH^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{W} \\ (\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{S}\theta_k \leq -(\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{W} \end{cases} \tag{4.14}$$

and over the above critical region, the solution to the multi-parametric quadratic programming of Eq. (4.13) at stage k is given by the following affine function of the parameters vector θ_k:

$$\begin{aligned} u(k) = &\big(H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{S} - H^{-1}F^T\big)\theta_k \\ &+ H^{-1}\tilde{G}^T(\tilde{G}H^{-1}\tilde{G}^T)^{-1}\tilde{W} \end{aligned} \tag{4.15}$$

For the sake of simplicity, the solution in Eqs. (4.14)-(4.15) is written as the following explicit

linear piecewise affine expression of the parameters vector θ_k [28, 29]:

$$u(k) = f_k(\theta_k) = \begin{cases} K_k^1\theta_k + L_k^1 & \text{if } A_k^1\theta_k \leq B_k^1 \\ \vdots & \vdots \\ K_k^{s_k}\theta_k + L_k^{s_k} & \text{if } A_k^{s_k}\theta_k \leq B_k^{s_k} \end{cases} \tag{4.16}$$

where the supperscript s_k denotes the number of critical regions; K^i, L^i, A^i_{ineq}, and B^i_{ineq} are matrices and vectors of appropriate dimensions, $i = 1, 2, \cdots, s_k$.

At stage k, the expression in Eq. (4.16) gives the control variable $u(k)$ as an explicit function of $x(k)$, $u(k+1)$, $u(k+2)$, $\cdots$, $u(N-1)$. Furthermore, since the future control variables $u(k+1)$, $u(k+2)$, $\cdots$, $u(N-1)$ are obtained from the previous stages, they can be eliminated from Eq. (4.16) such that $u(k)$ can be derived as an explicit piecewise affine expression of the current state $x(k)$, i.e., $u(k) = \mu_k\big(x(k)\big)$.

The procedure to solve the explicit/multi-parametric MPC problem in Eq. (4.12) has the characteristic of multi-stage optimization and is based on the solution to the multi-parametric quadratic programming problem in Eq. (4.13).

At each stage $k = N-1, N-2, \cdots, 0$, the solution of the multi-parametric quadratic programming problem in Eq. (4.13) is expressed as $u(k) = \mu_k\big(x(k)\big)$. Therefore, after the final stage $k = 0$, the sequence of control variables U is computed as a sequence of explicit piecewise affine functions $U = \{\mu_0\big(x(0)\big), \cdots, \mu_{N-2}\big(x(N-2)\big), \mu_{N-1}\big(x(N-1)\big)\}$. Furthermore, the first element of the resulting control sequence will be utilized as the optimal control law and implemented to the system. Thus, an explicit/multi-parametric MPC controller is obtained as the following:

$$u^* = U(0) = \mu_0\big(x(0)\big) \tag{4.17}$$

4.3.2 Explicit/multi-parametric MPC on dissipative distributed parameter systems

Now, we apply the explicit/multi-parametric MPC control law on the discrete-time infinite-dimensional system arising from the modeling of a dissipative distributed-parameter system. In order to guarantee both the manipulated input constraints and the PDE state constraints, the explicit controller is constructed as the explicit solution to the multi-stage optimization problem where the object function is concerned with only slow dynamics while the PDE state constraint equation is concerned with both slow dynamics and fast dynamics [38].

Remark 9 *There are two main methods to involve the fast dynamics in the PDE state constraints. One method is to incorporate the fast modes explicitly and update the evolution of fast dynamics at each step over the whole horizon. The other method is to use the infinity norm of the fast dynamics in the PDE state constraints equation instead of using the fast dynamics directly. However, if the fast dynamics are explicitly incorporated in the PDE state constraints, the solution to the multi-parametric quadratic programming will be an explicit function of both slow states and fast states. As the number of fast states is relatively large, the computation effort will be very large if the fast dynamics are explicitly incorporated and updated at each stage. Furthermore, the resulting control move will be represented by a relatively large number of parameters (including slow states and fast states), which will lead to a very complex set of critical regions. Owing to the reasons above, it is not an ideal option to incorporate the fast dynamics explicitly although it has good performance in the case of the regular MPC.*

To avoid the problem mentioned in Remark 9, we use the infinity norm of fast modes to express the explicit fast states evolution in the PDE state constraints equation. The fast

states evolution satisfies the following expression:

$$a_f(i) = A_{df}^{i} a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} u(i) \tag{4.18}$$

which implies:

$$\begin{aligned} x_f(i) &= C_{df} a_f(i) \\ &= C_{df}\Big(A_{df}^{i} a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} u(i) \Big) \end{aligned} \tag{4.19}$$

Since,

$$\mathcal{U}^{min} \leq u \leq \mathcal{U}^{max}$$

we can obtain

$$C_{df}\Big(A_{df}^{i} a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} \mathcal{U}^{min} \Big) \leq x_f(i) \tag{4.20}$$

and

$$x_f(i) \leq C_{df}\Big(A_{df}^{i} a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} \mathcal{U}^{max} \Big) \tag{4.21}$$

Thus,

$$\|x_f(i)\|_\infty = C_{df}\Big(A_{df}^{i} a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} \sup |u(i)| \Big) \tag{4.22}$$

Therefore, satisfaction of the following inequalities:

$$\mathcal{X}^{min} \leq x_s(i) + \|x_f(i)\|_\infty \leq \mathcal{X}^{max} \tag{4.23}$$

can guarantee the PDE state constraints:

$$\mathcal{X}^{min} \leq x_s(i) + x_f(i) \leq \mathcal{X}^{max}$$

We use the inequalities of Eq. (4.23) as the PDE state constraints in stead of using the explicit fast dynamics. Then, the optimal explicit control move, under the explicit MPC law, is calculated by solving the following multi-stage optimization problem:

$$\min_{u(k)} \quad \sum_{i=k}^{N-1} \left[x_s(i)^T Q x_s(i) + u(i)^T R u(i) \right] + x_s(N)^T \bar{Q} x_s(N)$$

$$
\begin{aligned}
s.t. \quad & a_s(i+1) = A_{ds}a_s(i) + B_{ds}u(i) \\
& x_s(i) = C_{ds}a_s(i) \\
& \|x_f(i)\|_\infty = C_{df}\Big(A_{df}^i a_f(0) + \sum_{j=0}^{i-1} A_{df}^{i-1-j} B_{df} \sup|u(i)|\Big) \\
& \mathcal{U}^{min} \le u(i) \le \mathcal{U}^{max} \\
& \mathcal{X}^{min} \le x_s(i) + \|x_f(i)\|_\infty \le \mathcal{X}^{max} \\
\text{for all } & k = N-1,\, N-2,\, \cdots,\, 0
\end{aligned} \tag{4.24}
$$

where $\{u(0), \cdots, u(N-2), u(N-1)\}$ is a sequence of control inputs computed on the horizon N; $a_f(0)$ is the initial condition of $a_f(k)$; $Q \ge 0$, $R > 0$, $\bar{Q}$ is the solution to the discrete Lyapunov equation $\bar{Q} = C_{ds}^T Q C_{ds} + A_{ds}\bar{Q}A_{ds}$. The above multi-stage constrained optimization problem can be considered as a multi-stage multi-parametric quadratic programming problem. Given any appropriate initial conditions $a_s(0)$ and $a_f(0)$, a feasible solution can be computed by solving the multi-parametric quadratic programming problem using the algorithm and procedure given in Section 4.3.1.

In the explicit MPC law of Eq. (4.24), the constrained multi-parametric program is concerned with only slow dynamics; however, the PDE state constraints equation consists of two parts. One part explicitly involves the slow dynamics, and the other complimentary part is baed on the bounds of fast dynamics. In this way, the PDE state constraints account for the dynamics of fast subsystem in the explicit MPC formulation. However, the fast dynamics are not involved in the objective function although they are involved in the PDE state constraints, which keeps relatively low computation effort.

4.4 Simulation studies

In this section, the performance of the explicit/multi-parametric MPC formulation developed in the previous section is demonstrated and compared through computer simulations.

4.4.1 Simulation 1

For the first simulation, we consider the transport-reaction process represented by a linear parabolic PDE, of the form:

$$\frac{\partial x(z,t)}{\partial t} = b\frac{\partial^2 x(z,t)}{\partial z^2} + cx(z,t) - cu(t) \tag{4.25}$$

with the following boundary and initial conditions:

$$x(0,t) = 0, \quad x(1,t) = 0, \quad x(z,0) = x_0(z) \tag{4.26}$$

subject to the following input and state constraints:

$$\begin{aligned} &\mathcal{U}^{min} \leq u \leq \mathcal{U}^{max} \\ &\mathcal{X}^{min} \leq \int_0^1 r_k(z)x(z,t)dz \leq \mathcal{X}^{max} \end{aligned} \tag{4.27}$$

The eigenvalues and eigenvectors of the PDE system in Eqs. (4.25) and (4.26) are given as:

$$\begin{aligned} &\lambda_n = c - bn^2\pi^2 \\ &\phi_n(z) = \sqrt{2}sin(n\pi z) \end{aligned} \tag{4.28}$$

We consider the parameters $b = 1$ and $c = 2\pi^2$. Then, the first 3 eigenvalues of the system are $\lambda_1 = 9.870$, $\lambda_2 = -19.739$, $\lambda_3 = -69.087$. Since the first eigenvalue is in the right half plane, the steady state $x_s(z)$ with these values of parameters is verified to be an unstable one. The state constraints distribution function, $r_k(z)$, is given as $r_k(z) = \delta(z - z_{ck})$ for $k = 1, 2$, $z \in [0, 1]$, which means that state constraints are to be actualized at two points

$z = z_{c1}$ and $z = z_{c2}$ within the spatial interval $[0, 1]$. For this simulation, we choose $z_{c1} = 0.16$ and $z_{c2} = 0.86$. We transform the infinite-dimensional PDE system of Eq. (4.25) into a discrete-time equivalent system. Then, the first 15 modes of the discrete-time system are used to approximate the plant model and the first 3 modes are considered to be the dominant ones, which means that the slow subsystem contains only the first 3 modes and the fast subsystem includes 12 modes, i.e., the slow modes are $a_s(k) = [a_1(k)\ a_2(k)\ a_3(k)]^T$ and the fast modes are $a_f(k) = [a_4(k)\ a_5(k)\ \cdots\ a_{15}(k)]^T$.

We implement the proposed explicit MPC law of Eq. (4.24) on the discrete-time system. The control objective is to compute the control variable $u(k)$ as an explicit piecewise affine function of the eigen-coefficient $a(k)$ and then stabilize the plant (full-state) system and enforce the input and PDE state constraints through implementing the resulting explicit control input on the plant system. The initial condition of the PDE state is considered to be $x(z, 0) = \phi_1(z) = \sqrt{2}sin(\pi z)$, which implies that $a_n(0) = \big(\phi_n(z), x(z, 0)\big) = \big(\phi_n(z), \phi_1(z)\big)$. For this simulation, we choose $Q = I$, $R = 0.0001$, $N = 4$. The input and PDE state constraints are considered as $\mathcal{U}^{min} = -23$, $\mathcal{U}^{max} = 23$, $\mathcal{X}^{min} = -1.5$, $\mathcal{X}^{max} = 1.5$. The resulting explicit MPC controller consists of 19 critical regions and corresponding affine control laws. The closed-loop evolution of the PDE state $x(z, t)$ under the implementation of the explicit MPC law given by Eq. (4.24) is shown in Fig. 4.1. The constrained optimal control move computed by the explicit MPC law is given in Fig. 4.2, and the constrained PDE states $x(z = 0.16, t)$ and $x(z = 0.86, t)$ of the closed-loop system are shown in Fig. 4.3. It is clearly demonstrated in Fig. 4.1 that the PDE system is stabilized under the implementation of the explicit MPC. Fig. 4.2 and Fig. 4.3 show that both the input and the PDE state constraints are guaranteed with the explicit predictive controller. The critical regions for the explicit control law are provided in Fig. 4.4. The resulting control laws corresponding

to these critical regions in Fig. 4.4 are as following:

$$u^*(k) = \begin{cases} 46.05a_1 + 13.43a_3, & \text{yellow region} \\ -23, & \text{green region} \\ 23, & \text{red region} \end{cases} \tag{4.29}$$

4.4.2 Simulation 2

The second simulation is conducted on the flexible Euler-Bernoulli beam system which is developed in Chapter 3. The model of the flexible beam system is given by the following fourth-order PDE:

$$\frac{\partial^2 w}{\partial t^2} - 2\alpha \frac{\partial^3 w}{\partial t \partial z^2} + c\frac{\partial^4 w}{\partial z^4} = 0 \tag{4.30}$$

subject to the following boundary and initial conditions:

$$w(0,t) = 0 = w(1,t) = \frac{d^2 w(0,t)}{dz^2}$$

$$\frac{d^2 w(1,t)}{dz^2} = u(t) \tag{4.31}$$

$$w(z,0) = w_0(z) \tag{4.32}$$

where $w(z,t)$ is the vertical deflection of the beam at time t and at a distance z from one end, $0 \leq z \leq 1$; $u(t)$ is the control variable.

As in the previous chapter, we transform the non-homogeneous boundary condition into a homogeneous equivalent by introducing a new state variable $v(z,t)$, which satisfies

$$w(z,t) = v(z,t) + b(z)u(t)$$

and

$$b(z) = \frac{1}{6}(z^3 - z)$$

In order to transform the term of second-order derivative respect to time into a first-order derivative, we introduce a state vector

$$\begin{bmatrix} x_1(z,t) & x_2(z,t) \end{bmatrix}^T = \begin{bmatrix} v(z,t) & \frac{\partial w(z,t)}{\partial t} \end{bmatrix}^T$$

and a new control variable:

$$\tilde{u}(t) = \frac{du(t)}{dt}$$

Thus the system representation is transformed into the following equivalent:

$$\frac{\partial}{\partial t} \begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c\mathcal{A}_0 & 2\alpha\mathcal{A}_0^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_1(z,t) \\ x_2(z,t) \end{bmatrix} + \begin{bmatrix} -b(z) \\ 0 \end{bmatrix} \tilde{u}(t) \tag{4.33}$$

subject to

$$\begin{aligned} & x_1(0,t) = x_1(1,t) = 0 \\ & \frac{\partial^2 x_1(0,t)}{\partial z^2} = \frac{\partial^2 x_1(1,t)}{\partial z^2} = 0 \end{aligned} \tag{4.34}$$

Where $\mathcal{A}_0\phi(z) = d^4\phi(z)/dz^4$. For the sake of simplicity for controller design in the following sections, the control variable $u(t)$ is incorporated into the state variables. Therefore, the model representation of the system can be decomposed as:

$$\frac{d}{dt} \begin{bmatrix} u(t) \\ a_{1n}(t) \\ a_{2n}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \Delta_n \end{bmatrix} \begin{bmatrix} u(t) \\ a_{1n}(t) \\ a_{2n}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ -\beta_n \\ 0 \end{bmatrix} \tilde{u}(t) \tag{4.35}$$

For this simulation, We consider the parameters $\alpha = 0.05$ and $c = 1$. The operating steady state $w_s(z)$ with these values of parameters is verified to be a stable one. We enforce the input constraints on the first-order derivative of control variable, i.e., $\tilde{u}$, and the state constraints on $x_2(z,t) = \partial w(z,t)/\partial t$, which is the vertical velocity of the beam. The input and state constraints are given by the following:

$$\tilde{\mathcal{U}}^{min} \leq \tilde{u} \leq \tilde{\mathcal{U}}^{max}$$

$$\mathcal{X}^{min} \leq \int_0^1 r(z)x_2(z,t)dz \leq \mathcal{X}^{max} \tag{4.36}$$

The state constraints distribution function, $r(z)$, is given as $r(z) = \delta(z - z_c)$ for $z \in [0,1]$ and $z_c = 1$, which means that state constraints are only to be actualized at the single point of the right boundary. The first 15 pairs of modes are used to approximate the plant model and the first 3 pairs of modes are considered to be the dominant ones, which means that the slow subsystem contains the control variable u and the first 3 pairs modes and the fast subsystem includes 12 pairs modes, i.e., the slow modes are $a_s(k) = [u(k)\ a_{11}(k)\ a_{21}(k)\ \cdots\ a_{13}(k)\ a_{23}(k)]^T$ and the fast modes are $a_f(k) = [a_{14}(k)\ a_{24}(k)\ \cdots\ a_{115}(k)\ a_{215}(k)]^T$.

Remark 10 *Although the operating steady state $w_s(z)$ is stable, the whole system is unstable because the input variable u is incorporated into the state vectors. In other words, the eigenvalues of the discrete-time system consist of the eigenvalues of PDE system which are stable and the unstable eigenvalue which is associated with the input variable $u(k)$.*

The explicit MPC formulation of Eq. (4.24) is implemented on the discrete-time beam system. The control objective are to compute $\tilde{u}(k)$ as an explicit piecewise affine function of the eigen-coefficient $a(k)$, and enforce the input and state constraints. The initial condition of the PDE state is considered to be

$$x_1(z, t=0) = \begin{cases} 0.4z, & z \in [0, 0.5) \\ -0.4z + 0.4, & z \in [0.5, 1] \end{cases}$$

We consider $Q = 0.1I$, $R = 0.001$, $N = 4$ as the MPC parameters. The input and PDE state constraints are considered as $\tilde{\mathcal{U}}^{min} = -3$, $\tilde{\mathcal{U}}^{max} = 3$, $\mathcal{X}^{min} = -1$, $\mathcal{X}^{max} = 0.8$. The closed-loop evolutions of the beam displacement $w(z,t)$ and the beam velocity $\partial w(z,t)/\partial t$ under the implementation of the explicit MPC law given by Eq. (4.24) are given

in Fig. 4.5 and Fig. 4.6, respectively. The constrained PDE state $x_2(z=1,t)$ of the closed-loop system is shown in Fig. 4.7, and the constrained optimal control move $\tilde{u}$ computed by the explicit MPC law is given in Fig. 4.8. It is clearly demonstrated that both the PDE state and the input constraints are guaranteed with the explicit predictive controller. The corresponding boundary input profile is given in Fig. 4.9. The resulting explicit MPC controller consists of 7 critical regions; due to space limitations, we present here only the mathematical representation of the control law associated with the first three critical regions:

$$\tilde{u}^*(k) = \begin{cases} -3, & \text{region 1} \\ -4.66a_{11} - 0.45a_{12} + 0.15a_{21} + 0.14a_{22} - 0.02a_{13} - 0.05a_{23}, & \text{region 2} \\ 3, & \text{region 3} \\ \vdots & \vdots \end{cases} \tag{4.37}$$

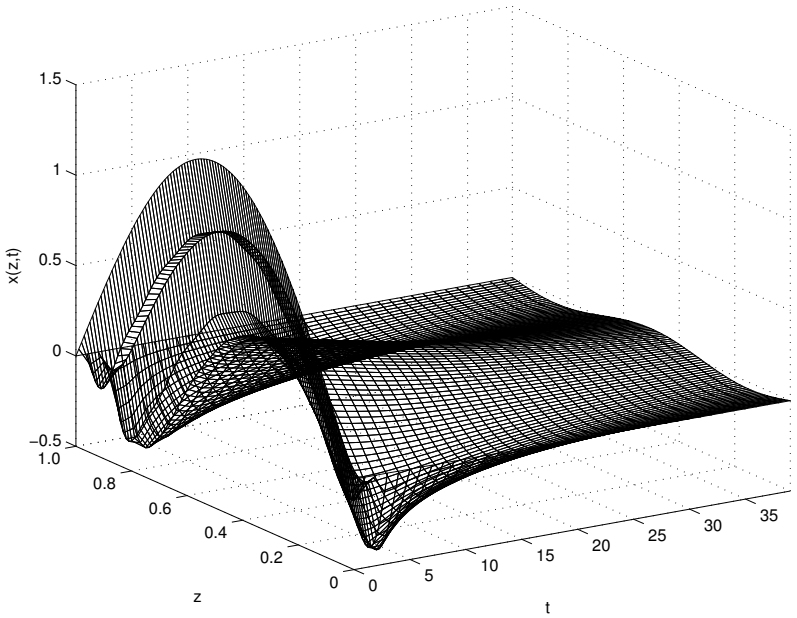


Figure 4.1: Profile of the PDE state $x(z,t)$ under the explicit MPC law in simulation 1.

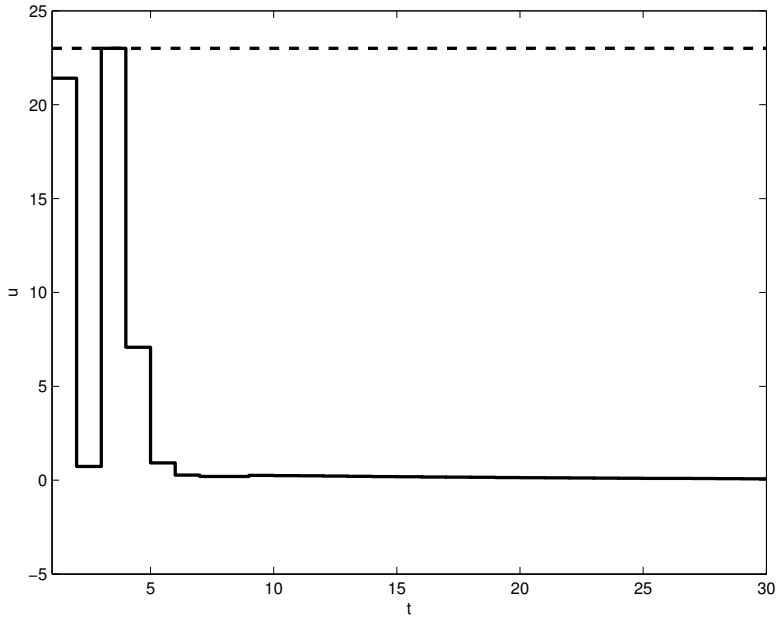


Figure 4.2: Profile of constrained control input u computed by the explicit MPC law in simulation 1 (solid-line); input constraints in simulation 1 (dashed line).

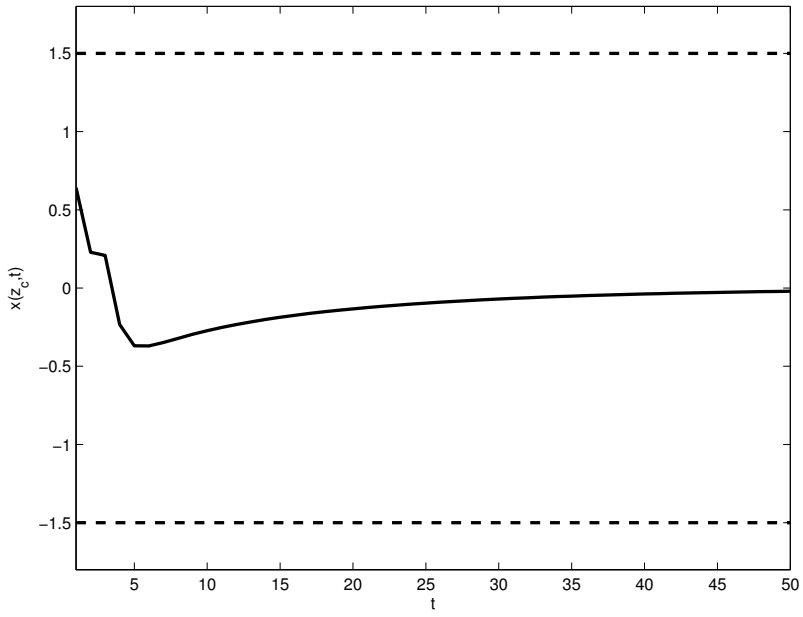


Figure 4.3: Profile of constrained PDE state $x(z = 0.16, t)$ and $x(z = 0.86, t)$ under the explicit MPC law in simulation 1 (solid line); state constraints in simulation 1 (dashed line).

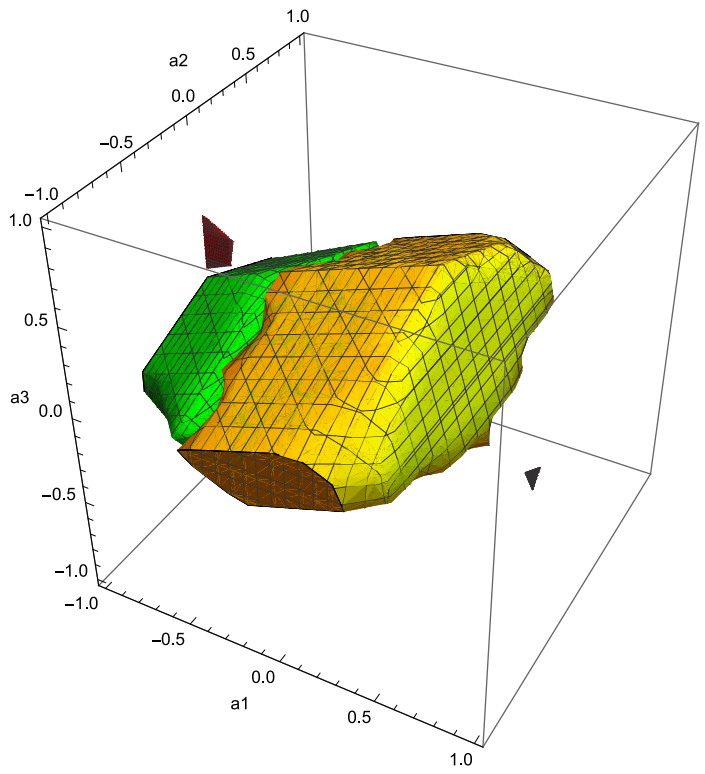


Figure 4.4: Critical regions for the solution of the explicit MPC law in simulation 1.

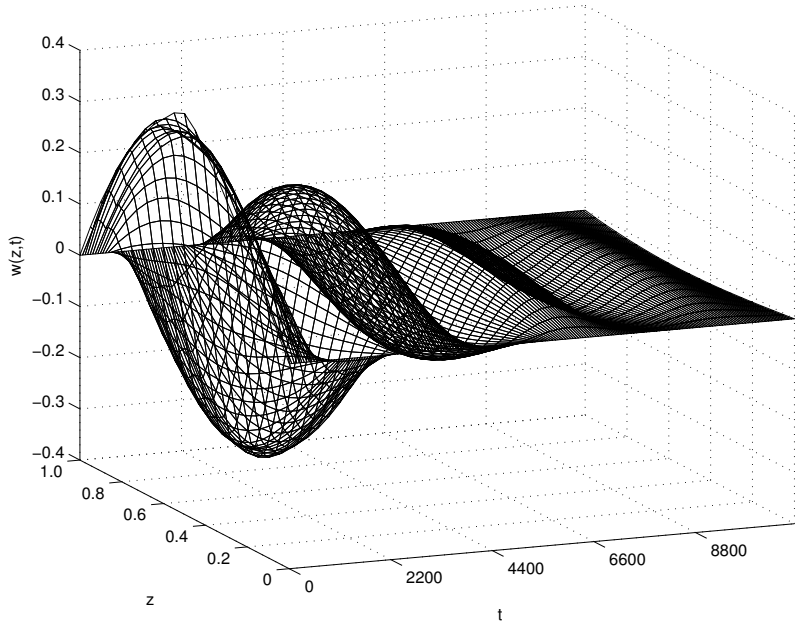


Figure 4.5: Profile of the beam displacement $w(z,t)$ under the explicit MPC law in simulation 2.

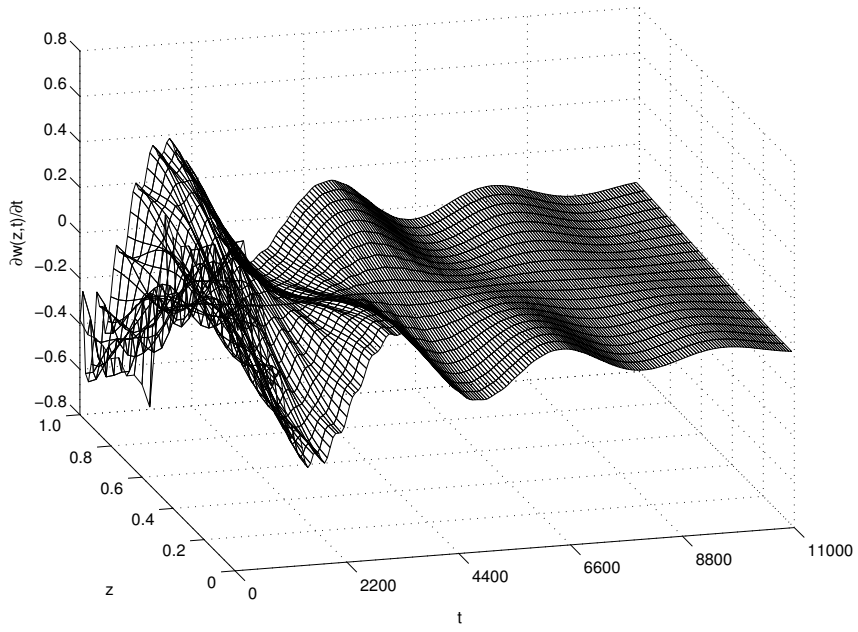


Figure 4.6: Profile of the beam velocity $\partial w(z,t)/\partial t$ under the explicit MPC law in simulation 2.

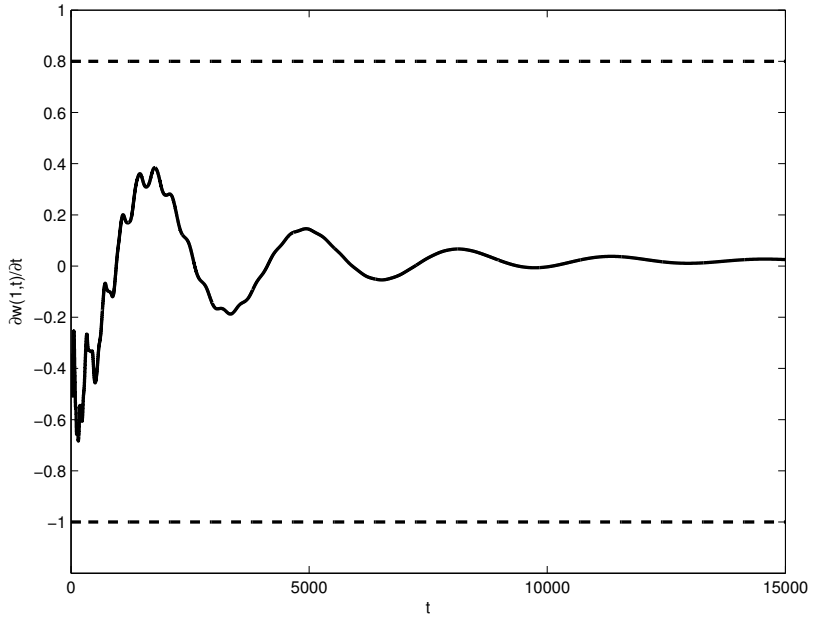


Figure 4.7: Profile of the constrained PDE state $\partial w(z=1,t)\partial t$ under the explicit MPC law in simulation 2 (solid line); state constraints in simulation 2 (dashed line).

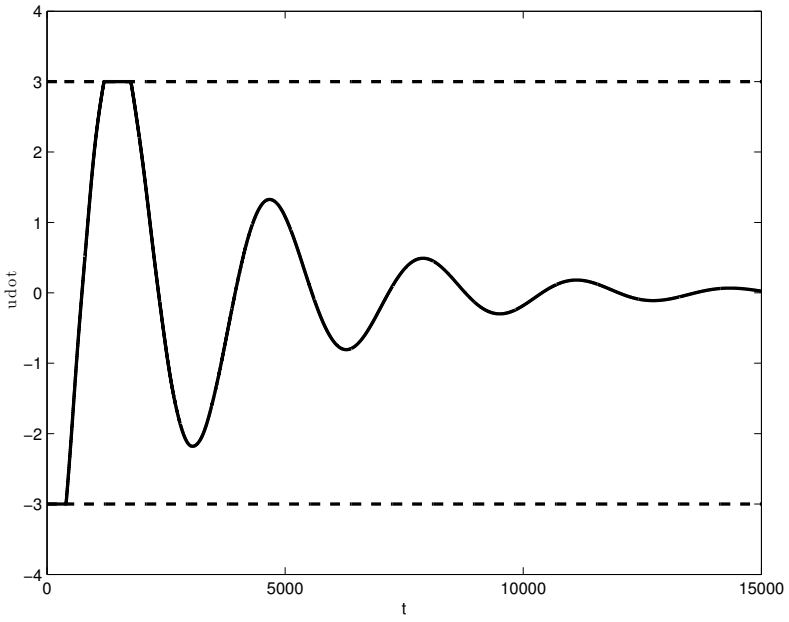


Figure 4.8: Profile of the constrained first-order derivate of control variable $\tilde{u}$ computed by the explicit MPC law in simulation 2 (solid line); constraints of $\tilde{u}$ in simulation 2 (dashed line).

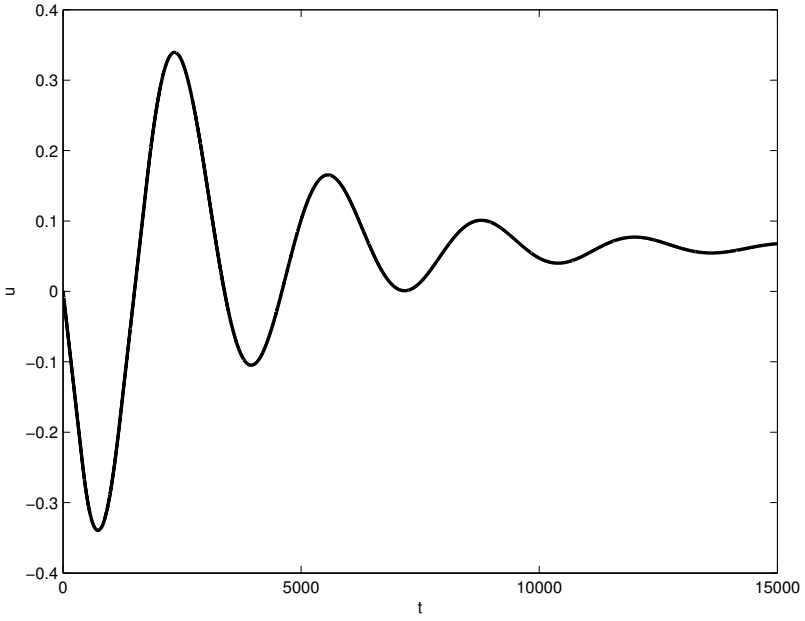


Figure 4.9: Profile of boundary input u in simulation 2.

4.5 Conclusions

In this chapter, an explicit/multi-parametric MPC law is developed in this work for the certain class of dissipative distributed-parameter system with consideration of manipulated input and state variable constraints. We consider the linear infinite-dimensional system and develop a finite-dimensional system representation capturing the dominant dynamics of the PDE system through Galerkin's method and modal decomposition techniques, which is utilized as the basis for reduced-order predictive controller design. An explicit/multi-parametric model predictive control formulation is proposed in a way that the low-order modes contribute to both the objective function and the PDE state constraints, while the higher-order modes contribute only to the PDE state constraints. In particular, the infinity norm of the higher-order modes is adopted to express the explicit higher-order modes evolution in the PDE state constraints. The explicit MPC formulation is solved through multi-parametric quadratic programming and multi-parametric dynamic programming methods. Finally, the performance of the proposed explicit MPC is demonstrated via its implementation on the transport-reaction system and the Euler-Bernoulli beam system.

Chapter 5

Conclusions

In this thesis, model predictive control taking into account manipulated input and state variable constraints are presented for distributed parameter systems. Furthermore, its performance are illustrated by transport-reaction process examples and flexible Euler-Bernoulli beam example.

Specifically, Chapter 2 proposed four model predictive control algorithms for axial dispersion chemical reactor system described by parabolic PDEs system with consideration of manipulated input and PDE state constraints. The spatial operator of this nonlinear parabolic PDEs is non-self-adjoint, and the eigenvalue problem of the operator is addressed by transforming the operator into the Sturm-Liouville form. The model predictive control algorithms are designed on the basis of low-order model representations which are derived through Galerkin's method and modal decomposition techniques. Standard discretization method and Tustin's discretization method with Cayley transform are adopted to transform the continuous-time model into discrete-time equivalent. The transfer function of the PDEs system is also derived in order to apply Cayley transform. The proposed MPC formulations differ in the way that the fast dynamics are involved in the objective function and PDE state constraints.

In order to implement the proposed MPC algorithms on other classes of distributed

parameter systems, in Chapter 3, the performance of the proposed modal model predictive control formulations are illustrated by flexible Euler-Bernoulli beam system which describes the vertical motion of a thin horizontal beam with small displacements from rest. The flexible beam system is given by a fourth order PDE system, which is neither parabolic nor hyperbolic. The fourth order spatial operator is decomposed by a particular projection method and modal decomposition techniques. Then a finite-dimensional model representation that captures the dominant dynamics of the PDE system is obtained and utilized for subsequent predictive controller design. The transfer function of the fourth order PDE system is derived to obtain Tustin's discrete-time system. The input and PDE state constraints are guaranteed by accounting for the fast modes in state constraints equation.

In Chapter 4, the linear infinite-dimensional system arising from modeling of the certain class of dissipative distributed parameter system is considered. An explicit/multi-parametric MPC algorithm is proposed in a way that the infinity norm of fast modes are accounted for in the PDE state constraints and only slow modes are involved in objective function. The shortcomings of regular MPC is overcome by explicit/multi-parametric MPC, which gives the control law as an explicit function of the state variable with its corresponding critical regions. Multi-parametric quadratic programming and multi-parametric dynamic programming methods are used to solve the explicit MPC formulation. Performance of the proposed explicit MPC law is illustrated via its implemention on the transport-reaction system and the flexible Euler-Bernoulli beam system through simulation studies. The proposed explicit MPC formulation is proved to guarantee input and state constraints and reduce computational effort as well.

Bibliography

[1] R. F. Curtain, H. Zwart, An Introduction to Infinite-dimensional Linear Systems Theory, Vol. 21, Springer, New York, 1995.

[2] M. J. Balas, Feedback control of linear diffusion processes, International Journal of Control 29 (3) (1979) 523–534.

[3] R. Curtain, Finite-dimensional compensator design for parabolic distributed systems with point sensors and boundary input, Automatic Control, IEEE Transactions on 27 (1) (1982) 98–104.

[4] P. D. Christofides, P. Daoutidis, Finite-dimensional control of parabolic PDE systems using approximate inertial manifolds, in: Decision and Control, 1997., Proceedings of the 36th IEEE Conference on, Vol. 2, IEEE, 1997, pp. 1068–1073.

[5] A. Armaou, P. D. Christofides, Dynamic optimization of dissipative PDE systems using nonlinear order reduction, Chemical Engineering Science 57 (24) (2002) 5083–5114.

[6] J. Baker, P. D. Christofides, Finite-dimensional approximation and control of nonlinear parabolic PDE systems, International Journal of Control 73 (5) (2000) 439–456.

[7] P. D. Christofides, Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport Reaction Processes, Springer, New York, 2001.

[8] J. L. Lions, S. K. Mitter, Optimal control of systems governed by partial differential equations, Vol. 1200, Springer, Berlin, 1971.

[9] C. Kubrusly, H. Malebranche, Sensors and controllers location in distributed systems: a survey, Automatica 21 (2) (1985) 117–128.

[10] W. Waldraff, D. Dochain, S. Bourrel, A. Magnus, On the use of observability measures for sensor location in tubular reactor, Journal of Process Control 8 (5) (1998) 497–505.

[11] M. A. Demetriou, N. Kazantzis, A new actuator activation policy for performance enhancement of controlled diffusion processes, Automatica 40 (3) (2004) 415–421.

[12] N. El-Farra, M. Demetriou, P. Christofides, Actuator and controller scheduling in nonlinear transport-reaction processes, Chemical Engineering Science 63 (13) (2008) 3537–3550.

[13] P. Dufour, Y. Touré, D. Blanc, P. Laurent, On nonlinear distributed parameter model predictive control strategy: on-line calculation time reduction and application to an experimental drying process, Computers & Chemical Engineering 27 (11) (2003) 1533–1542.

[14] H. Shang, J. F. Forbes, M. Guay, Model predictive control for quasilinear hyperbolic distributed parameter systems, Industrial & Engineering Chemistry Research 43 (9) (2004) 2140–2149.

[15] S. Dubljevic, N. H. El-Farra, P. Mhaskar, P. D. Christofides, Predictive control of parabolic PDEs with state and control constraints, International Journal of Robust and Nonlinear Control 16 (16) (2006) 749–772.

[16] S. Dubljevic, P. D. Christofides, Predictive control of parabolic PDEs with boundary control actuation, Chemical Engineering Science 61 (18) (2006) 6239–6248.

[17] S. Dubljevic, P. D. Christofides, Predictive output feedback control of parabolic partial differential equations (PDEs), Industrial & Engineering Chemistry Research 45 (25) (2006) 8421–8429.

[18] D. Ni, P. D. Christofides, Multivariable predictive control of thin film deposition using a stochastic PDE model, Industrial & Engineering Chemistry Research 44 (8) (2005) 2416–2427.

[19] M. R. García, C. Vilas, L. O. Santos, A. A. Alonso, A robust multi-model predictive controller for distributed parameter systems, Journal of Process Control 22 (1) (2012) 60–71.

[20] I. Lasiecka, R. Triggiani, Exact controllability of the wave equation with Neumann boundary control, Applied Mathematics and Optimization 19 (1) (1989) 243–290.

[21] V. A. Il'in, Two-endpoint boundary control of vibrations described by a finite-energy generalized solution of the wave equation, Differential Equations 36 (11) (2000) 1659–1675.

[22] O. Morgul, Dynamic boundary control of a Euler-Bernoulli beam, Automatic Control, IEEE Transactions on 37 (5) (1992) 639–642.

[23] B. Z. Guo, Riesz basis property and exponential stability of controlled Euler–Bernoulli beam equations with variable coefficients, SIAM Journal on Control and Optimization 40 (6) (2002) 1905–1923.

[24] L. Lao, M. Ellis, P. D. Christofides, Economic model predictive control of parabolic PDE systems: Addressing state estimation and computational efficiency, Journal of Process Control 24 (4) (2014) 448–462.

[25] A. Armaou, A. Ataei, Piece-wise constant predictive feedback control of nonlinear systems, Journal of Process Control 24 (4) (2014) 326–335.

[26] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (1) (2002) 3–20.

[27] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit solution of model predictive control via multi-parametric quadratic programming, in: American Control Conference, 2000. Proceedings of the 2000, Vol. 2, IEEE, 2000, pp. 872–876.

[28] E. N. Pistikopoulos, M. C. Georgiadis, V. Dua, Multi-Parametric Programming: Theory, Algorithms and Applications, Volume 1, Wiley-VCH, Weinheim, 2007.

[29] E. N. Pistikopoulos, M. C. Georgiadis, V. Dua, Multi-Parametric Model-Based Control: Theory and Applications, Volume 2, Wiley-VCH, Weinheim, 2007.

[30] E. N. Pistikopoulos, Perspectives in multipara-metric programming and explicit model predictive control, AIChE Journal 55 (8) (2009) 1918–1925.

[31] K. I. Kouramas, N. P. Faísca, C. Panos, E. N. Pistikopoulos, Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming, Automatica 47 (8) (2011) 1638–1645.

[32] D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. 1, Athena Scientific Belmont, MA, 1995.

[33] N. P. Faísca, K. I. Kouramas, P. M. Saraiva, B. Rustem, E. N. Pistikopoulos, A multi-parametric programming approach for constrained dynamic programming problems, Optimization Letters 2 (2) (2008) 267–280.

[34] W. H. Ray, Advanced Process Control, McGraw-Hill, New York, 1981.

[35] V. Havu, J. Malinen, The Cayley transform as a time discretization scheme, Numerical Functional Analysis and Optimization 28 (7-8) (2007) 825–851.

[36] N. C. Besseling, Stability Analysis in Continuous and Discrete Time (Ph.D. dissertation), Department of Applied Mathematics, University of Twente, Enschede, Netherlands, 2012.

[37] K. R. Muske, J. B. Rawlings, Model predictive control with linear models, AIChE Journal 39 (2) (1993) 262–287.

[38] S. Dubljevic, Model predictive control of Kuramoto–Sivashinsky equation with state and input constraints, Chemical Engineering Science 65 (15) (2010) 4388–4396.

[39] J. C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Siam, 2004.

[40] R. Curtain, Finite dimensional compensators for some hyperbolic systems with boundary control, in: Control Theory for Distributed Parameter Systems and Applications, Springer, 1983, pp. 77–91.

[41] L. Liu, B. Huang, S. Dubljevic, Model predictive control of axial dispersion chemical reactor, Journal of Process Control. 24 (11) (2014) 1671–1690.

[42] K. I. Kouramas, N. P. Faísca, C. Panos, E. N. Pistikopoulos, Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming, Automatica 47 (8) (2011) 1638–1645.

[43] E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, M. Morari, On-line optimization via off-line parametric optimization tools, Computers & Chemical Engineering 26 (2) (2002) 175–185.