The Pennsylvania State University
The Graduate School
College of Engineering

DEVELOPMENT OF EFFICIENT MODELING AND DIFFUSION CONTROL APPROACHES OF NONLINEAR SPATIALLY DISTRIBUTED PROCESSES

A Dissertation in
Chemical Engineering
by
Manda Yang

© 2018 Manda Yang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

April 2018

The dissertation of Manda Yang was reviewed and approved* by the following:

Antonios Armaou
Associate Professor of Chemical Engineering
Dissertation Advisor
Chair of Committee

Ali Borhan
Professor of Chemical Engineering

Constantino Lagoa
Professor of Electrical Engineering

Rob Rioux
Friedrich G. Helfferich Professor of Chemical Engineering and Professor of Chemistry

Kristen Fichthorn
Merrell Fenske Professor of Chemical Engineering and Professor of Physics

*Signatures are on file in the Graduate School.

Abstract

The distributed parameter system (DPS) is of significant importance in many chemical and material industry processes. This kind of systems shows spatial variation because of the existence of diffusion, convection and reaction. Due to the motivation to improve product quality and increase economic profit, controlling these processes is significantly important. Examples include plasma enhanced chemical vapor deposition, plasma etching reactors and reaction in porous catalyst particles.

A standard way to control these processes is to construct reduced order models (ROMs) via the Galerkin method and then design observers and controllers based on the ROMs, given the fact that the behavior of most of the above chemical and material industry processes can be captured by finite dimensional systems. To improve the accuracy of the ROM and reduce the computational cost, many research studies have been done. The proposed work will relax the assumptions of existing methods. The contribution of this work can be summarized to: (1) proposing an advanced POD method that combines the standard POD method and the analytical approach; (2) combining Adaptive Proper Orthogonal Decomposition (APOD) and Discrete Empirical Interpolation Method (DEIM) to reduce the computational cost of ROMs; (3) proposing Discrete Adaptive Proper Orthogonal Decomposition (DAPOD) to relax the assumptions of APOD and improved its performance; (4) improving the accuracy of ROM for systems with strong convection using DAPOD; (5) proposing an equation-free control method that can control PDE based on model reduction when the governing equation of the system is not available.

Table of Contents

List of Figures

List of Tables

List of Symbols

Abbreviations

β_T	dimensionless heat of reaction
β_u	dimensionless heat transfer coefficient
$\mathcal{A}$	a linear operator
$\mathcal{L}$	linear spatial operator
$\mathfrak{A}$	spatial operator
$\mathfrak{f}$	discretized f
$\mathfrak{g}$	freshness vector
$\mathfrak{h}$	importance vector
$\mathfrak{L}$	Lagrange function
$\mathfrak{q}$	$\mathfrak{q}_j = \mathfrak{h}_j + K\mathfrak{g}_j$
G	a matrix used in Passthrou POD
V	the eigenvector matrix of G
v	the eigenvector of G
V_l	the reduced eigenvector matrix
W	passthrou basis function matrix
w	passthrou basis function
W_l	reduced passthrou basis function matrix
$\tilde{c}$	estimated c by the observer

$\tilde{y}$	estimated y by the observer
$\tilde{y}$	estimated output by the observer
A	snapshot matrix
a	a snapshot
B	matrix in ROM that corresponding to b
C	coefficient matrix
c	coefficient in MWR
C_k	covariance matrix
C_q	$C_q = QC_kQ$
d	$g \approx Ud$
D, D_b	matrices used in the observer based on DEIM/APOD combination
f	$f = \mathcal{L}(z)x + g(x)$
G	matrix in ROM that corresponding to g
g	nonlinear function
H	$H = V_l^T C_k V_l$
H	Heaviside step function
I	identity matrix
K	a coefficient that indicates the importance of keeping recent snapshots
k	number of interpolation indices
K_g	the observer gain matrix
L	Lie derivative
L_o	matrix in ROM that corresponding to $\mathcal{L}$
L_r	length of rod
L_O	an ODE
L_P	a PDE
M	number of grid points

m number of basis functions in DAPOD

N number of snapshots

n number of basis functions

n_g size of G

n_y number of measured output

P matrix that contains interpolation indices

Q $Q = I - V_l V_l^T$

r residual in MWR

t time

U nonlinear basis function matrix

u manipulated variables

V eigenvector matrix

v eigenvector $v = A^T \phi$

V_l eigenvector matrix corresponding to large eigenvalues

w weight function

x the state variable

x_0 the desired point

y output

y_m measured output

z spatial coordinate

z' spatial coordinate

APOD Adaptive Proper Orthogonal Decomposition

DAPOD Discrete Adaptive Proper Orthogonal Decomposition

DEIM Discrete Empirical Interpolation Method

MWR Method of Weighted Residues

ODE ordinary differential equation

PCA Principal Component Analysis

PDE partial differential equation

POD Proper Orthogonal Decomposition

ROM reduced order model

SVD Singular Value Decomposition

Greek Symbols

Δz grid interval

δ_{ij} Kronecker delta

γ dimensionless activity energy

ι Lagrange multiplier/eigenvalue

ι_l large eigenvalue

Λ eigenvalue matrix of C_k

λ eigenvalue

Λ_n eigenvalue matrix corresponding to n eigenvalues

$\mathsf{\Lambda}$ the eigenvalue matrix of G

τ fast time scale

μ eigenvalue

Ω_z domain of z

Φ basis function matrix

ϕ basis function

Φ_g basis function matrix corresponding to the n_g largest eigenvalues

Φ_l basis function matrix corresponding to large eigenvalues

$\tilde{\phi}$ nonlinear basis function

ε energy threshold

ξ ratio of energy captured by basis functions

Superscripts

T	transpose

Acknowledgments

I'm fortunate to get support from many people. I apologize for not listing all of them.

My thanks go first to my dissertation advisor Dr. Antonios Armaou, for his support and guidance for my research work. I'm impressed by his enthusiasm for math.

I would like to thank Dr. Rioux, Dr. Lagoa, and Dr. Borhan for serving on my committee and giving me advice.

I really appreciate the patience, and support given by my parents.

I would also like to thank Dr. Jun Wang, Dr. Rong Zhong, Dr. Pengyi Zhang, Dr. Li Zhao, Dr. Arthur Small, Paul Sloan, Jingwei Li, Xinyi Yu, Dr. Yuexiao Shen, Adam Gibbons, Wenjun Jiang, Laura Anderson, Margaret Oakar, William Fritz, Simo Wu, Zehao Zhang, Baihua Xie, Ben Heidorn, John Camin, Tingwei Ren, Dr. Negar Hashemian, Mohammad Mazharul Islam, Yizhou Fang, Dr. Zifeng Li, Dr. Liwei Li, Xiaoduo Qi for their help.

Financial support from the National Science foundation, CMMI award #13-00322 is gratefully acknowledged

Chapter 1 Introduction

1.1 Modeling and Control of Distributed Parameter System

In recent years, control of distributed parameter systems (DPS) has been an important issue in many chemical industry process due to spatial variation as a result of diffusion, convection and chemical reaction. Some of the examples include plasma enhanced chemical vapor decomposition [2], polymerization [3], plasma etching reactors [4] and reaction in porous catalyst particles [5].

One of the important application is the plasma etching in semiconductor manufacturing. To reduce the variance of the product, the uniformity in the reactor is desired; while maintaining the uniformity becomes more difficult as the diameter of wafers is larger than 200 nm [6].

The controller design for DPS is nontrivial since the state varies in both time and space, which makes the theoretical problem infinite dimensional. The infinite dimension issue can be partially addressed by finite volume method (FV) or finite difference method (FD) discretization. However, the resulting system is usually high dimensional, making controller design complex and computationally intensive.

Another approach is to construct a reduced order model (ROM) using the method of weighted residuals (MWR) [7–9] that takes advantage of the property of dissipative PDEs that their behavior can be approximated by low dimensional systems [10, 11]. It approximates the state variable by superposition of basis functions multiplied by time dependent coefficients. Basis functions are analytically derived by solving the eigenproblem of the system's spatial operator. MWR

can create ROMs with lower dimension than FV and FD. However, solving the eigenproblem analytically is uncertain for systems with complex geometry and nonlinear operators.

One way to circumvent the above issue is to use proper orthogonal decomposition (POD) [12]. This method constructs basis functions numerically using an ensemble of data of state profile (snapshot) collected from experiment or simulation. Basis functions are determined by maximizing the inner product of the snapshots and the basis functions so that the dimension of the ROM can be reduced. As one of the primary methods to construct ROM, POD method is widely applied in many dynamic research areas, such as fluid flow [13–20], structural dynamics [21–23], internal combustion engine [24], aeroelasticity [25] and four-dimensional variational data assimilation [26]. POD method is also known as principal component analysis (PCA), Karhunen-Loève expansion and singular value decomposition (SVD).

Despite the advances of POD, a large ensemble of PDE solution profile or snapshots is required to generate basis functions that capture dominant behavior of the system. Even when a large ensemble of data is available, initial condition and input in experiment or simulation, sampling method, discretization method all have impact on the basis functions. Currently, there is no rigorous guarantees on whether the snapshots ensemble is properly collected or not [27]. Another drawback is when the process visit another region in state space, basis function might not capture the new trend.

To circumvent the limitations and robustify POD, some new methods and modifications to POD have been proposed, including balanced POD [20], trust region POD [16], dynamic mode decomposition [28], global POD [29], incremental SVD/POD [30] and Adaptive POD (APOD) [31].

1.2 Mathematical Formulation

We consider processes described by the following semi-linear PDE:

$$\frac{\partial x}{\partial t} = \mathcal{L}(z)x + g(x) + b(z)u = f(x) + b(z)u, \tag{1.1}$$

where $f(x) = \mathcal{L}(z)x + g(x)$, subject to the boundary condition

$$h(x, \frac{\partial x}{\partial z}) = 0 \ on \ \Gamma \tag{1.2}$$

and the initial condition

$$x(z, 0) = x_0(z) \tag{1.3}$$

where $x \in \mathbb{R}$ is the state variable, t refers to time and $z \in \Omega_z \subset \mathbb{R}^3$ represents the spatial coordinate; $\mathcal{L}(z)$ is a linear spatial operator and $g(x)$ is a nonlinear function. $u \in \mathbb{R}^{s\times 1}$ denotes manipulated variables, where s denotes the number of manipulated inputs. $b(z) \in \mathbb{R}^{1\times s}$ describes how the manipulated variables u control the system spatially and is known. h is a function of x and its spatial derivative. Γ is the process boundary.

1.3 Research Objective

We propose to improve the accuracy, reduce the computational cost and relax the assumptions of existing method for dissipative PDE systems based on model reduction. To achieve this objective, the individual goals include:

- Develop an algorithm that combines the analytical way to determine basis functions and the statistical approach (POD) so that the new algorithm has the advantages of both methods.
- Relax some of the assumptions of APOD and further reduce the computational cost
- Improve the accuracy of ROM in system with strong convective phenomena.
- Develop a control framework based on model reduction when the governing equation of the system is not available

To evaluate the developed methods, controller and observer will be designed based on the proposed methods and implemented in examples via simulation.

1.4 Dissertation Structure

Chapter 2 explains the algorithm of MWR/Galerkin method and how they are used to construct ROM form PDE. The algorithm of POD and APOD is also introduced for completeness.

Chapter 3 is a version of M. Yang and A. Armaou, "Feedback control of semi-linear distributed parameter systems using advanced POD method," *2015 54th IEEE Conference on Decision and Control (CDC)*, Osaka, 2015, pp. 4680-4687, which focuses on the control of dissipative distributed parameter systems employing passthrou proper orthogonal decomposition (Passthrou POD) method. The objective of the approach is to develop a model order reduction and control method that is robust to model uncertainty and disturbances and relaxes some of the requirements of standard POD. The proposed method is successfully applied to regulate a representative diffusion-reaction process. To assess the performance of passthrou POD, both passthrou POD and standard POD are used to construct low dimensional reduced order models (ROMs). Linear feedback controllers and nonlinear dynamic observers are subsequently designed based on the obtained ROMs. Simulation results illustrate that passthrou POD method is more efficient and robust to model uncertainty, disturbances and ensemble construction.

Chapter 4 is a version of Manda Yang and Antonios Armaou, 2018, "Dissipative distributed parameter systems on-line reduction and control using DEIM/APOD combination", *Proceedings of the American Control Conference*, accepted. The output feedback control of distributed parameter systems based on adaptive model reduction is explored in this paper. A significant computational hurdle when using model reduction for control (MRC) is the numerical computation of integrals that appear in the reduced order model limiting MRC's applicability when dealing with nonlinearities. The objective of this paper is to further reduce the computational cost in APOD. It is addressed by using DEIM in the observer and controller to reduce the computational cost associated with the nonlinear functions. The proposed method is successfully illustrated on a diffusion reaction process and a fluid flow system described by the Kuramoto-Sivashinsky equation.

Chapter 5 is a version of Manda Yang and Antonios Armaou, 2017, "Revisiting APOD accuracy for nonlinear control of transport reaction processes: a spatially discrete approach", *Chemical Engineering Science*. This article addresses the

problem of output feedback control of dissipative distributed parameter systems. The reduced order model used for controller and observer synthesis is recursively updated using a revised version of adaptive proper orthogonal decomposition (APOD), based on decomposing spatially discrete solution profiles. This approach eliminates the basis size oscillation resulting from the inaccuracy of estimation of energy in APOD when the sampling speed is too slow. The performance of this method is illustrated by applying it to regulate a diffusion-reaction process and a fluid flow system described by the Kuramoto-Sivashinsky equation.

Chapter 6 is a version of Manda Yang and Antonios Armaou, 2016, "Control of dissipative PDE systems with strong convective phenomena based on model reduction", *Proceedings of the 22nd International Symposium on Mathematical Theory of Networks and Systemsl*. This article focuses on controlling dissipative distributed parameter systems with strong convective phenomena based on model reduction. The accuracy of the derived reduced order model (ROM) and the ROM-based observer may decrease as a system behavior switches from diffusion-dominant to convection-dominant. To mitigate the issue, we propose to use adaptive proper orthogonal decomposition (APOD) to capture the new behavior of the system. Reduced order model based Lyapunov-based control is applied to control the system in presence of disturbance and strong convective phenomena. Galerkin method is used to construct the reduced order model. The basis functions needed in Galerkin method are updated using APOD. A case study of regulating a tubular reactor at an open-loop unstable steady state is presented, where the performance of the ROM-based observer is evaluated. It is also demonstrated that APOD can improve the accuracy of the observer when an unexpected disturbance occurs.

Chapter 7 is a version of Manda Yang and Antonios Armaou, 2017, "On the design of equation-free controllers for dissipative PDEs via DEIM", *Proceedings of the American Control Conference*, which proposes an equation-free control method to control dissipative distributed parameter systems, in which the dynamics of the system are unknown while the effect of the control action is. A static observer is used to estimate the state using proper orthogonal decomposition (POD) so that a complete profile of the system can be estimated when a limited number of point sensors are available. Sensor locations are determined by interpolation indices in discrete empirical interpolation method (DEIM). By using both velocity and state sensors an explicit form of the complete equation become superfluous,

needing to only have a description of the actuator effect. The proposed method is successfully employed in a diffusion-reaction process with Dirichlet and Neumann boundary conditions. Feedback linearization is combined with the proposed method to regulate the system. Computational results demonstrate that this method can regulate a dissipative distributed parameter system without explicitly requiring a model of it and is robust to disturbances.

Chapter 8 is a version of Manda Yang and Antonios Armaou, 2017, "Synthesis of equation-free control structures for dissipative distributed parameter systems using proper orthogonal decomposition and discrete empirical interpolation methods", *Ind. Eng. Chem. Res*, in which we describe an equation-free control framework for the regulation of dissipative distributed parameter systems, with emphasis on improving the accuracy of the estimation by using a correction term. This control method is capable of regulating systems that have unknown dynamics but known effect of the control action. The system state and the dynamics are estimated by using the offline observations (snapshots ensemble) and the online continuous measurement of a restricted number of point sensors. First, we construct a reduced order model (ROM) with unknown terms using Galerkin/proper orthogonal decomposition (POD). Then the state of the ROM is estimated by a static observer with the information from the state sensors; and the mapping between the dynamics of the system and velocity sensors are generated using a similar approach. Discrete empirical interpolation method (DEIM) is employed to determine the sensor locations. To improve the accuracy of the estimation, a correction term is updated consistently. The proposed equation free control framework is illustrated through a diffusion-reaction process and the performance of the proposed method is evaluated by simulation.

Chapter 2
Reduced Order Model

2.1 Method of Weighted Residuals

Method of Weighted Residuals (MWR) makes an assumption that the solution of an ordinary differential equation (ODE) or partial differential equation (PDE) can be approximated by a superposition of basis function multiplied by coefficients. For an ODE

$$L_O(x(z)) = 0, \tag{2.1}$$

the coefficients c_i are constants; basis functions $\phi_i(z)$ depend on all the independent variables.

$$x(z) = \sum_{i=1}^{n} c_i \phi_i(z) \tag{2.2}$$

In PDE case,

$$L_P(x(z,t)) = 0 \tag{2.3}$$

the coefficients c_i are time dependent; basis functions $\phi_i(z)$ depend on all the independent variables except time.

$$x(z,t) = \sum_{i=1}^{n} c_i(t) \phi_i(z) \tag{2.4}$$

Since the solution is an approximation, the equation L_O or L_P is not 0. Take L_P for example,

$$L_P(x) = r \tag{2.5}$$

MWR determines the coefficients c by minimizing residual r in the sense that the

weighted averages of residual are 0

$$\int_{\Omega_z} L_P(c, \phi(z)) w_i(z) \mathrm{d}z = 0 \qquad i = 1, 2, \cdots, n \tag{2.6}$$

where w_i represents to a set of weight functions. The idea is if the projections of residual in all the subspaces are zero, residual itself will be zero.

In this algorithm, basis functions ϕ_i and weight functions w_i are predetermined. Coefficients c_i are solved by projecting original equation into subspaces spanned by weight functions. Based on how basis functions and weight functions are chosen, MWR describes several different methods, including subdomain method, point matching method, collocation method, least-squares method [32] and Galerkin method.

2.2 Galerkin Method

In Galerkin method, weight functions are chosen to be the same with basis functions and basis functions must satisfy all the boundary conditions. Therefore, it only applies to homogeneous problems. Using Galerkin methd, the order reduction problem is transformed to determination of basis functions. One way to determine the basis functions is to solve the eigenfunction problem analytically.

$$\mathfrak{A}\phi = \lambda\phi \tag{2.7}$$

where $\mathfrak{A}$ is the spatial operator in governing equation and ϕ is the basis function.

For the easy of exposition, we consider a linear PDE:

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} - 2x \tag{2.8}$$

with the boundary condition

$$x(0, t) = x(1, t) = 0 \tag{2.9}$$

Basis functions are determined by solving

$$\begin{aligned} &\frac{\partial^2 \phi}{\partial z^2} = \lambda \phi \\ &\phi(0) = \phi(1) = 0 \\ &\int_0^1 \phi_i \phi_j \mathrm{d}z = \delta_{ij} \end{aligned} \tag{2.10}$$

where δ_{ij} denotes Kronecker delta. We can obtain

$$\phi_i(z) = \sqrt{2}\sin(i\pi z) \tag{2.11}$$

$$\lambda_i = -(i\pi)^2 \tag{2.12}$$

With above basis functions, Eq. 2.8 is converted to a set of ODEs:

$$\begin{aligned} &\frac{\mathrm{d}c_1}{\mathrm{d}t} = (\lambda_1 - 2)c_1 \\ &\frac{\mathrm{d}c_2}{\mathrm{d}t} = (\lambda_2 - 2)c_2 \\ &\cdots \\ &\frac{\mathrm{d}c_n}{\mathrm{d}t} = (\lambda_n - 2)c_n \end{aligned} \tag{2.13}$$

It is clear that ODEs that contain small eigenvalues λ can be truncated since they decay to zero very fast.

2.3 Galerkin Method in Distributed Parameter System

To approximate the PDE with a set of ODEs, we substitute Eq. 2.4 into the governing equation (Eq. 1.1).

$$\sum_{i=1}^{n} \dot{c}_i \phi_i = \sum_{i=1}^{n} c_i \mathcal{L}(\phi_i) + g(\sum_{i=1}^{n} c_i \phi_i) + b(z)u \tag{2.14}$$

Multiply both sides by $\phi_j (j = 1, 2, \cdots, n)$ and integrate over the domain of z Ω_z:

$$\sum_{i=1}^{N} \int_{\Omega_z} \dot{c}_i \phi_i \phi_j \mathrm{d}z = \sum_{i=1}^{n} \int_{\Omega_z} c_i \mathcal{L}(\phi_i)\phi_j \mathrm{d}z + \int_{\Omega_z} g(\sum_{i=1}^{n} c_i \phi_i)\phi_j \mathrm{d}z + \int_{\Omega_z} b(z) u \phi_j \mathrm{d}z, \quad j = 1, 2, \cdots, n \tag{2.15}$$

When basis functions are orthonormal, Eq. 2.15 is simplified to

$$\dot{c}_j = \sum_{i=1}^{n} \int_{\Omega_z} c_i \mathcal{L}(\phi_i)\phi_j \mathrm{d}z + \int_{\Omega_z} g(\sum_{i=1}^{n} c_i \phi_i)\phi_j \mathrm{d}z + \int_{\Omega_z} b(z) u \phi_j \mathrm{d}z, \quad j = 1, 2, \cdots, n \tag{2.16}$$

Eq. 2.16 can be written in the matrix form:

$$\dot{c} = L_o c + G(c) + B(c)u \tag{2.17}$$

This method is based on the assumption that the dominant behavior of the system can be captured by finite number of modes, which is always satisfied in highly dissipative PDE systems [33].

2.4 Proper Orthogonal Decomposition

The POD algorithm [12, 34] constructs a set of basis functions that capture the dominant behavior of the system better than other basis functions of the same dimension. First a set of state profiles of the system $a(z, t)$ are collected experimentally or via numerical simulation. Each profile $a(z, t_n) = a_n(z)$ is defined as a snapshot. The POD basis functions are determined by solving the following optimization problem:

$$\max \frac{\langle (a_i, \phi)^2 \rangle_{i=1}^{N}}{\|\phi\|^2} \tag{2.18}$$

where $\langle \cdot \rangle$ denotes average, $(\cdot, \cdot)$ represents L^2-inner product: $(p, q) = \int_{\Omega_z} pq \mathrm{d}z$ and $\|p\| = \sqrt{(p, p)}$. To solve the optimization problem the following Lagrange function is considered:

$$\mathfrak{L} = \langle (a_i, \phi)^2 \rangle_{i=1}^{N} - \iota(\|\phi\|^2) \tag{2.19}$$

where ι is the Lagrange multiplier. Take the derivative with respect to ϕ, it returns:

$$2 \int \langle a_i(z') a_i(z) \rangle \phi(z') \mathrm{d}z' - 2\iota\phi(z) = 0 \tag{2.20}$$

If the snapshots are discretized in space using M spatial grid points, the above problem is replaced by a eigen decomposition problem of $M \times M$ matrix

$$AA^T\phi = \iota\phi \tag{2.21}$$

where $A = [a_1(z), a_2(z), \ldots, a_M(z)] \in \mathbb{R}^{M\times N}$. Usually M is large making solving this problem computationally intensive. To reduce the computational cost, both sizes are multiplied by A^T and $A^T\phi$ is solved first. It returns

$$A^TAA^T\phi = \iota A^T\phi \tag{2.22}$$

$A^T\phi$ is the eigenvector of $A^TA \in \mathbb{R}^{N\times N}$. Since in most cases $N \ll M$, solving the eigenvalue problem of A^TA is computationally cheaper than AA^T. After $v = A^T\phi$ is solved from 2.22, it is substituted into 2.21 to solve ϕ. Since AA^T is symmetric, the generated ϕ is orthogonal to each other ($\phi_i\phi_j^T = 0$ for any $i \neq j$). We multiply ϕ by proper coefficients to make it orthonormal. It returns

$$\Phi = AV\iota^{-\frac{1}{2}} \tag{2.23}$$

where $\iota = \text{diag}(\iota_1, \cdots, \iota_N)$, $V = [v_1, v_2, \cdots, v_N] \in \mathbb{R}^{N\times N}$ and $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]$

Basis functions corresponding to small eigenvalues are truncated since they represent basis functions that do not capture the major trend in the data ensemble and they are usually contaminated with significant round-off error, be it from noise in the snapshots or due to numerical calculations [35]. Therefore, the number of basis function is decreased to n.

$$\Phi_l = AV_l\iota_l^{-\frac{1}{2}} \tag{2.24}$$

where $\iota_l = \text{diag}(\iota_1, \cdots, \iota_n)$, $V_l = [v_1, v_2, \cdots, v_n] \in \mathbb{R}^{N\times n}$. It should be mentioned that given a set of snapshots, the basis functions can also be solved using singular value decomposition, which has the advantage being numerically more stable than POD.

2.5 Batch POD & incremental POD

In finite dimensional context, SVD and POD are mathematically equivalent [36]. Most research about POD method focuses on batch POD [37], which constructs basis functions based on given historic observations without updating ROM by incorporating new snapshots. The resulting ROM might be inaccurate if the historic observations are not properly collected. Compared with batch POD, incremental POD methods circumvents this issue by successively updating basis function and enabling dimension increase when new observation becomes available. Since solving eigen decomposition problem of covariance matrix recursively is computationally intensive, a significant amount of research has focused on designing incremental POD methods with less computational cost.

In the work of M. Singer and W. Green, adaptive POD (aPOD) was introduced to solve reaction-diffusion equation [38]. Several set of basis functions are predetermined. Selection of basis functions is based on the state of the system.

In the work of Chao Xu [39], recursive POD was proposed using perturbation theory. In this algorithm, approximation error norm of snapshots is used to determine whether basis functions are updated or not. The distinction in covariance matrix induced by new snapshots is considered to be perturbation. Basis functions are updated using perturbation in covariance matrix. However, an assumption that perturbed eigenvectors of covariance matrix are linear combinations of original eigenvectors was made without proof.

In 2013, the same group proposed another variant of POD, called incremental POD (iPOD) [40]. In this algorithm, new basis functions are assumed to be linear combinations of old basis functions and approximation error of new snapshot. Based on this assumption eigenvalue problem of a smaller matrix is solved to estimation new basis functions.

In the work of T. Braconnier [30], incremental SVD/POD method is proposed to construct ROM for aeronautic design. This algorithm employs rank-1 SVD update [30] to relax computational burden in updating basis function step. It also has the advantage that less memory storage is requested since historic snapshots are not stored.

In 2009, Adaptive POD (APOD) [31] was proposed. This method partition the eigenspace covariance matrix into 2 subspaces so that the eigenvalues of covariance

matrix can be estimated with less computational cost than solving eigenvalue problem of covariance matrix directly. In 2013, modified APOD [1] was proposed, which eliminates least important snapshot instead of oldest snapshot. Compared with original APOD, it has the advantage that important trend can be retained.

2.6 Adaptive Proper Orthogonal Decomposition

As a variant of POD, APOD updates basis functions when new snapshots become available. The flow chart of APOD is displayed in Fig. 2.1. This algorithm consists of 2 steps: off-line initial basis function construction and on-line basis function refinement. In the first step, standard POD [12] (in section 2.4) is exploited to construct initial basis function. A set of N snapshots $A = [a_1(z), a_2(z), \ldots, a_N(z)]$ are collected from experiment or numerical simulation. Covariance matrix $C_k \in \mathbb{R}^{N\times N}$ is defined as follows:

$$C_k = \int A^T A \mathrm{d}z \tag{2.25}$$

We solve the eigen decomposition problem of covariance matrix and obtain a set of orthonormal eigenvectors $V = [v_1, v_2, \ldots, V_l] \in \mathbb{R}^{N\times N}$ of C_k

$$C_k V = V\Lambda \tag{2.26}$$

n eigenvectors $V_l = [v_1, v_2, \ldots, V_l] \in \mathbb{R}^{N\times n}$ corresponding to n largest eigenvalues are used to construct initial basis functions such that the n eigenvalues μ_i satisfy:

$$\frac{\sum_{i=1}^{n} \mu_i}{\sum_{i=1}^{N} \mu_i} > 1 - \varepsilon \tag{2.27}$$

The initial basis functions are

$$\Phi = [\phi_1, \phi_2, \ldots, \phi_n] = A V_l \Lambda_n^{-\frac{1}{2}} \tag{2.28}$$

where $\Lambda_n = \mathrm{diag}([\mu_1, \mu_2, \ldots, \mu_n])$. These n basis functions capture $(1 - \varepsilon)$ energy in the snapshot ensemble.

In on-line step, basis functions are updated when new snapshots become available. First, the snapshot ensemble and covariance matrix are updated by eliminating old snapshots. There are two approaches to select this old snapshot: the newest

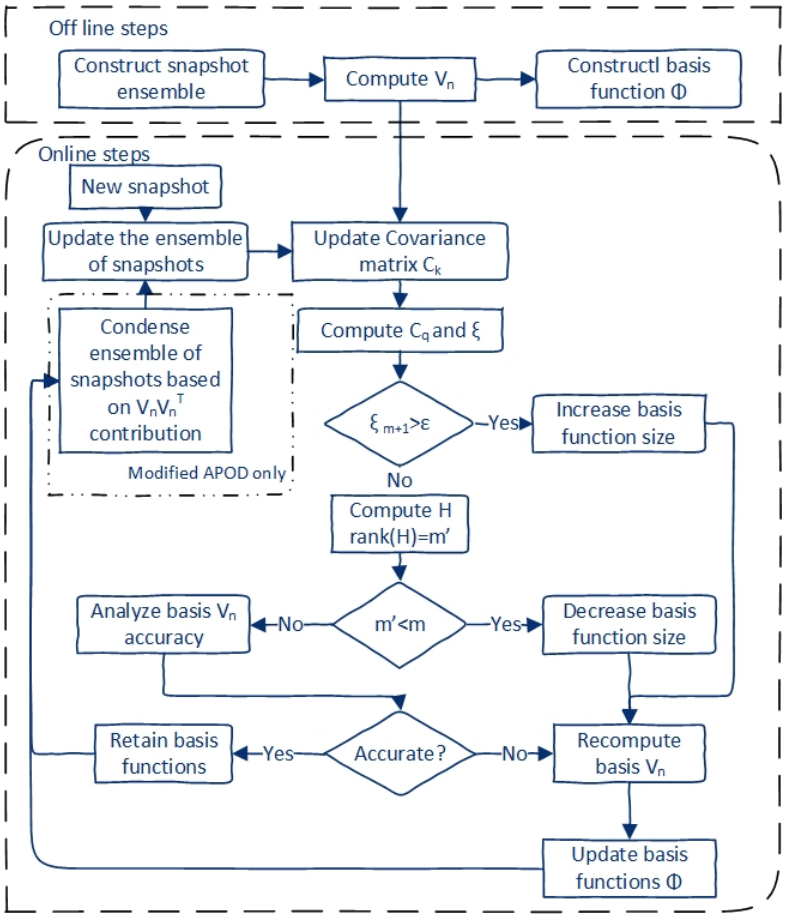


Figure 2.1: Flow chart of adaptive proper orthogonal decomposition [1]

snapshots approach [31] and the most important snapshots approach [1]. In the newest snapshots approach, the oldest snapshot is replaced by newest snapshot in each iteration. In the most important snapshots approach, the contribution of each snapshot in dominant eigenvectors of covariance matrix is compared. The snapshot corresponding to least contribution is replaced by newest snapshot. APOD taking the newest snapshots approach is called "original APOD" and the other one called "modified APOD".

After covariance matrix is updated, the size of basis functions used to capture the new trend may increase, decrease or remain unchanged. This is determined by the eigenvalues of covariance matrix. First, the ratio of energy captured by $(n+1)$th basis function need to be determined. The size of basis functions will increase or not accordingly. APOD assumes at most one basis function needs to be

added or deducted at each iteration. Since solving eigenvalue problem of covariance matrix can be computationally expensive, eigenvalues corresponding to n dominant basis function are estimated by eigenvalues of H:

$$H = V_l^T C_k V_l \tag{2.29}$$

and eigenvalue of $(n+1)$th basis function is estimated by the largest eigenvalue of C_q:

$$C_q = Q C_k Q \tag{2.30}$$

where

$$Q = I - V_l V_l^T \tag{2.31}$$

I denotes an identity matrix of proper size. If energy captured by $(n+1)$th basis function is less than ε, the basis size may decrease or remain the same. To determine whether there is any unnecessary basis function, the condition number of H is evaluated, based on which basis size decreases or is retained. If basis size neither increases nor decreases, accuracy of the eigenvectors is checked. Only when basis size is unchanged and eigenvectors are accurate, are the basis functions retained. In other cases, basis function need to be updated using economy singular value decomposition to maintain the accuracy of eigenvectors.

$$V_l = \mathrm{orth}(C_k V_l) \tag{2.32}$$

$$\Phi = A V_l \tag{2.33}$$

Then basis functions are scaled by proper scalar to make them orthonormal. It should be mentioned that since the new eigenvectors in V_l are only an estimation of eigenvectors of the covariance matrix, as a result, the basis functions generated by this algorithm are not strictly orthogonal.

Chapter 3
Feedback Control of Semi-linear Distributed Parameter Systems Using Advanced POD Method

In this chapter, we propose a new method called passthrou POD method to circumvent the issue that solving the eigenfunction problem of linear spatial operator analytically can be very complex in most systems. In this method standard POD and linear spatial operator analysis are combined. Then we take advantage of the concept of singular perturbations [41] to decompose a system into slow subsystem and fast complement with the slow part capturing the dominant phenomena of the system. Compared with snapshot POD method whose basis functions and order of importance are affected by the quality of ensemble of snapshots, passthrou POD circumvents this limitation by attaching more importance to the system. In other words, passthrou POD solves the modes excitation and number of snapshots problems in the standard POD method to properly order the basis functions. Simulation results show that controller and observer based on passthrou POD are also more robust than those based on standard POD. In particular, the quality of snapshots has much less impact on passthrou POD than standard POD.

3.1 Passthrou POD

The analytical approach (section 2.2) has the advantage that the full order system can be decomposed into slow subsystem and fast complement, but might be

difficult to compute; POD is easier to compute, but does not guarantee the fast-slow separation. Motivated by the advantage and disadvantage of the analytical approach and POD, we proposed passthrou POD that combines the two methods and has the advantages of both methods.

In passthrou POD, we first use standard POD (section 2.4) to construct POD basis functions ϕ. Then we construct matrix $\mathsf{G} \in \mathbb{R}^{n_g \times n_g}$ using ϕ.

$$\mathsf{G}_{ij} = \int_{\Omega_z} \phi(z)_i (\mathcal{A}\phi(z)_j) \mathrm{d}z \tag{3.1}$$

Where $\mathcal{A}$ is the linear operator with respect to space in the governing equation of linearized system at desired point x_0.

$$\mathcal{A} = \mathcal{L} + \left.\frac{\partial g}{\partial x}\right|_{(x=x_0)} \tag{3.2}$$

Then the diagonal eigenvalues matrix $\mathsf{\Lambda}$ and the column eigenvectors matrix V of G are identified.

$$\mathsf{GV} = \mathsf{V\Lambda} \tag{3.3}$$

Passthrou POD basis functions are computed as follows:

$$\mathsf{W}(z) = \Phi_g \mathsf{V} \tag{3.4}$$

where Φ_g contains basis functions corresponding to the n_g largest eigenvalues. $\mathsf{W} = [\mathsf{w}_1; \cdots; \mathsf{w}_{n_g}]$, and w_i denotes ith passthrou basis function.

To decrease the dimension of the reduced order model, eigenvectors with largest n eigenvalues are used to construct reduced order system. V becomes V_l, where $\mathsf{V}_l = [\mathsf{v}_1, \cdots, \mathsf{v}_n]$, n is the number of passthrou POD basis functions we will use. Eq. (3.4) becomes:

$$\mathsf{W}_l = \Phi_g \mathsf{V}_l \tag{3.5}$$

To justify this truncation of V, we will use singular perturbation method [11], [41] to illustrate how this method decomposes a system into slow subsystem and a fast complement one (that can be neglected when compared with slow subsystem), which allows us to truncate the basis functions corresponding to small eigenvalues of matrix G, and thus obtain W_l. For presentation simplicity, we use linearized

system here.

$$\frac{\partial x}{\partial t} = \mathcal{A}(x) + b(z)u \tag{3.6}$$

First, we substitute passthrou basis functions in Eq.(3.4) into the linearized system

$$x = \mathsf{W}C \tag{3.7}$$

where $C = [c_1, c_2, \cdots, c_{n_g}]^T \in \mathbb{R}^{n_g \times 1}$. We obtain:

$$\sum_{i=1}^{n_g} \dot{c}_i \mathsf{w}_i = \sum_{i=1}^{n_g} c_i \mathcal{A}(\mathsf{w}_i) + b(z)u \tag{3.8}$$

Multiply both sides by $\mathsf{w}_j (j = 1, 2, \cdots, n_g)$ and integrate over Ω_z

$$\begin{aligned} \sum_{i=1}^{n_g} \int_{\Omega_z} \dot{c}_i \mathsf{w}_i \mathsf{w}_j \mathrm{d}z = \sum_{i=1}^{n_g} \int_{\Omega_z} c_i \mathcal{A}(\mathsf{w}_i) \mathsf{w}_j \mathrm{d}z + \int_{\Omega_z} \mathsf{w}_j b(z) u \mathrm{d}z, \\ j = 1, \cdots, n_g \end{aligned} \tag{3.9}$$

It can be expressed in matrix form:

$$\int_{\Omega_z} \mathsf{W}^T \mathsf{W} \dot{C} \mathrm{d}z = \int_{\Omega_z} \mathsf{W}^T \mathcal{A}(\mathsf{W}) C \mathrm{d}z + \Big(\int_{\Omega_z} \mathsf{W}^T b \mathrm{d}z \Big) u \tag{3.10}$$

$$\int_{\Omega_z} \mathsf{V}^T \Phi_g^T \Phi_g \mathsf{V} \dot{C} \mathrm{d}z = \int_{\Omega_z} \mathsf{V}^T \Phi_g^T \mathcal{A}(\Phi_g) \mathsf{V} C \mathrm{d}z + \Big(\int_{\Omega_z} \mathsf{W}^T b \mathrm{d}z \Big) u \tag{3.11}$$

Since standard POD basis functions are orthonormal,

$$\int_{\Omega_z} \Phi_g^T \Phi_g \mathrm{d}z = I \tag{3.12}$$

where I denotes the identity matrix, we obtain,

$$\begin{aligned} \mathsf{V}^T \mathsf{V} \dot{C} &= \mathsf{V}^T \mathsf{G} \mathsf{V} C + \Big(\int_{\Omega_z} \mathsf{W}^T b \mathrm{d}z \Big) u \\ &= \mathsf{V}^T \mathsf{V} \mathsf{\Lambda} C + \Big(\int_{\Omega_z} \mathsf{W}^T b \mathrm{d}z \Big) u \end{aligned} \tag{3.13}$$

$$\begin{aligned} \dot{C} &= \mathsf{\Lambda} C + (\mathsf{V}^T \mathsf{V})^{-1} \Big(\int_{\Omega_z} \mathsf{W}^T b \mathrm{d}z \Big) u \\ &= \mathsf{\Lambda} C + \mathsf{B} u \end{aligned} \tag{3.14}$$

The eigenvalues in Λ are $\lambda_1, \cdots, \lambda_n, \lambda_{n+1}, \cdots, \lambda_{n_g}$. C can be decomposed as

$$C = [C_s; C_f] \tag{3.15}$$

Using this decomposition, Eq. (3.14) can be written in the form:

$$\begin{aligned} \dot{C}_s &= \Lambda_s C_s + \mathsf{B}_s u \\ \dot{C}_f &= \Lambda_f C_f + \mathsf{B}_f u \end{aligned} \tag{3.16}$$

where $\Lambda_s = \text{diag } \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$, which are the eigenvalues corresponding to eigenvectors in V_l. $\Lambda_f = \text{diag } \{\lambda_{n+1}, \lambda_{n+2}, \cdots \lambda_{n_g}\}$. $\text{Re}\lambda_1 \geq \text{Re}\lambda_2 \geq \cdots \geq \text{Re}\lambda_n$, and $0 \geq \text{Re}\lambda_{n+1} \geq \text{Re}\lambda_{n+2} \geq \cdots \geq \text{Re}\lambda_{n_g}$

Defining $\epsilon = |\text{Re}\lambda_1|/|\text{Re}\lambda_{n+1}| \ll 1$ and using ϵ, Eq.(3.16) can be written in the form:

$$\begin{aligned} \dot{C}_s &= \Lambda_s C_s + \mathsf{B}_s u \\ \epsilon \dot{C}_f &= \Lambda_{f\epsilon} C_f + \epsilon \mathsf{B}_f u \end{aligned} \tag{3.17}$$

where $\Lambda_{f\epsilon} = \epsilon \Lambda_f$. Note that the equation of C_f is multiplied by ϵ at both sides, the eigenspectra of Λ_s and $\Lambda_{f\epsilon}$ are of same order.

To investigate the fast dynamic behavior, we introduce a fast time-scale τ:

$$\epsilon \frac{dC}{dt} = \frac{dC}{d\tau} \tag{3.18}$$

and then set $\epsilon = 0$, eq.(3.17) becomes

$$\frac{dC_f}{d\tau} = \Lambda_{f\epsilon} C_f \tag{3.19}$$

Based on the fact that $0 \geq \text{Re}\lambda_{n+1} \geq \cdots \geq \text{Re}\lambda_{n_g}$ and the new time variable τ is stretched, we have that C_f decays to zero exponentially with a rate faster than the process of C_s. Therefore, using Tikhonov's theorem, the system state $c \approx c_s$ and the system dynamics can be approximated by

$$\dot{C}_s = \Lambda_s C_s + \mathsf{B}_s u \tag{3.20}$$

which is identical to the one obtained by passthrou POD method with only n basis

functions kept.

3.2 Observer and Controller

We assume that point measurements of the system state are continuously available using a finite number of sensors. To decrease the requirement on the number of sensors, a dynamic observer is designed. Observer gain matrix is obtained based on the linearized system so that the linearized system is stable. The structure of observer is as follows:

$$\begin{aligned}\sum_{i=1}^{N}\int_{\Omega_z}\dot{\tilde{c}}_i\phi_i\phi_j\mathrm{d}z =& \sum_{i=1}^{N}\int_{\Omega_z}\tilde{c}_i\mathcal{L}(\phi_i)\phi_j\mathrm{d}z+\int_{\Omega_z}g(\sum_{i=1}^{N}\tilde{c}_i\phi_i)\phi_j\mathrm{d}z+\int_{\Omega_z}b(z)u\phi_j\mathrm{d}z\\ &+K_g(y-W_{lm}\tilde{c}),\\ &j=1,2,\cdots,n\end{aligned} \tag{3.21}$$

where K_g is the observer gain matrix, $W_{lm}\in\mathbb{R}^{n_y\times n}$ and $W_{lm}\tilde{c}$ denotes the estimated y by the ROM, n_y is the number of measured output.

With the dynamic observer estimating all the states of the system, a linear state feedback regulator is then designed. The controller gain matrix is determined using LQR theory [42] with performance criterion

$$V=\int_0^t[C^T(\tau)QC(\tau)+u^T(\tau)Ru(\tau)]\mathrm{d}\tau \tag{3.22}$$

The gain matrices of both controller and observer are calculated based on the linearized system. In passthrou POD method, this linearized system is obtained from the slow subsystem of Eq. (3.20). Combining the linear regulator with the nonlinear observer and assuming the separation principle holds, a nonlinear output feedback controller is derived.

3.3 Diffusion-reaction process

To compare the performance of passthrou POD method and standard POD method, we employ both methods to design a controller and observer pair to stabilize a

system. By assessing how efficient and robust the controller and observer are, we can compare the performance of these two methods.

In this example, an elementary exothermic reaction takes place along the surface of a catalytic rod. The temperature at both ends is maintained at T_0. Cooling is used to control the temperature of the rod, which affects the interval $[0.3\pi, 0.7\pi]$ uniformly. There is one sensor to measure the temperature of the rod at point $z = \pi/6$. The governing equation is as follows:

$$\rho C \frac{\partial T}{\partial \tilde{t}} = k_1 \frac{\partial^2 T}{\partial z^2} - A k_2(z) \mathrm{e}^{-\frac{E}{RT}} c_0 \Delta H + \frac{k_3 l_s}{\Delta A}(T_c(z) - T) \tag{3.23}$$

where concentration c_0 and parameters ρ, C, ΔA, E, R, A, k_1, k_3, l_s are all constants. The objective is to force the temperature to a uniform unstable steady state $T = T_0$. To regulate the temperature, cooling (or heating), T_c is employed. T_c contains 2 components:

$$T_c = T_{c0} + T_c' \tag{3.24}$$

where T_{c0} is a cooling system determined during process design, chosen so that $T = T_0$ is a viable (albeit unstable) open-loop steady state and T_c' is the manipulated input.

New variables are defined to simplify Eq. 7.15. We obtain:

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_T(z)(\mathrm{e}^{-\gamma/(1+x)} - \mathrm{e}^{-\gamma}) + \beta_u(b(z)u - x) \tag{3.25}$$

$$y_m = \int_{\Omega_z} \delta(\frac{\pi}{6}) x(z) \mathrm{d}z \tag{3.26}$$

subject to the following boundary conditions

$$x(0,t) = 0, x(\pi,t) = 0 \tag{3.27}$$

and initial condition

$$x(z,0) = 0.5 \tag{3.28}$$

where

$$b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \tag{3.29}$$

We assume a spatial distribution on the activity of the catalyst.

$$\beta_T(z) = 16[\cos(z) + 1] \tag{3.30}$$

x denotes dimensionless temperature.

$$x = \frac{T - T_0}{T_0} \tag{3.31}$$

$H(\cdot)$ denotes Heaviside step function; β_T is dimensionless heat of reaction. β_u denotes the dimensionless heat transfer coefficient; γ is dimensionless activity energy. The values of parameters are $\beta_u = 2, \gamma = 2$. The initial condition is $x(z, 0) = 0.5$. Without controller, the system evolves to a spatially varying steady state, which is depicted in Fig. 3.1. As a result, we can conclude that the uniform steady state $x = 0$ is unstable. Thus, the objective of the controller becomes to force the system state to the spatially uniform steady state $x(z, t) = 0$.

3.3.1 Standard POD and Passthrou POD Basis Functions

To compare passthrou POD method and standard POD method, controller and observers are designed based on passthrou POD basis functions and standard ones. First, we construct standard POD basis functions and passthrou POD ones. 30 snapshots are generated from open loop system simulation with initial condition being $x(z, 0) = 0.5$. Each snapshot is discretized in space at 60 equally spaced locations. Matrix A is formed using these snapshots ($\mathsf{A}_{ij} = \int_{\Omega_z} a_i(z)a_j(z)\mathrm{d}z$). Eigenvalues of A are $[5042.8, 5.3, 1.3, 0.1, 0.0, \cdots]$. Since eigenfunctions corresponding to small eigenvalues are contaminated with significant round-off error [43], 4 standard POD basis functions are used to construct passthrou POD basis functions. In passthrou POD method, eigenvalues of G are $[2.584, -2.076, -7.531, -42.588]$, because the 4th eigenvalue is negative and one order larger than the other 3 eigenvalues, we keep 3 basis functions in passthrou POD algorithm. To compare the performce of controller and observer based on passthrou POD and standard POD, ROMs will be constructed using these 3 passthrou basis functions and first 3 dominant standard POD basis functions, respectively. These 2 sets of basis functions are compared in Fig. 3.2a-3.2c

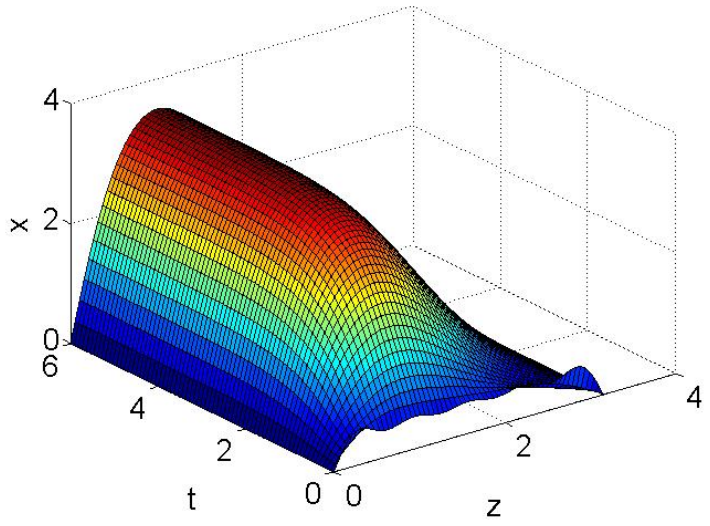


Figure 3.1: open-loop temperature profile of the system. The system converges to a nonuniform steady state in open-loop ssytem.

3.3.2 Linearization

To design the controller and observer pair, we use Galerkin method to approximate this infinite dimensional system by finite number of modes.

$$x = \phi C \tag{3.32}$$

$$\sum_{i=1}^{n} \dot{c}_i \phi_i = \sum_{i=1}^{n} c_i \frac{\mathrm{d}^2 \phi_i}{\mathrm{d}z^2} + \beta_T (\mathrm{e}^{-\gamma/(1+(\sum_{i=1}^{n} c_i \phi_i))} - \mathrm{e}^{-\gamma}) + \beta_u \Big(bu - \sum_{i=1}^{n} c_i \phi_i \Big) \tag{3.33}$$

In standard POD and passthrou POD method, ϕ denotes vector of standard POD basis functions and passthrou POD basis functions, respectively.

Multiply both sides by $\phi_j (j = 1, 2, \cdots, n)$ and integrate over Ω_z, which is $(0, \pi)$ in this case.

$$\begin{aligned} \sum_{i=1}^{n} \int_0^\pi \dot{c}_i \phi_i \phi_j \mathrm{d}z = & \sum_{i=1}^{n} \int_0^\pi c_i \frac{\mathrm{d}^2 \phi_i}{\mathrm{d}z^2} \phi_j \mathrm{d}z + \int_0^\pi \beta_T (\mathrm{e}^{\frac{-\gamma}{1+(\sum_{i=1}^{n} c_i \phi_i)}} - \mathrm{e}^{-\gamma}) \phi_j \mathrm{d}z \\ & + \Big(\int_0^\pi \beta_u b \phi_j \mathrm{d}z \Big) u - \beta_u c_j \\ & j = 1, 2, \cdots, n \end{aligned} \tag{3.34}$$

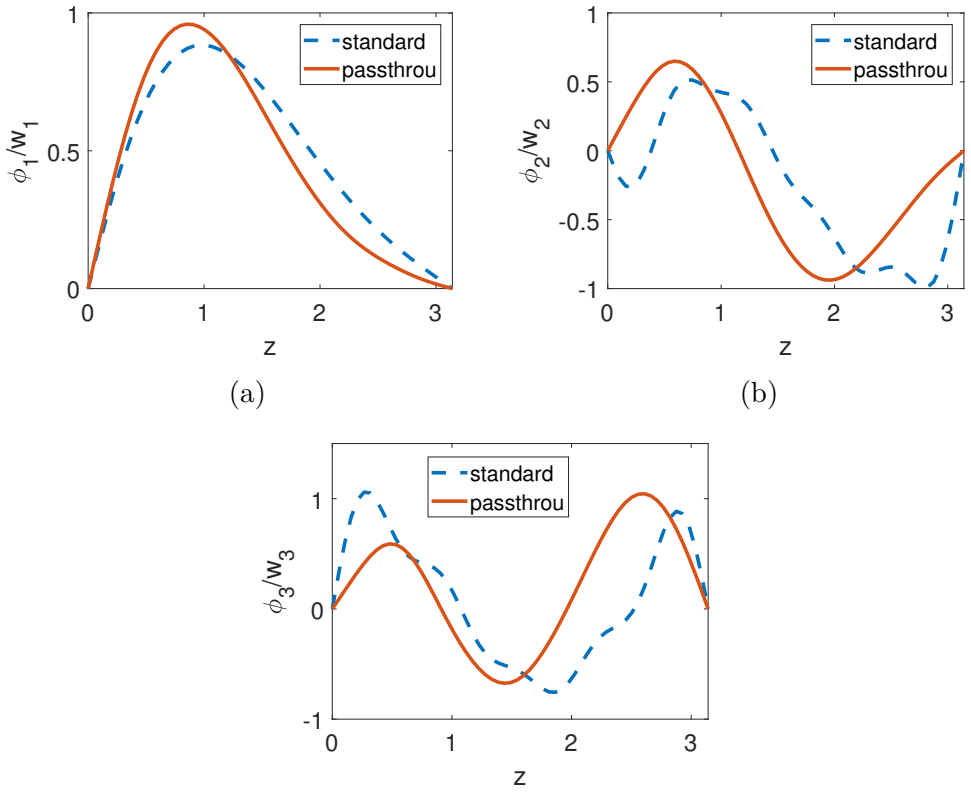


Figure 3.2: Comparison of spatial profiles of (a) 1st; (b) 2nd; (c)3rd basis function of standard POD method and passthrou POD method.

The ODEs of (3.34) can be expressed in matrix form:

$$\psi \dot{C} = F(C, u) \tag{3.35}$$

where

$$\psi_{ij} = \int_0^\pi \phi_i \phi_j \mathrm{d}z \tag{3.36}$$

and $F(C, u)$ represents the right hand side of Eq.(3.34)

To design the observer and controller gains, the system is linearized at $x = 0$; in the above ODEs, the corresponding point is $c_j = 0 (j = 1, 2, \cdots, n)$

$$\begin{aligned} \sum_{i=1}^{n} \int_0^\pi \dot{c}_i \phi_i \phi_j \mathrm{d}z = & \sum_{i=1}^{n} \int_0^\pi c_i \frac{\mathrm{d}^2 \phi_i}{\mathrm{d}z^2} \phi_j \mathrm{d}z + \sum_{i=1}^{n} \int_0^\pi \beta_T e^{-\gamma} \gamma \phi_i \phi_j c^i \mathrm{d}z - \beta_u c_j \\ & + \Big(\int_0^\pi \beta_u b \phi_j \mathrm{d}z \Big) u \end{aligned} \tag{3.37}$$

This set of equations can be compactly expressed as

$$\psi \dot{C} = \mathsf{A_s} C + \mathsf{B} u \tag{3.38}$$

$$\dot{C} = \psi^{-1} \mathsf{A_s} C + \psi^{-1} \mathsf{B} u \tag{3.39}$$

The gain matrices of controller and observer are determined based on the linearized system.

In standard POD method, the basis functions are orthonormal, so ψ is identity matrix.

In this case, the passthrou POD basis functions are also orthonormal.

3.3.3 Observer and Controller

To reduce the number of sensors required to estimate the states of the system, a dynamic observer is constructed using Luenberger's method [42]. To take full advantage of the nonlinear term in the governing equation, the observer gain matrix is constructed using the linearized system of (3.39) and the nonlinear system of (3.35) is used to constitute observer.

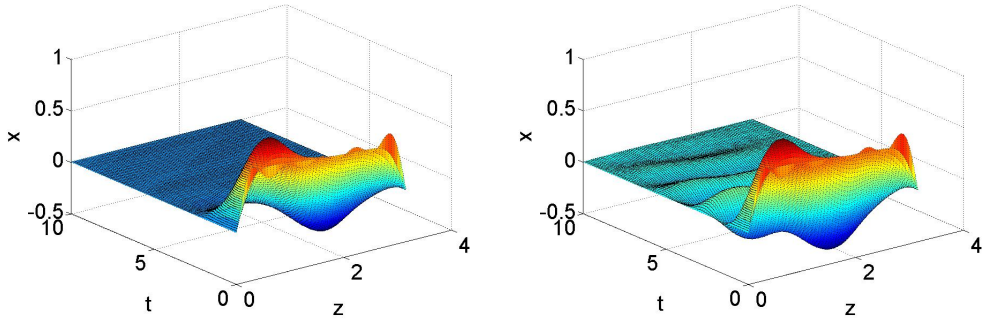
$$\begin{aligned} \dot{\tilde{c}}_j = & \sum_{i=1}^{n} \int_0^{\pi} \tilde{c}_i \frac{\mathrm{d}^2 \phi_i}{\mathrm{d}z^2} \phi_j \mathrm{d}z + \int_0^{\pi} \beta_T (\mathrm{e}^{-\gamma/(1+(\sum_{i=1}^{n} \tilde{c}_i \phi_i))} - \mathrm{e}^{-\gamma}) \phi_j \mathrm{d}z \\ & + \int_0^{\pi} \beta_u b u \phi_j \mathrm{d}z - \beta_u \tilde{c}_j + K_g (y - W_{lm} \tilde{c}) \end{aligned} \tag{3.40}$$

where K_g is the gain matrix designed based on the linearized system using LQR theory. In both systems based on passthrou POD and standard POD, the weighting matrices Q and R are identity matrix I and $2I$, respectively. The controller gain matrix was designed using the same method and same weighting matrices.

$$u = -G_0 \tilde{c} \tag{3.41}$$

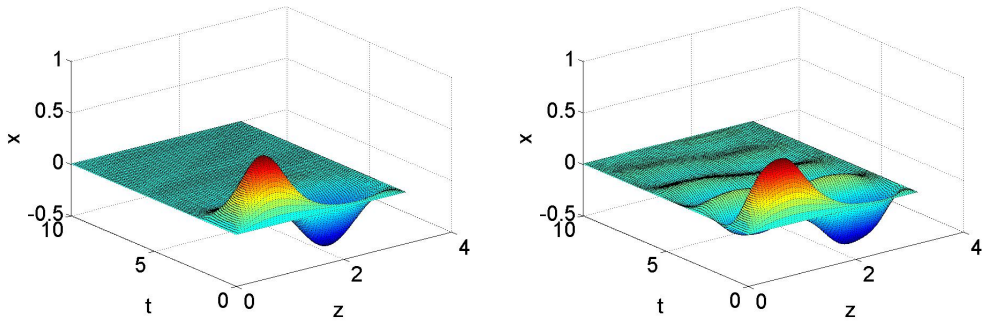
The closed loop simulation results are as follows:

In Fig. 3.3a and 3.3b, two sets of controller and observer are constructed based on standard POD method and passthrou POD method respectively. Using standard POD method, the system has more oscillation. The temperature estimated by observer based on passthrou and standard POD are depicted in Fig. 3.4a and 3.4b.



(a) (b)

Figure 3.3: Closed-loop temperature profile of the system with controller and observer based on (a) passthrou POD method; (b) standard POD method. System with controller and observer based on standard POD oscillates



(a) (b)

Figure 3.4: temperature estimated by observer based on (a) passthrou POD method; (b) standard POD method. The initial guess of temperature by the observer is $x(z) = 0$.

The initial condition is assumed to be unknown by observer.

3.3.4 Impact of Number of Snapshots and Modes Excitation

Since standard POD does not have a criterion on how many snapshots are needed to construct basis functions or whether the system is properly excited, low quality snapshots may lead to inaccurate ROM. To show passthrou POD circumvents these two barriers and passthrou POD basis functions capture dominant modes better than standard POD, in this section, we compared the basis functions of standard POD and passthrou POD when the number of snapshots increases to 50 and 70.

Meanwhile, snapshots are also obtained from open loop system with 2 different initial conditions to construct basis functions. Finally, controllers and observers are redesigned for new ROMs.

From Fig.3.5a to Fig.3.5f, basis functions of standard POD and passthrou POD based on different snapshot ensembles are compared. In the cases denoted by '30','50', and '70', snapshots are generated by the same open loop profile, with the number of snapshots varying from 30 to 70; while in '2ic', 60 snapshots are used to construct basis functions, with 30 snapshots from one initial condition, and the other 30 snapshots from another initial condition. It can be seen that all the basis functions of passthrou POD are almost the same, while different snapshots ensembles have much larger impact on standard POD method. In other words, although passthrou POD uses standard POD as its first step, under-sampling and low quality snapshots are less likely to affect passthrou POD basis functions compared with standard POD.

To illustrate the impact of snapshots quality and quantity on the system, we compared the temporal profile of error of observer, norm of temperature and control action. The error of observer is defined as follows:

$$e(t) = \sqrt{\frac{1}{L_r}\int_0^{\pi} \Big(x(z,t) - \tilde{x}(z,t)\Big)^2} \tag{3.42}$$

where L_r denotes the length of rod, which is π in this case; x and $\tilde{x}$ represent temperature profile and temperature predicted by observer, respectively. Norm is defined as

$$n(t) = \sqrt{\frac{1}{L_r}\int_0^{\pi} x^2(z,t)} \tag{3.43}$$

Fig 3.6 to 3.8 display the error of observer, norm of temperature and control action of systems based on passthrou POD and standard POD with different snapshots ensembles. Since the passthrou basis functions corresponding to different snapshots ensemble are almost the same, all systems have nearly the same response (figures not shown here due to the limit of space). Hence, only one system based on passthrou is compared with systems based on standard POD. It can be seen in Fig. 3.6 that, observer based on standard POD has larger error and the error decreases when number of snapshots increases. Collecting snapshots from system with different initial conditions can also reduce the error. Fig. 3.7 shows temperature of

system using passthrou POD reaches set point faster than standard POD. This is due to the oscillation in control action (Fig. 3.8).

Based on above results, we draw the conclusion that enriching snapshots by increasing number of snapshots or collecting snapshots with different initial conditions can both improve the quality of basis functions of standard POD in capturing the dominant modes and has little influence on passthrou POD.

3.3.5 Impact of Disturbance

To evaluate how robust these two methods are, a parametric disturbance is applied to the system. We assume there is disturbance that increases the activity of of catalyst starting from $t = 0$ with the coefficient in $\beta_T(z)$ increasing from 16 to 17.

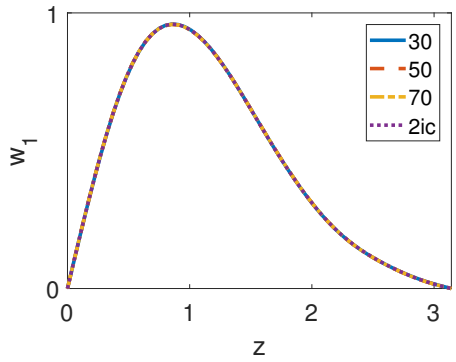
In Fig. 3.9 to 3.11, we compared the error of observer, norm of temperature and control action of systems based on passthrou POD and standard POD. Passthrou POD and standard POD without disturbance are denoted by 'p' and 's' in the figures. We observe that the disturbance dramatically increases the error of observer based on standard POD and has less impact on the observer based on passthrou POD. Besides, the control action when using passthrou to design the controller and observer exhibits less oscillations. Fig. 3.10 shows that temperature reaches set point much slower in presence of disturbance when using standard POD than passthrou POD.

3.3.6 Open-loop Approximation Error Analysis

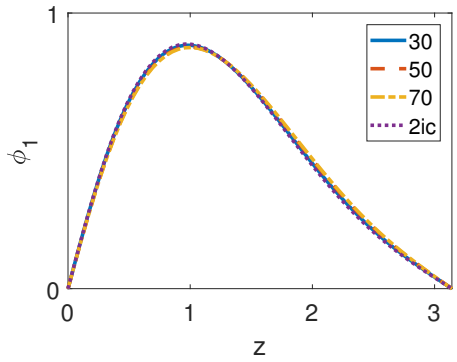
In this section we compare the approximation error of passthrou POD and standard POD in open-loop process to exclude the influence of observer and controller despite the fact that they are designed using same methods in previous sections. We excite the system with 2 series of boxcar functions as input signals. The time interval ΔT and amplitude obey uniform distribution $U(0, 1)$ and $U(-0.5, 0.5)$ respectively. The input signals are displayed in Fig. 3.12 and 3.13. Error of ROM is defined as

$$e(t) = \|x(z, t) - \tilde{x}(z, t)\|_\infty \tag{3.44}$$

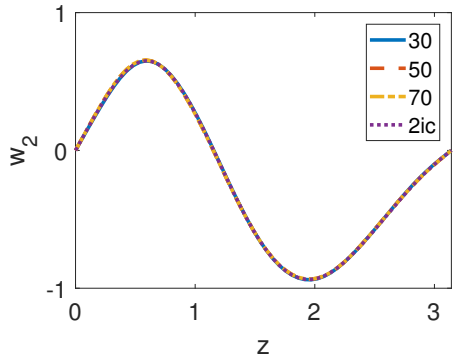
where $\|\cdot\|_{infty}$ denotes L infinity norm. Errors of ROM with these 2 signals are shown in Fig. 3.14 and Fig. 3.15.



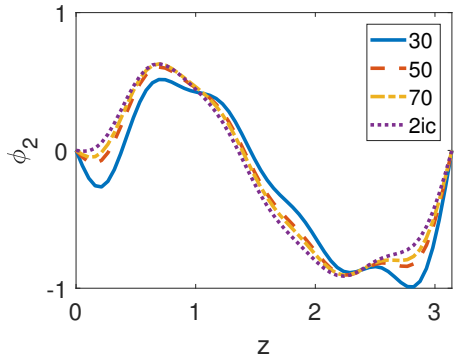
(a) First basis function of passthrou POD



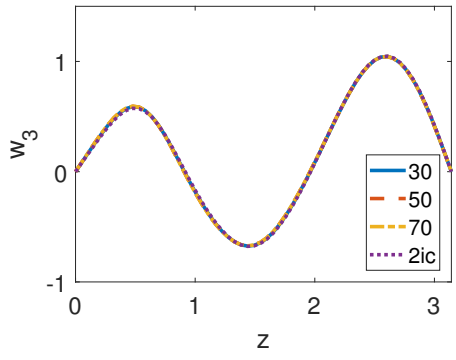
(b) First basis function of standard POD



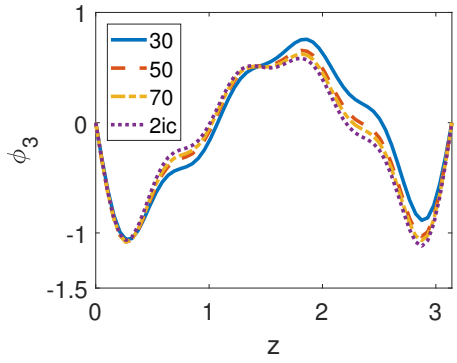
(c) Second basis function of passthrou POD



(d) Second basis function of standard POD



(e) Third basis function of passthrou POD



(f) Third basis function of standard POD

Figure 3.5: Comparison of basis function generated by passthrou POD and standard POD using different snapshot ensembles. The basis functions of standard POD are affected by the snapshot ensembles; the basis functions of passthrou POD are almost the same when different snapshot ensembles are used.

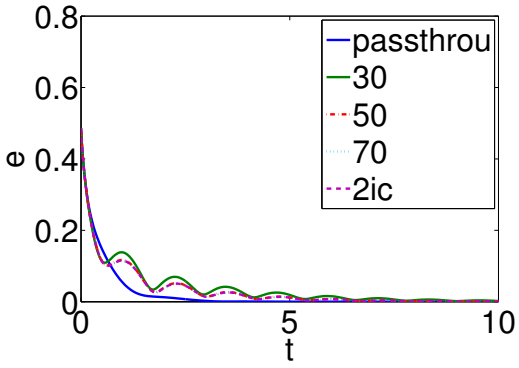


Figure 3.6: error of observer

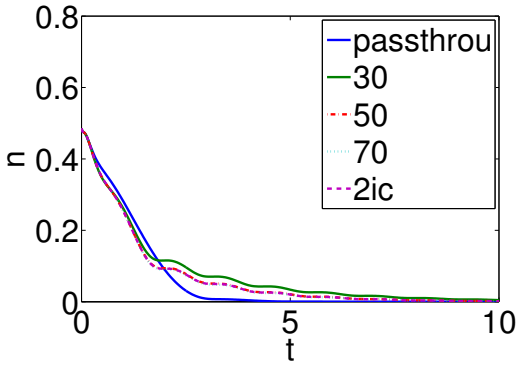


Figure 3.7: norm of temperature

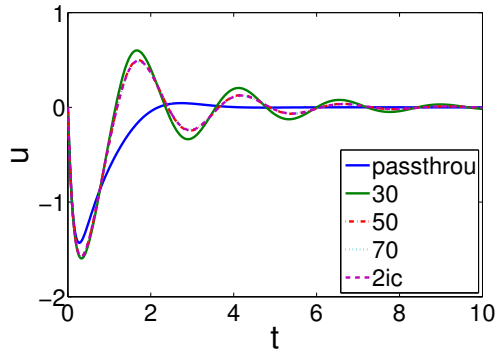


Figure 3.8: control action

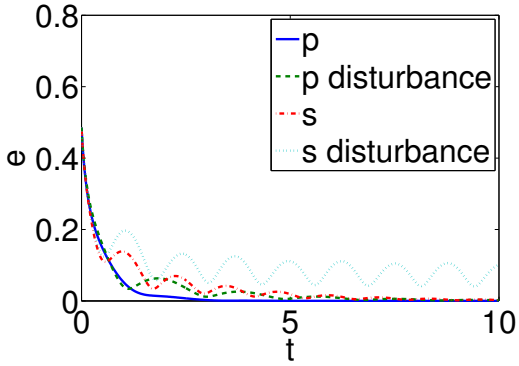


Figure 3.9: error of observer in the presence of disturbance

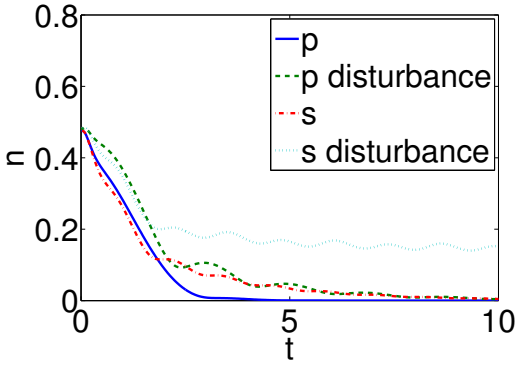


Figure 3.10: norm of temperature in the presence of disturbance

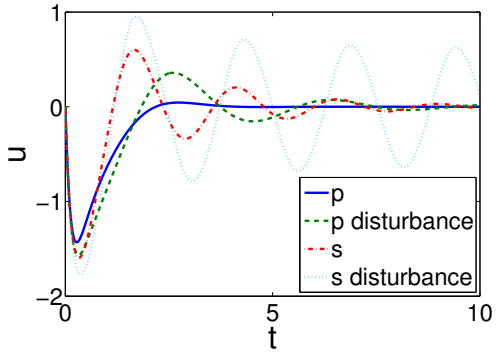


Figure 3.11: control action in the presence of disturbance

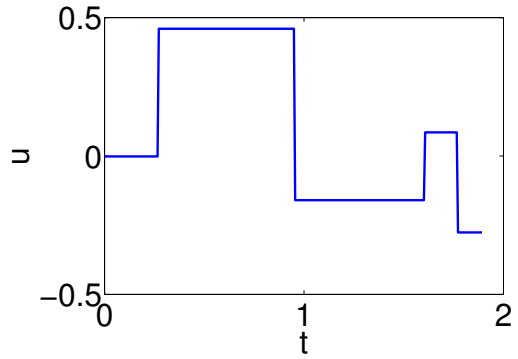


Figure 3.12: First input signal

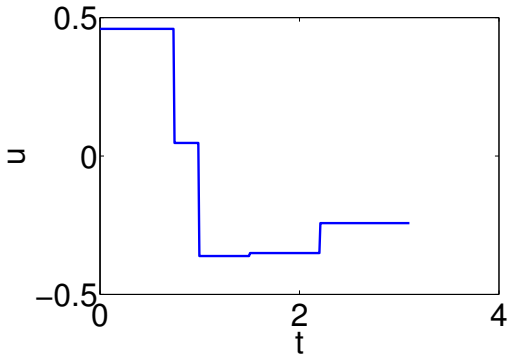


Figure 3.13: Second input signal

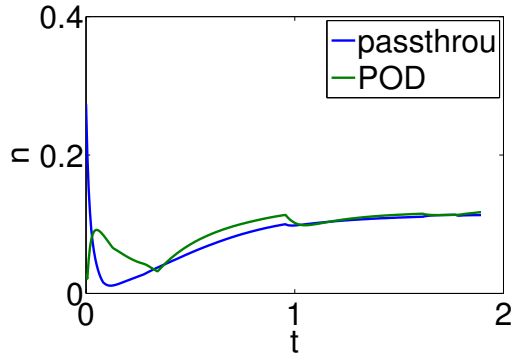


Figure 3.14: Error of ROM with first input signal

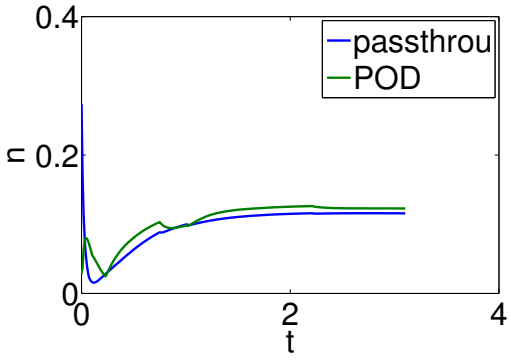


Figure 3.15: Error of ROM with second input signal

Chapter 4
Dissipative Distributed Parameter Systems On-line Reduction and Control Using DEIM/APOD Combination

One of the disadvantages of POD is the quality of the basis functions depends on the quantity of the snapshots and how the snapshots are collected [27].

One way to address these issues is to update the basis functions when new information about the system is available. This kind of methods is called incremental POD. Compared with standard POD, incremental POD can mitigate the impact of initial snapshots ensemble. Since applying standard POD iteratively poses computational difficulties, significant research effort about incremental POD has been devoted to alleviate the computational cost, including [40, 44, 45].

One of incremental POD methods is adaptive proper orthogonal decomposition (APOD). Compared with other incremental POD methods, APOD has an advantage that not only are new snapshots incorporated into the snapshot ensemble, but also previous snapshots are eliminated. Keeping the ensemble size uniform can reduce the computational burden resulting from increasing ensemble size. Small ensemble size can also help in better capturing the system behavior change by using limited new observations. When the observation is incomplete, gappy APOD can be employed [46].

One promising approach to mitigate this situation is adaptive proper orthogonal decomposition (APOD), which updates the basis function when the controller

and observer are implemented on-line. The basis functions are updated using new snapshot collected from the process as it evolves. Since applying POD iteratively is computational intensive, APOD algorithm updates snapshot ensemble by eliminating least important snapshots [1] and determines new basis function size by estimating eigenvalues of new covariance matrix. Compared with the newest snapshot approach in original APOD [31], the modified APOD that retains the most important snapshots has the advantage of capturing important profiles over a wider state-space region.

Another issue with ROMs is that the resulting ordinary differential equations (ODEs) system may still be computationally expensive to evaluate since nonlinearities in the original partial differential equation(s) have to be numerically integrated in space in the resulting ROM. This can lead to delays in the computation of control action. To circumvent this issue, discrete empirical interpolation method (DEIM) [47] is adopted to reduce computational cost. In this method, the nonlinear term in the governing equation in the whole domain is estimated by measurement at k points with k much less than the number of spatial grid points using nonlinear basis functions. Nonlinear basis functions are constructed off-line and the positions of those k points are determined based on DEIM algorithm.

In this chapter, we propose a combination of APOD and DEIM to design nonlinear controllers of reduced computational requirements to force the closed-loop DPS evolution to a desired operating point. To illustrate the performance of the proposed control scheme and investigate the stability of it, an open-loop unstable diffusion reaction process and a fluid flow system described by the Kuramoto-Sivashinsky equation are investigated.

4.1 Preliminaries

4.1.1 Notation

The L^2 inner product of function g_1 and g_2 is denoted by

$$(g_1, g_2) = \int_{\Omega_z} g_1 g_2 \mathrm{d}z,$$

where $z \in \Omega_z$ is the spatial coordinate. Arithmetic mean is denoted by $\langle \cdot \rangle$, and $\|p\| = \sqrt{(p,p)}$.

4.2 Discrete Empirical Interpolation Method

Discrete empirical interpolation method was proposed by Sorensen in 2010 [47]. The method seeks to reduce the computational cost associated with evaluating nonlinear terms in POD with Galerkin projection. Since the nonlinear term needs to be evaluated at each time instant and at each grid point, solving the model might be computational intensive. DEIM circumvents this issue by approximating the nonlinear term profile using the values of nonlinear term at k grid points with k being much smaller than the number of spatial grid points M and a set of projection basis. The projection basis for nonlinear term is called nonlinear basis function in this manuscript; it is predetermined using POD algorithm with snapshots of the nonlinear term. The location of those grid points are determined based on the nonlinear basis functions.

In the semi-linear PDE (Eq. 1.1), evaluating the nonlinear term $g(x)$ at each spatial grid point is computationally expensive. DEIM approximates $g(x)$ by a superposition of "basis functions"

$$g \approx Ud \tag{4.1}$$

where $U = [\tilde{\phi}_1 \ \tilde{\phi}_2 \ \dots \ \tilde{\phi}_k] \in \mathbb{R}^{M \times k}$ represents the nonlinear basis functions. Nonlinear basis functions are generated by applying POD to a set of nonlinear snapshots $\{g_i\}$, and $d \in \mathbb{R}^{k \times 1}$ corresponds to the mode (c) in MWR. Recalling that $k \ll M$, this problem is overdetermined for d. To determine d, k rows from this problem are selected to solve d. To accomplish this, the equation is multiplied by a matrix P^T at both sides:

$$P^T g = (P^T U)d \tag{4.2}$$

where $P \in \mathbb{R}^{M \times k}$. Each column of the P matrix is taken from certain column of an identity matrix. The locations of those columns in the identity matrix is called interpolation indices, which is determined based on the nonlinear basis functions U [47].

As a result, the nonlinear term g is approximated by:

$$g \approx U(P^T U)^{-1} P^T g \tag{4.3}$$

Note that $P^T g$ only requires the value of g at k points to be generated.

In summary, DEIM contains the following steps:

- generate nonlinear snapshots set using g and snapshots set $\{a_i\}$
- apply POD to nonlinear snapshots to get nonlinear basis function U
- determine interpolation indices by using U

4.3 Observer & Controller design using DEIM/APOD combination

In this section, we employ DEIM in APOD-based reduced order model to reduce the computational cost in controller and observer. Many controller design methods, such as feedback linearization and Lyapunov based control, and dynamic observer design methods require the evaluation of the nonlinear term. By using DEIM, evaluating nonlinear term at each grid point can be circumvented and the integration step can also be predetermined.

We consider a Luenberger-type dynamic observer based on the reduced order model (Eq. 2.17) in which the basis functions are determined by APOD

$$\frac{\mathrm{d}\tilde{c}}{\mathrm{d}t} = L_o\tilde{c} + G(\tilde{c}) + B(\tilde{c})u + K_g(y - \tilde{y}) \tag{4.4}$$

where $\tilde{c}$ refers to the estimated state. The gain matrix K_g is determined using LQR theory [42].

Since the linear part $L_o\tilde{c}$ can be evaluated with lower cost than the nonlinear

part $G(\tilde{c})$, we apply DEIM (Eq. 4.3) to $G(\tilde{c})$,

$$\begin{aligned} G(\tilde{c}) = & \int_{\Omega_z} \varphi_j \tilde{f}(\sum_{q=1}^{n} c_q \phi_q) \mathrm{d}z \\ \approx & \Phi^T \tilde{\mathfrak{f}} \Delta z \\ \approx & \underbrace{\Phi^T U (P^T U)^{-1} \Delta z}_{D} P^T \tilde{\mathfrak{f}} \end{aligned} \tag{4.5}$$

where Φ denotes discretized basis functions, $\mathfrak{f}$ is discretized f, $\tilde{\mathfrak{f}}$ *and* $\tilde{f}$ indicate that the value of $\mathfrak{f}$ and f are estimated. We assume the grid points are evenly spaced and Δz denotes the grid interval.

Another complication and source of computational cost relates to the computation of the effect the manipulated variables have on the systsem dynamics. If $b(x, z) \equiv b(z)$, B in Eq. 4.4 is computed off-line employing POD. If $b(x, z)$ is a linear function of x, $B(\tilde{c})$ in Eq. 4.4 can be written as $\mathscr{B}\tilde{c}$, we still apply standard POD when evaluating $B(\tilde{c})$. If $b(x, z)$ is a nonlinear function of x, we generate another set of nonlinear basis functions to reduce the cost of evaluating $B(\tilde{c})$ like Eq. 4.5

$$\begin{aligned} B(\tilde{c}) = & \int_{\Omega_z} \varphi_j \tilde{b}(\sum_{q=1}^{n} c_q \phi_q) \mathrm{d}z \\ \approx & \Phi^T \tilde{b} \Delta z \\ \approx & \underbrace{\Phi^T U_b (P_b^T U_b)^{-1} \Delta z}_{D_b} P_b^T \tilde{b} \end{aligned} \tag{4.6}$$

We obtain,

$$\frac{\mathrm{d}\tilde{c}}{\mathrm{d}t} = L_o \tilde{c} + D P^T \tilde{f}(t) + D_b P_b^T \tilde{b} u + K_g (y - \tilde{y}) \tag{4.7}$$

where

$$D = \Phi^T U (P^T U)^{-1} \Delta z$$

$$D_b = \Phi^T U_b (P_b^T U_b)^{-1} \Delta z$$

can be predetermined and updated when the basis functions are updated.

Similarly, we can use this approach in controller design methods, which will be

explained in detail in the examples.

4.4 Application

4.4.1 Diffusion-reaction process

To better explain the approach, we consider an illustrative diffusion reaction process [31]. In this process, an elementary exothermic reaction takes place on the surface of a catalytic rod. Without control, the system evolves to a spatially varying steady state. The objective is to regulate the temperature. After normalizing the energy conservation equation, the governing equation

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_T(z)(\mathrm{e}^{-\gamma/(1+x)} - \mathrm{e}^{-\gamma}) + \beta_u(b(z)u - x) \tag{4.8}$$

is defined in $[0, \pi]$, where x represents normalized temperature. The values of parameters are $\beta_u = 2, \gamma = 2$.

$$\beta_T(z) = 16[\cos(z) + 1] \tag{4.9}$$

indicates a spatial distribution on the activity of the catalyst. The initial condition is $x(z, 0) = 0.5$ and the objective is $x = 0$. There is one sensor that measures the state at 0.33π and one actuator (cooler) that affects the interval $[0.3\pi, 0.7\pi]$ uniformly.

$$b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \tag{4.10}$$

where $H(\cdot)$ denotes Heaviside step function.

We use APOD to construct a ROM. Then we use feedback linearization to regulate the system. In the ROM, the control objective $x = 0$ becomes $c = [c_1, c_2, \cdots, c_n]^T = [0, 0, \cdots, 0]^T$. Controller is designed to regulate the unstable state variable c_1

$$u = \frac{-\left(L_1 c + D_1 P^T \tilde{f}\right) + k_d \tilde{c_1}}{B_1} \tag{4.11}$$

where L_1 denotes the first row of L, $D_1 = \Phi_1^T U (P^T U)^{-1} \Delta z$ and B_1 is the first row of B; k_d is a negative scalar.

The estimated state is obtained from a dynamic observer Eq. 4.4.

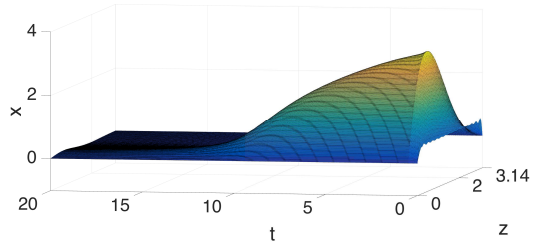


Figure 4.1: closed-loop temperature profile of the system using APOD.

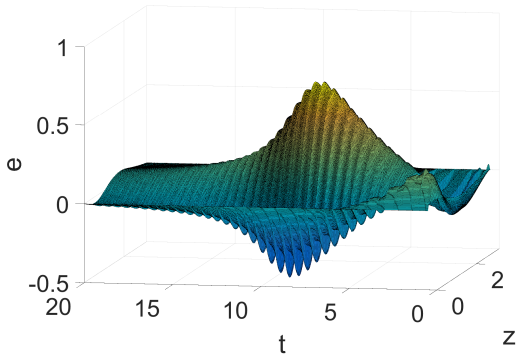


Figure 4.2: error of the observer in the system using APOD.

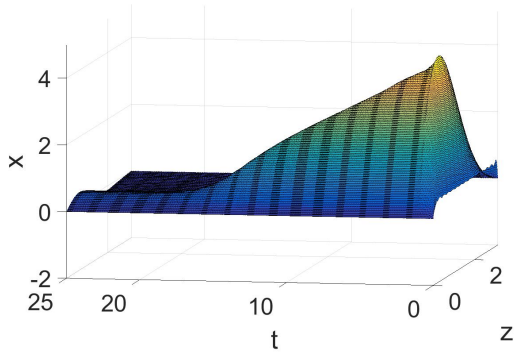


Figure 4.3: closed-loop temperature profile of the system using APOD/DEIM combination.

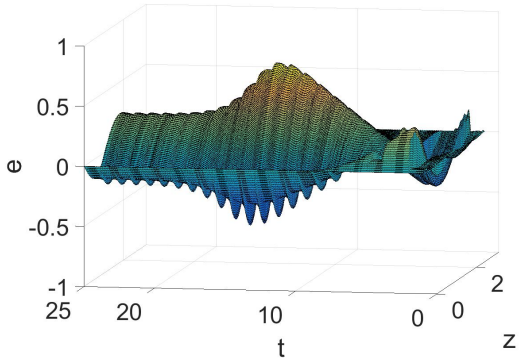


Figure 4.4: error the observer in the system using APOD/DEIM combination.

The required initial snapshot ensemble for APOD is collected during the first 1s. During this time, $u = 0$. Then controller and observer start regulating the system. The state in the dynamic observer and the basis functions are updated every 1s. Fig. 7.3 shows the temperature profile of the system using APOD, and the error of the observer is depicted in Fig. 7.4. The corresponding result for the system using APOD/DEIM combination ($k = 3$) is given in Fig. 4.3 and 4.4. It can be seen that the controller can regulate the system using both APOD and APOD/DEIM combination and the estimation of the observer is accurate. The spatial temporal profile of the nonlinear term and the errror of the approximation by DEIM are given in Fig. 4.5-4.6. We observe that DEIM can accurately approximate the profile of the nonlinear term. Note that the amplitude of the nonlinear term is around 15

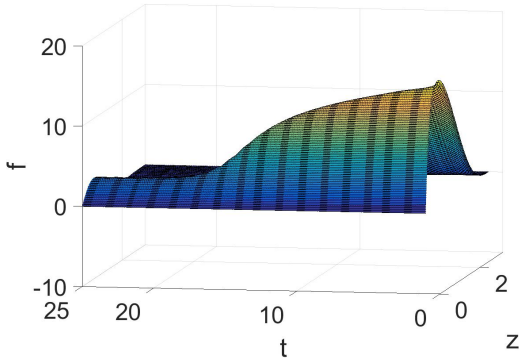


Figure 4.5: spatial temporal spatial profile of the nonlinear term.

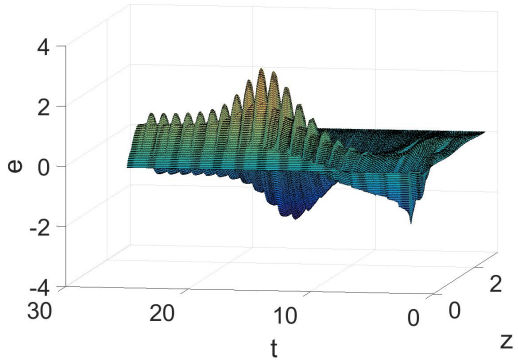


Figure 4.6: error of the nonlinear term approximated by DEIM.

(Fig. 4.5), so the relative error of the approximation of DEIM is small.

To analyze the effect of k on the result, the error of the observer, the error of the approximation of the nonlinear term, and the norm of the state in systems using different number of nonlinear basis functions k are compared in Fig. 4.7-4.9, where the subscript refers to the value of k. The CPU time spent in controller for the whole period of time is given in table 4.1. DEIM reduces the computational cost in controller by about 50% and has small influence on the performance of the controller. As k increases, the result is closer to the system using modified APOD; when k increases to 10, the "error" introduced by DEIM is negligible since both the error of the observer and the norm of the state in systems based on APOD and APOD/DEIM combination are almost identical when $k = 10$. The dimension of APOD (basis size) is given in Fig. 4.10.

Table 4.1: comparison of computation cost

k	APOD	APOD/DEIM combination
3		0.430s
5	0.801s	0.401s
10		0.419s

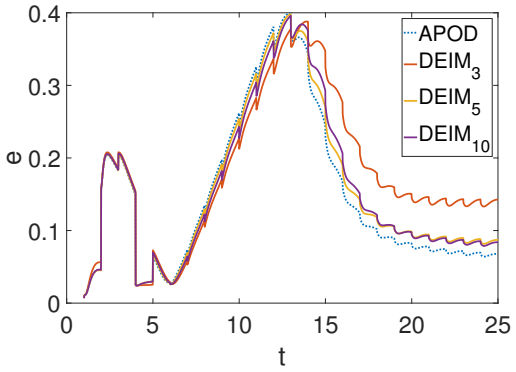


Figure 4.7: comparison of the error of the observer.

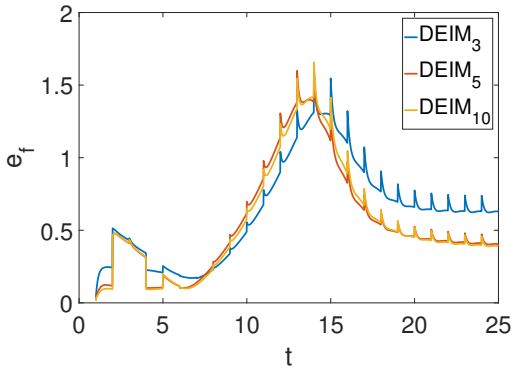


Figure 4.8: comparison of the error of the approximation of the nonlinear term by DEIM.

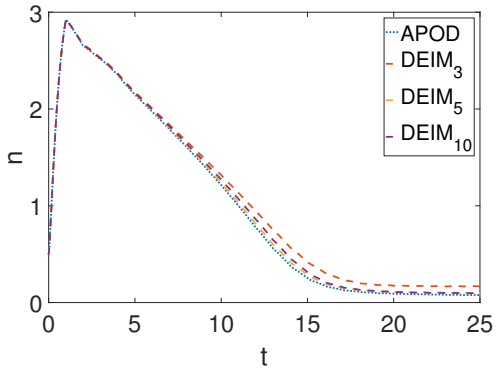


Figure 4.9: comparison of the norm of the state.

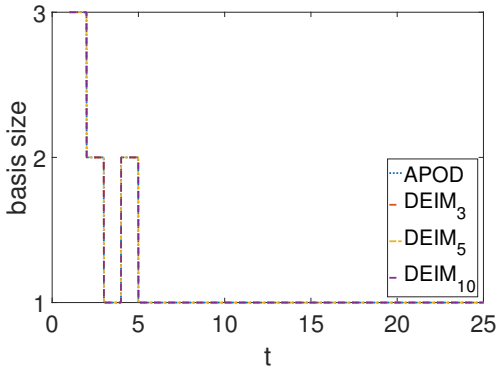


Figure 4.10: comparison of the basis size.

4.4.2 Kuramoto-Sivashinsky equation

To present the capability and robustness of the method, we apply modified APOD and APOD/DEIM combination to stabilize Kuramoto-Sivashinsky equation (KSE) [48]. KSE can be used to describe falling liquid film thickness [49] and flame front propagation [50]. KSE is given by

$$\frac{\partial x}{\partial t} = -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x\frac{\partial x}{\partial z} + \sum_{i=1}^{l} b_i u_i(t), \tag{4.12}$$

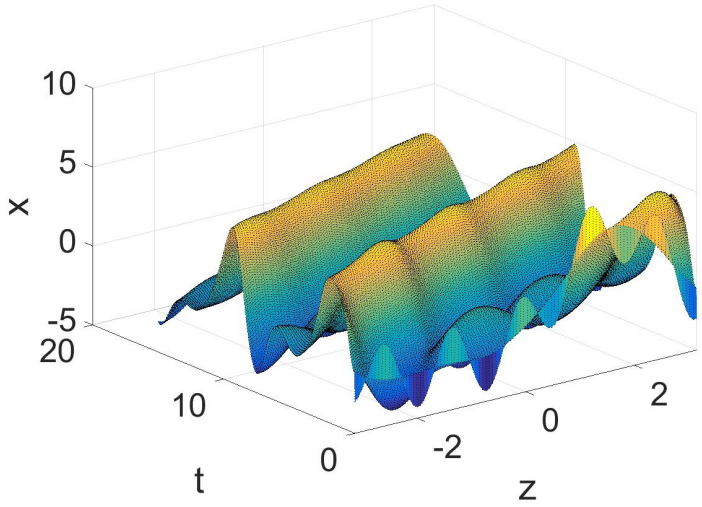


Figure 4.11: open loop profile of the state

with periodic boundary condition

$$\frac{\partial^j x}{\partial z^j}(-\pi, t) = \frac{\partial^j x}{\partial z^j}(\pi, t) \quad j = 0, \cdots, 3 \tag{4.13}$$

and the initial condition

$$x(z, 0) = 3\sin(z) - \cos(2z) - \sin(5z) + 2\cos(5z) \tag{4.14}$$

where $z \in [-\pi, \pi]$, $b_i = \delta(z - z_i)$, and $\nu = 0.25$. There are 7 actuators at locations

$$z_i = -\frac{5\pi}{6}, -\frac{\pi}{2}, -\frac{\pi}{4}, -\frac{\pi}{6}, \frac{\pi}{3}, \frac{2\pi}{3}, \frac{7\pi}{8} \qquad i = 1,\ 2,\ \ldots,\ 7 \tag{4.15}$$

20 sensors are uniformly distributed in $[-\pi, \pi]$ and measure the state continuously.

The objective is to stabilize the system at the origin. The open loop profile of the system is displayed in Fig. 5.9. Similar to the diffusion reaction process, the initial snapshots are collected during the first 1 unit of time and basis function are updated every 1 unit of time. Static observer is applied to estimated the state

$$\tilde{c} = (\Phi_m^T \Phi_m)^{-1} \Phi_m^T y \tag{4.16}$$

where Φ_m is obtained from the basis functions Φ so that $y = \Phi_m c$. We use Lyapunov based controller [51] to regulate the system.

$$u = -k(c)(L_B \mathcal{V})^T(c) \tag{4.17}$$

where $\mathcal{V} = c^T c$, L refers to Lie derivative.

$$k(c) = \frac{L^*_{F_0}\mathcal{V}(c) + \sqrt{\left(L^*_{F_0}\mathcal{V}(c)\right)^2 + \|(L_B\mathcal{V})^T(c)\|^4}}{\|(L_B\mathcal{V})^T(c)\|^2} \tag{4.18}$$

and

$$L^*_{F_0}\mathcal{V}(c) = L_{F_0}\mathcal{V}(c) + \rho_0\mathcal{V}, \tag{4.19}$$

ρ_0 is a positive number. By applying DEIM, we can obtain

$$F_0 = Lc + F(c) \approx Lc + DP^T f \tag{4.20}$$

The system response and the estimated state using modified APOD and APOD/DEIM combination ($k = 3$) are depicted in Fig. 4.12-4.15. The results using these two methods are almost identical. The comparison of results using different k is given in Fig. 4.16-4.17. As k increases, the difference between these two methods becomes smaller. In Fig. 4.16, the error of the observer during $t = [12, 13]\&[14, 15]$ when $k = 3$ is smaller than other cases is because the basis size is larger. It can been seen that when $k = 3$, the error of the observer increases faster after updating the basis functions using new snapshots. This is expected since the larger the k, the smaller the error introduced by DEIM. The computational cost spent in controller during the whole period of time is displayed in table 4.2, using DEIM decreases the computational cost by about 20%.

Table 4.2: comparison of computation cost

k	APOD	APOD/DEIM combination
3		196s
5	251s	201s
10		205s

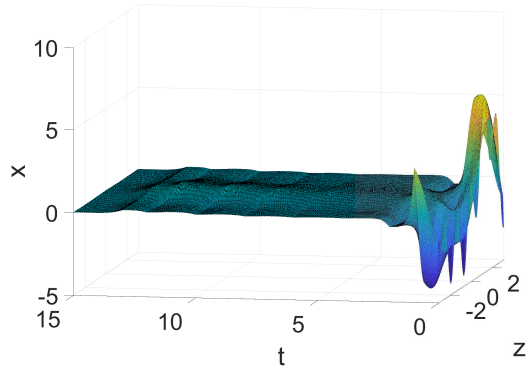


Figure 4.12: closed-loop response of the system using APOD.

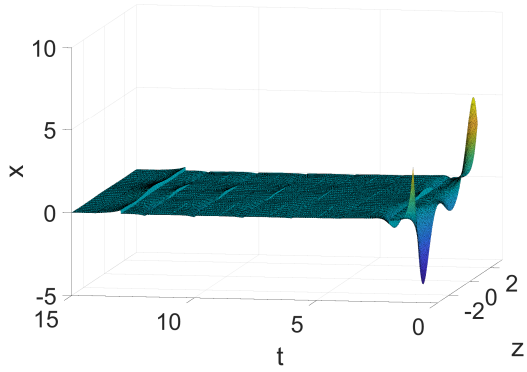


Figure 4.13: estimated state by the observer in the system using APOD.

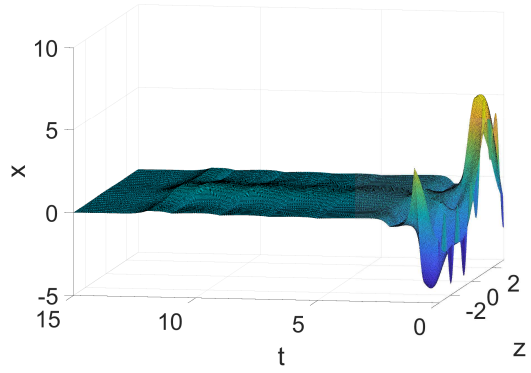


Figure 4.14: closed-loop response of the system using APOD/DEIM combination.

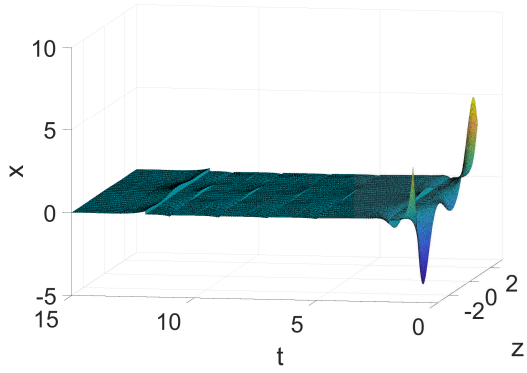


Figure 4.15: estimated state by the observer in the system using APOD/DEIM combination.

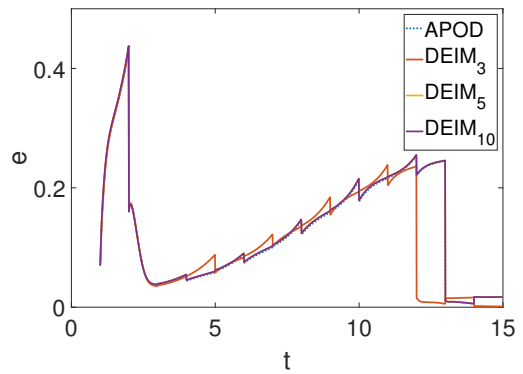


Figure 4.16: comparision of the error of the observer.

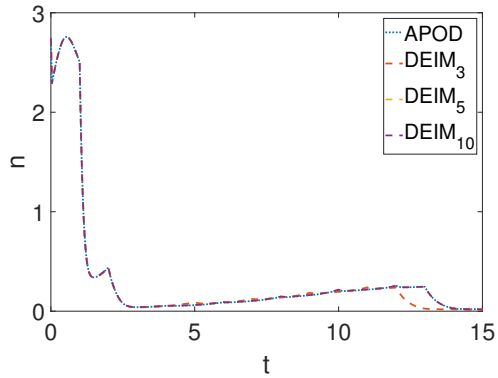


Figure 4.17: comparision of the norm of the state.

Chapter 5
Revisiting APOD accuracy for nonlinear control of transport reaction processes: a spatially discrete approach

There are two versions of APOD: original APOD [31] and modified APOD [1]. In original APOD, the least recent snapshot is eliminated; while in modified APOD, the least important snapshot is dropped. Although the modified version has the advantage of capturing important profiles, it suffers from newest snapshots being eliminated when new snapshots show significant difference from previous ones. Another concern is the mode size may oscillates in systems with less aggressive controllers.

In this chapter, we propose another modified version of APOD to circumvent the above limitations and also provide better performance in terms of stability and accuracy. In the rest of this chapter, we call it discrete adaptive proper orthogonal decomposition (DAPOD). Since the spatial component of the system response over a period of time is usually more stable than the temporal component, we propose to use the spatial component instead of the temporal component in basis function update and eigenvalue estimation. This approach therefore relaxes the assumption that the system evolves slowly. Compared with modified APOD, it circumvents the problem that the newest snapshot may keep being eliminated. It also has the advantage that it can provide a more accurate estimate of energy captured by each

basis function and the basis functions it generates are closer to those generated by "iterative" POD. Besides, the updated basis functions are strictly orthogonal, which will reduce the error when Galerkin projection is employed to build ROMs.

5.1 Problem Formulation

In this manuscript, we consider the output feedback regulation problem of processes described by the following semi-linear partial differential equation

$$\frac{\partial x}{\partial t} = \mathcal{L}x + g(x) + b(z)u \tag{5.1}$$

Assumption 1 *$g(0) = 0$ and $g(x)$ is a Lipschitz nonlinear vector function.*

5.2 Discrete Adaptive Proper Orthogonal Decomposition and Controller/Observer Design

5.2.1 Discrete Adaptive Proper Orthogonal Decomposition

Compared with original APOD, modified APOD has the advantage that old important snapshots will not be eliminated. However, when the process visits a state space region far away from that of the snapshots ensemble, new snapshots may keep being eliminated because the contribution of the new snapshot is small in the updated basis functions according to the importance evaluation algorithm. Also, in both original APOD and modified APOD, the inaccuracy of the estimation of the eigenvalues might lead to basis size oscillation or more basis functions than necessary. The third disadvantage is even though economy singular value decomposition can increase the accuracy of eigenvectors of covariance matrix, the unmatched covariance matrix and eigenvectors will result in a set of basis functions that are not orthogonal. To address the above issues, we propose a method called discrete adaptive proper orthogonal decomposition (DAPOD).

We develop this method in the context of finite difference or finite volume discretized systems. Without loss of generality, we assume $x \in \mathbb{R}$. Finite difference

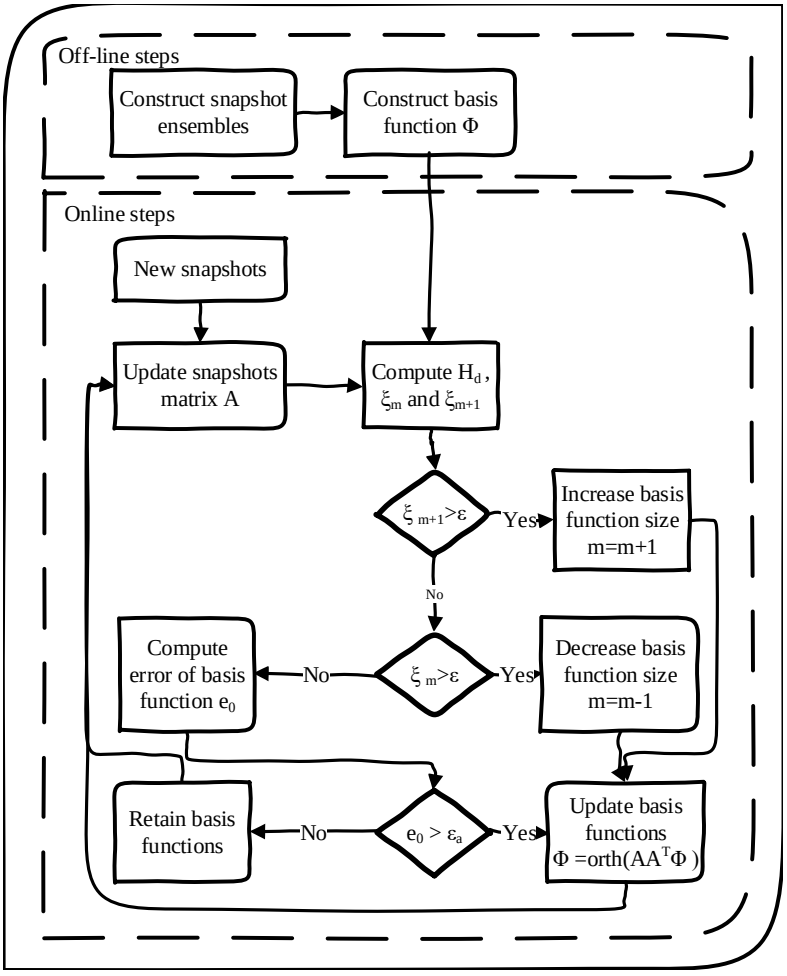


Figure 5.1: Flow chart of discrete adaptive proper orthogonal decomposition

discretization of PDE results in the following ODEs system:

$$\frac{\mathrm{d}\hat{x}}{\mathrm{d}t} = \hat{\mathcal{L}}\hat{x} + \hat{g}(\hat{x}) + \hat{b}u \tag{5.2}$$

where $\hat{\mathcal{L}}$ corresponds to $\mathcal{L}$ and $\hat{g}$ corresponds to g in Eq. 1.1. $\hat{x} \in \mathbb{R}^{M\times 1}$ and M is the dimension of the finite difference discretized system. Basis functions generated by DAPOD are denoted by $\hat{\Phi} = [\hat{\phi}_1\ \hat{\phi}_1\ \ldots\ \hat{\phi}_m] \in \mathbb{R}^{n\times m}$. Note that usually $m \ll n$. To simplify notation, Φ and ϕ will still be used.

5.2.1.1 off-line basis function construction

In the off-line initial basis function construction step, we can use singular value decomposition (SVD) to construct the basis function Φ:

$$A = \Phi_0 \Sigma Z^T \tag{5.3}$$

where $A \in \mathbb{R}^{n \times M}$ is obtained from discretization of snapshots $[a_1(z)\ a_2(z)\ \ldots\ a_M(z)]$ and $\Phi_0 = [\phi_1\ \phi_2,\ \ldots\ \phi_n] \in \mathbb{R}^{n \times n}$. To make sure the new trend can be captured by updated basis functions, M is usually less than n. The diagonal entries of Σ are denoted by $\sigma_1, \sigma_2, \cdots, \sigma_M$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$. The basis functions Φ are obtained as

$$\Phi = [\phi_1\ \phi_2\ \ldots\ \phi_m],\ m \ll M \tag{5.4}$$

where m is determined to satisfy

$$\frac{\sum_{i=1}^{m} \sigma_i^2}{\sum_{i=1}^{M} \sigma_i^2} > 1 - \varepsilon \tag{5.5}$$

5.2.1.2 snapshots update

In the on-line step, the snapshot matrix A is updated when a new snapshot becomes available. Since the modified APOD may keep eliminating the newest snapshot, we propose an approach that considers both importance and "freshness" of snapshots. First we consider mode vectors of snapshots

$$D = \Phi^T A \tag{5.6}$$

Mode vector D_j is normalized as follows:

$$\tilde{D}_j = D_j / d_j \tag{5.7}$$

where $\tilde{D}_j$ and D_j denote the jth row of $\tilde{D}$ and D, respectively and d_j indicates the 2-norm of D_j. An importance vector $\mathfrak{h}$ is defined as :

$$\mathfrak{h}_j = (P_A)_{jj} \tag{5.8}$$

where

$$P_A = \tilde{D}^T \tilde{D} \tag{5.9}$$

and $()_{jj}$ denotes the element in the j^{th} row and the j^{th} column. The definition of the importance vector is analogous to z-scores in principal component analysis. The difference is z-scores algorithm has an extra step: the data mean is subtracted at the beginning.

To keep the newest snapshots from being eliminated, we construct the "freshness" vector $\mathfrak{g}$ based on the importance vector $\mathfrak{h}$ at each iteration. $\mathfrak{g}$ is an arithmetic sequence with the first and the last elements equal to the smallest and the largest elements in $\mathfrak{h}$, respectively. The total number of elements in the sequence is equal to the number of snapshots in the snapshot ensemble. The largest number is assigned to the most recent snapshot. The value of $\mathfrak{g}$ represent the "freshness" of snapshots. We define a vector $\mathfrak{q}$ as:

$$\mathfrak{q}_j = \mathfrak{h}_j + K\mathfrak{g}_j \tag{5.10}$$

where the coefficient K indicates the importance of keeping recent snapshots. The snapshot corresponding to the lowest value in $\mathfrak{q}$ will be eliminated. By adding $\mathfrak{h}_j$ and $\mathfrak{g}_j$, we keep the snapshots that are both important and represent the current behavior of the system. When $K \to \infty$, this step reverts to the newest snapshots approach; when $K = 0$, this step reverts to a variant of the most important snapshots approach. Note that the definition of the importance vector is different from that in modified APOD [1]. When K is chosen to be greater than 1, newest snapshot is guaranteed to be kept when next snapshot becomes available. It is recommended to choose K in $[0.5, 2]$ since $\mathfrak{h}_j$ has the same range as $\mathfrak{g}_j$.

5.2.1.3 update basis function size

As new snapshots become available, the number of basis functions needed to construct the ROM may increase or decease. We change the basis functions size accordingly based on the energy captured by the basis functions truncated in previous steps. To facilitate exposition, we will explain how to update the basis functions size when at most 1 basis function is added or deleted each time when a new snapshot becomes available. We determine whether the $(m+1)^{\text{th}}$ basis function will be added and the m^{th} basis function will be eliminated based on the

energy captured by each basis function

$$\xi_{m+1} = \frac{\mu_{m+1}}{\sum_{i=1}^{m+1} \mu_i} \tag{5.11}$$

$$\xi_m = \frac{\mu_m}{\sum_{i=1}^{m+1} \mu_i} \tag{5.12}$$

In the above definition, eigenvalues corresponding to from the $(m+2)^{\text{th}}$ to the last basis functions are truncated since they are negligible. If ξ_{m+1} increases to above ε, basis functions size increases by 1. Calculating these eigenvalues of the covariance matrix by solving the eigenvalue problem can be computationally expensive. To decrease the computational cost of this step and provide a more accurate way to estimate the eigenvalues, we take the following approach:

We project the space spanned by the snapshots into a subspace spanned by the first $m+1$ basis functions. By doing this, the first $m+1$ eigenvalues of the covariance matrix can be estimated by the eigenvalues of H_d

$$H_d = \Phi_{m+1}^T A A^T \Phi_{m+1} \tag{5.13}$$

where $\Phi_{m+1} = [\phi_1 \ \phi_2 \ \cdots \ \phi_{m+1}]$. Since at this step the snapshot matrix A has been updated using the new snapshot, while basis function has not been updated, solving the eigenvalue problem of H_d only returns an approximation of the eigenvalues of the new covariance matrix. Compared with the approach in both original APOD and modified APOD, in which the eigenspace of the covariance matrix instead of the space spanned by snapshot is partitioned to $\mathbb{P}$ and $\mathbb{Q}$, this approach can provide a more accurate approximation of the eigenvalues of the covariance matrix. The second advantage is the proposed approach is computationally cheaper than APOD because $H_d \in \mathbb{R}^{(m+1)\times(m+1)}$ and $m+1 \ll M$, while $H \in \mathbb{R}^{m\times m}$ and $C_q \in \mathbb{R}^{M\times M}$.

If ξ_{m+1} is smaller than ε, we check the energy captured by mth basis function ξ_m. The basis functions size will decrease by 1 when ξ_m decreases to below ε. When $\xi_{m+1} < \varepsilon$ and $\xi_m \geq \varepsilon$, the basis size is m.

If sampling is not frequent enough to guarantee new snapshots only show small difference from the existing snapshots ensemble, we can also increase or decrease at most $N(N > 1)$ basis functions each time snapshots ensemble is updated. In this situation, the eigenvalue problem of a $(m+N) \times (m+N)$ matrix is solved instead. As long as $m + N \ll M$, which is true in most cases, the computational cost of

this step will be much smaller than APOD.

5.2.1.4 check accuracy of projection

After calculating the energy captured by the basis functions, we check their accuracy. A residual e is defined as

$$e = \|\Phi\Phi^T a - a\| \tag{5.14}$$

where a denotes the newest snapshot and $\|\cdot\|$ indicates the 2-norm of a vector. We consider the relative residual as

$$\tilde{e} = \frac{\|\Phi\Phi^T a - a\|}{\|a\| + \epsilon_n} \tag{5.15}$$

where ϵ_n is a small constant. It is used to prevent $\tilde{e}$ from exploding when a approaches 0.

If the error increases to above ε_a, we reconstruct the basis functions. Compared with calculating the norm of the matrix $E = (I - ZZ^T)C_k(ZZ^T)$ in previous versions (original APOD and modified APOD), checking the error has more straightforward physical meaning and does not need to be adjusted as the size of E changes.

5.2.1.5 update basis function

The basis functions get updated when they are not accurate enough, or their number changes:

$$\Phi = \text{orth}(AA^T\Phi) \tag{5.16}$$

where "orth" refers to economy singular value decomposition operation. Compared with the previous algorithms, this modification has 2 advantages: first, it gives a better approximation of basis functions corresponding to updated snapshots ensemble. This point will be better explained and illustrated in the next section. The second advantage is it relaxes the limitation of APOD that new snapshots do not change abruptly. When it happens, basis functions generated by APOD lose orthogonality, while DAPOD guarantees the updated basis functions are orthonormal.

5.2.2 DAPOD-based model reduction

The reduced order model is constructed using Galerkin method.

$$\hat{x} = \sum_{i=0}^{n'} \phi_i c_i \tag{5.17}$$

where c_i are called modes, $n' \leq n$. The ODEs system (Eq. 5.2) is transformed to

$$\frac{\mathrm{d}c_s}{\mathrm{d}t} = A_{ss}c_s + A_{sf}c_f + g_s(c_s, c_f) + b_s u \tag{5.18}$$

$$\frac{\mathrm{d}c_f}{\mathrm{d}t} = A_{ff}c_f + A_{fs}c_s + g_f(c_s, c_f) + b_f u \tag{5.19}$$

in which $c_s = [c_1, c_2, \ldots, c_m]^T$, $c_f = [c_{m+1}, c_{m+2}, \ldots, c_{n'}]^T$, $A_{ss} \in \mathbb{R}^{m\times m}$, $A_{sf} \in \mathbb{R}^{m\times(n'-m)}$,$A_{ff} \in \mathbb{R}^{(n'-m)\times(n'-m)}$, $A_{fs} \in \mathbb{R}^{(n'-m)\times m}$, $b_s \in \mathbb{R}^{m\times n_1}$, $b_f \in \mathbb{R}^{(n'-m)\times n_1}$. Since the dynamics of Eq. 5.19 is fast and stable, c_f is negligible. We obtain

$$x \approx \Phi c_s(t) \tag{5.20}$$

$$\frac{\mathrm{d}c_s}{\mathrm{d}t} = A_{ss}c_s + g_s(c_s, 0) + b_s u \tag{5.21}$$

The resulting ROM is

$$\frac{\mathrm{d}c}{\mathrm{d}t} = \Phi^T \hat{\mathcal{L}} + \Phi^T \hat{g} + \Phi^T \hat{b} u \tag{5.22}$$

where the subscript s is omitted for simplicity. The ROM will be updated as basis functions Φ are updated when a new snapshot becomes available.

5.2.3 DAPOD-based observer design

In this paper, we use two methods to design the observer, dynamic observer and static observer.

5.2.3.1 dynamic observer

First, we express Eq. 6.3 in this form

$$\frac{\mathrm{d}c}{\mathrm{d}t} = L_o(c) + G(c) + Bu \tag{5.23}$$

The Luenberger-type dynamic observer is formulated as

$$\frac{\mathrm{d}\tilde{c}}{\mathrm{d}t} = L_o(\tilde{c}) + G(\tilde{c}) + Bu + G_m(y_m - \tilde{y}) \tag{5.24}$$

where $\tilde{c}$ refers to the estimated state. The gain matrix G_m is determined using LQR theory [42].

Assumption 2 *G is a Lipschitz nonlinear vector function.*

$$\|G(c_p) - G(c_q)\| \leq K_l \|c_p - c_q\|. \tag{5.25}$$

5.2.3.2 static observer

According to Eq. 5.17, the measured outputs are

$$y_m = \Phi_m c \tag{5.26}$$

where Φ_m is the matrix whose j^{th} row is the l_j^{th} row of basis functions Φ; l_j indicates the location of j^{th} output. We can obtain

$$\tilde{c} = (\Phi_m^T \Phi_m)^{-1} \Phi_m^T y_m \tag{5.27}$$

Note that we assume that continuous measurements are available and there are more than the number of modes that we employ for the controller design at all times. We also assume that there is no measurement noise.

5.2.4 DAPOD-based controller design

In this section, we design controllers to regulate the systems. Feedback linearization and Lyapunov based control are employed.

5.2.4.1 feedback linearization

Since the objective is to regulate the system at $x = 0$, the corresponding state in the ROM is $c = 0$. We assume the relative degree is 1. To reduce the number of actuators, for a system with m' unstable states, we only consider the state variables

in the unstable subsystem. The system can be written in this form

$$\frac{\mathrm{d}c'}{\mathrm{d}t} = L'_o(c', c_u) + G'(c', c_u) + B'u \tag{5.28}$$

$$\frac{\mathrm{d}c_u}{\mathrm{d}t} = L_{ou}(c', c_u) + G_u(c', c_u) + B_u u \tag{5.29}$$

The control action is

$$u = B'^{-1}\Big[- K_f \tilde{c}' - L'_o(\tilde{c}) + G'(\tilde{c})\Big] \tag{5.30}$$

where K_f is positive definite. If we ignore the error of the observer based on the separation principle between observation and control, we can obtain that

$$\frac{\mathrm{d}c'}{\mathrm{d}t} = -K_f \tilde{c}' \tag{5.31}$$

The stability of the full system is proven in Appendix A.

5.2.4.2 Lyapunov-based control

Another approach to design the controller is to use Lyapunov function. To consider the input constraint, for a Lyapunov function $\mathcal{V}$, we design the following feedback regulator [52, 53]. First, we use F to denote $L_o(c) + G(c)$ in Eq. 5.23

$$\dot{c} = F(c) + Bu \tag{5.32}$$

The manipulated input is computed as

$$u = -k(c)(L_B \mathcal{V})^T(c) \tag{5.33}$$

where

$$k(c) = \frac{L_F^* \mathcal{V}(c) + \sqrt{\Big(L_F^* \mathcal{V}(c)\Big)^2 + (u_{max}\|(L_B\mathcal{V})^T(c)\|)^4}}{\|(L_B\mathcal{V})^T(c)\|^2\Big[1 + \sqrt{1 + (u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}\Big]} \tag{5.34}$$

and

$$L_F^* \mathcal{V}(c) = L_F \mathcal{V}(c) + \rho_0 \mathcal{V} \tag{5.35}$$

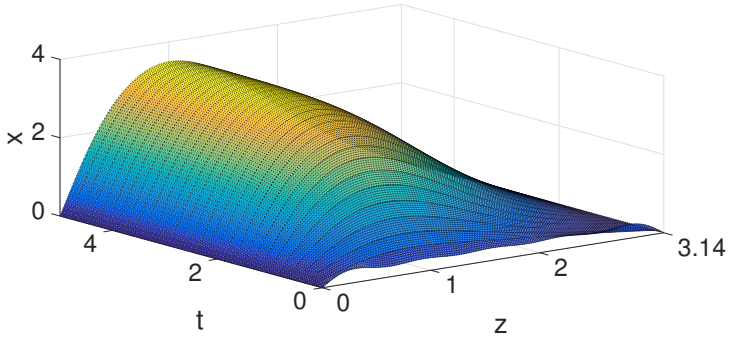


Figure 5.2: snapshots used to construct basis functions off-line.

ρ_0 is a positive number and L denotes the standard Lie derivative. When the initial condition is within the region $\{c \in C : L_F^* \mathcal{V}(c) \leq u_{max} \| (L_B \mathcal{V})^T(c) \|\}$, input constraints are satisfied and the time-derivative of Lyapunov function (Eq.5.59) is negative definite [52].

The proof of stability without involving separation principle is given in A.

5.3 Diffusion Reaction Process

In this section, we compare the performance of DAPOD and APOD by applying them to regulate a diffusion reaction process, in which an elementary exothermic reaction takes place along the surface of a catalytic rod [31, 54].

$$\rho C \frac{\partial T}{\partial \tilde{t}} = k_1 \frac{\partial^2 T}{\partial z^2} - A k_2(z) \mathrm{e}^{-\frac{E}{RT}} c_0 \Delta H + \frac{k_3 l_s}{\Delta A}(T_c - T) \tag{5.36}$$

where $k_2(z)$ indicates a spatial distribution on the activity of the catalyst, t is time and z denotes spatial coordinate. The concentration of reactant c_0 is assumed to be constant. The boundary condition is

$$\begin{aligned} z = 0 : T = T_0 \\ z = \pi : T = T_0 \end{aligned} \tag{5.37}$$

The output is the temperature at $z = 0.33\pi$

$$\tilde{y}_m = T(0.33\pi, t) \tag{5.38}$$

We define the following variables to simplify Eq. 7.15-5.38

$$\begin{aligned} x =& \frac{T - T_0}{T_0} \\ b(z)u =& \frac{T_c - T_1 - T_0}{T_0} \\ t =& \frac{\tilde{t} k_1}{\rho C} \\ y_m =& \frac{\tilde{y}_m - T_0}{T_0} \\ \beta_u =& \frac{A k_2 c_0 \Delta H}{T_0 \rho C} \\ \gamma =& \frac{E}{R T_0} \end{aligned} \tag{5.39}$$

where

$$T_1 = A k_2(z) \mathrm{e}^{-\frac{E}{RT_0}} c_0 \Delta H \cdot \frac{\Delta A}{k_3 l_s} \tag{5.40}$$

to obtain

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_T(z)(\mathrm{e}^{-\gamma/(1+x)} - \mathrm{e}^{-\gamma}) + \beta_u(b(z)u - x) \tag{5.41}$$

where

$$\beta_T(z) = 16[\cos(z) + 1] \tag{5.42}$$

β_u denotes the dimensionless heat transfer coefficient; γ is dimensionless activition energy. The values of parameters are $\beta_u = 2, \gamma = 2$.

Remark 1 *T_1 is defined so that $x = 0$ is a steady state when control action u is 0.*

The output and boundary conditions are

$$y_m = \int_0^\pi \delta(0.33\pi) x(z) \mathrm{d}z \tag{5.43}$$

$$x(0, t) = 0,\ x(\pi, t) = 0 \tag{5.44}$$

During open-loop process operation, the system evolves to a spatially varying steady state, which is displayed in Fig. 5.2, where the initial condition is

$$x(z, 0) = 0.3 \tag{5.45}$$

The uniform steady state $x = 0$ is unstable. Cooling is used to regulate the

temperature of the rod, which affects the interval $[0.3\pi, 0.7\pi]$ uniformly:

$$b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \tag{5.46}$$

where $H(\cdot)$ denotes Heaviside step function. The goal is to force the system state to a spatially uniform steady state $x(z,t) = 0$.

5.3.1 Updating Step in APOD & DAPOD

Since both APOD and DAPOD try to estimate the basis functions corresponding to the updated snapshots ensemble and reduce the computational cost, the basis functions generated by applying POD directly to the updated snapshot ensemble are considered as the best basis function set we can get. To illustrate that DAPOD gives a better set of basis functions, we compare the basis functions generated by APOD, POD, and DAPOD using the same snapshot ensemble. When using POD, first we collect some snapshots from the open loop process to form "old" snapshot ensemble, in which the initial condition is

$$x(z,0) = \begin{cases} -0.1 & 0 \leq z \leq \frac{\pi}{2} \\ 0.4 & \frac{\pi}{2} < z \leq \pi \end{cases} \tag{5.47}$$

The snapshots are given in Fig. 5.4. Then we replace one of the snapshots with a snapshot from the closed loop process to form a "new" snapshot ensemble. The basis functions corresponding to old and new snapshot ensembles and that estimated by APOD and DAPOD are compared in Fig. 5.3a and Fig. 5.3b. It can be seen that both APOD and DAPOD give very accurate estimates of the first basis function, while estimation of the second basis function by DAPOD is more accurate than APOD. Note that the discontinuity in the original basis functions results from the discontinuity in the initial condition in the open loop process which is used to generate these basis functions. After the "old" snapshot is replaced by a continuous snapshot, the previous behavior fades in the new basis functions.

5.3.2 Feedback Linearization & Dynamic Observer

In this example, we exploit feedback linearization to regulate the system. After a ROM is constructed, the objective of this problem becomes regulating the unstable

state variables in the ROM, which is the first state variable in this case.

First we employ Galerkin method to construct ROM (Eq.2.17). The unstable subsystem becomes

$$\frac{\mathrm{d}c_1}{\mathrm{d}t} = L_{o1}(c) + G_1(c) + B_1 u \tag{5.48}$$

In this example, L_o corresponds to $\frac{\partial^2 x}{\partial z^2} - \beta_u x$ and G corresponds to $g = \beta_T(z)(\mathrm{e}^{-\gamma/(1+x)} - \mathrm{e}^{-\gamma})$. The controller is computed as:

$$u = \frac{-\Big(L_{o1}(c) + G_1(c)\Big) + k_d c_1}{B_1} \tag{5.49}$$

where k_d is a negative scalar.

Since feedback linearization requires full information about the process and only the temperature at one point is available, a nonlinear dynamic observer is designed to estimate the states of the process.

$$\frac{\mathrm{d}\tilde{c}}{\mathrm{d}t} = L_o(\tilde{c}) + G(\tilde{c}) + Bu + G_m(y_m - \tilde{y}) \tag{5.50}$$

where $\tilde{y}$ represents the temperature at $z = 0.4$ estimated by the observer. $G_m \in \mathbb{R}^{m\times 1}$ refers to the observer gain matrix. The gain matrix G_m is determined by LQR theory [42]. In this algorithm Q and R are $2I$ and I respectively of proper dimensions, where I denotes the identity matrix.

5.3.3 Simulation Results

During the first 0.5s, the system evolves with $u = 0$. During this stage, 30 snapshots are collected. These snapshots are used to construct the initial basis functions for both DAPOD and APOD. After that, we assume new snapshots are available every 0.5 units of time during closed loop operation. The energy threshold ε is set to 98%. In this work, coefficient k in Eq. 5.49 is set to -0.1. The spatial-temporal profile of temperature in the system based on DAPOD and APOD are displayed in Fig. 5.5a and 5.5b, respectively. It can be seen that the offset in the system based on APOD is larger than that based on DAPOD.

To quantitatively evaluate the performance of APOD and DAPOD, we define 2 performance indices: the error of an observer and the norm of the state. The error of an observer represent the average error in space between the real state x and

the estimation of the observer $\tilde{x}$:

$$e_0 = \sqrt{\frac{\int_{\Omega_z}(x-\tilde{x})^2 \mathrm{d}z}{L}} \tag{5.51}$$

where L denotes the length of the rod. Since the set point is the origin, the second performance index represents the distance from the system state to the set point:

$$n_0 = \sqrt{\frac{\int_{\Omega_z} x^2 \mathrm{d}z}{L}} \tag{5.52}$$

The error of observers based on these 2 methods is compared in Fig. 5.6 and the norm of the state is given in Fig. 5.7. It can be seen that the observer in the system based on DAPOD has smaller error and the steady state has smaller offset.

The mode sizes in DAPOD and APOD are provided in Fig. 5.8. In APOD, the mode size oscillates before stabilizing at 1, while the mode size in DAPOD drops from 3 to 1 without oscillation. The reason for the difference is APOD overestimates the energy captured by the $(m+1)^{\text{th}}$ basis function and the redundant basis function is eliminated at next iteration through the condition number step. This issue only occurs when new snapshot shows large difference from the existing snapshot ensemble. Therefore, the mode size does not oscillate after the initial stage. This argument is supported by comparing the estimated eigenvalues of the new covariance matrix and the real eigenvalues (table 5.1). This oscillation phenomenon can also be seen in previous result [1]. The result looks better because an aggressive controller was employed.

The computational cost of APOD and DAPOD spent in updating basis functions are 0.08s & 0.03s.

Remark 2 *The performance of APOD depends on how aggressive the controller is. When the coefficient k in Eq. 5.49 is set to -1, the results of APOD and DAPOD are very similar.*

Remark 3 *When applying both APOD and DAPOD, the energy ratio ϵ is chosen to be "small" so that the order of the reduced order model can be low and the computational cost is reduced. It can be seen that the error of the observer is not trivial since only 1 basis function is used to model the system. While the ROM is not accurate, the controller still can regulate the system. This result demonstrates an*

advantage of APOD/DAPOD: the dominant behavior of the system is well captured, so that the system can be regulated without accurately modeling the system globally.

5.4 Kuramoto-Sivashinsky Equation

In this section, we apply modified APOD and DAPOD to stabilize Kuramoto-Sivashinsky equation (KSE) [48]. KSE is used extensively to describe falling liquid film thickness [49] and flame front propagation [50]. Here the integral form of KSE is considered.

$$\frac{\partial x}{\partial t} = -\nu\frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x\frac{\partial x}{\partial z} + \sum_{i=1}^{l} b_i u_i(t) \tag{5.53}$$

where

$$b_i = \delta(z - z_i) \tag{5.54}$$

with the boundary condition

$$\frac{\partial^j x}{\partial z^j}(-\pi, t) = \frac{\partial^j x}{\partial z^j}(\pi, t) \;\; j = 0, \cdots, 3 \tag{5.55}$$

and the initial condition

$$x(z, 0) = 3\sin(z) - \cos(2z) - \sin(5z) + 2\cos(5z) \tag{5.56}$$

7 actuators are used at locations

$$z_i = -\frac{5\pi}{6}, -\frac{\pi}{2}, \; -\frac{\pi}{4}, \; -\frac{\pi}{6}, \; \frac{\pi}{3}, \; \frac{2\pi}{3}, \; \frac{7\pi}{8} \qquad i = 1, \; 2, \; \ldots, \; 7 \tag{5.57}$$

sensors are uniformly distributed in $[-\pi, \pi]$ and measure the state continuously.

The objective is to stabilize the system at the origin. When $\nu < 1$, the system at the origin is locally unstable, and has chaotic behavior for very small ν [55]. To evaluate the performance of the proposed method under demanding conditions, ν is set to be 0.09. Note that for this value the onset of new trends is very fast. The open loop profile of the system is displayed in Fig. 5.9.

Table 5.1: Comparision of real eigenvalues and estimated eigenvalues

		DAPOD				APOD			
t=1	real eigenvalue	192.7442	0.5175	0.0604	0.0308	187.2918	0.635	0.0591	0.0308
	estimated eigenvale	192.7439	0.5172	0.0581	0.0306	186.118	0.6327	0.0323	1.2193
t=2	real eigenvalue	203.4395	0.4848	0.0903		195.3896	0.6351	0.0861	
	estimated eigenvale	203.4388	0.4848	0.0395		193.3236	0.635	0.635	
t=3	real eigenvalue	214.5709	0.3136			206.5103	0.4647		
	estimated eigenvale	214.5702	0.3042			199.4499	7.0786		

5.4.1 Lyapunov-based Control & Static Observer

In this example, Lyapunov based control is applied (section 6.2.4). We assume there is control constraint: $u_i \in [-10, 10]$. Since the objective corresponds to

$$c = 0 \tag{5.58}$$

the following Lyapunov function is considered:

$$\mathcal{V} = c^T c \tag{5.59}$$

Since this system has chaotic behavior, we use a static observer to estimate the state of the system.

5.4.2 Simulation Results

The complete profile of the system or new snapshot is assumed to be available every 1 units of time. During the first 1 unites of time, $u_i = 0$, 30 snapshots are collected to construct the initial basis functions. Since this system has chaotic behavior, a higher energy threshold is taken $\epsilon = 0.999$ to improve the accuracy of the estimation of the observer. We assume the snapshot is available every 1 unit of time. We use 30 equally distributed point sensors to measure the state.

The profiles of the systems based on DAPOD and modified APOD are displayed in Fig. 5.10a and 5.10b. Fig. 5.11 gives the corresponding norm of the profile. It can be seen that both systems reach the set point with negligible offset. But the system based on DAPOD arrives at the set point earlier than APOD.

The basis sizes in DAPOD and APOD are given in Fig. 5.12. It can also be observed that the basis size in APOD oscillates. After checking the real eigenvalues of the updated covariance matrices and the estimated ones, we find that the oscillation results from the overestimation of the energy captured by the $(m+1)^{\text{th}}$ basis function.

Fig. 5.13a and 5.13b compare the first basis function in APOD and DAPOD. While DAPOD stops updating the basis function after $t = 11$ because the error of the observer is very small, APOD has to keep updating the basis functions until $t = 20$ since the basis size oscillates.

The errors of the static observers are depicted in Fig. 5.14. Since the errors are similar in both observers, we decrease the number of sensors to 20 to evaluate the performance of the proposed method. The result is given in Fig. 5.15a to 5.18. Fig. 5.15a and 5.15b compare the system response. The distance from the set point (or norm of the profile) is displayed in Fig. 5.16. Fig. 5.17 shows the error of the observers. And the basis sizes are depicted in Fig. 5.18. It can been observed that after deceasing the number of sensors, the system based on DAPOD can still be regulated (Fig. 5.15a and 5.16) and has similar observer error (Fig. 5.17) and basis size (Fig. 5.18), while the closed-loop system based on APOD oscillates.

The computational cost of APOD and DAPOD spent in updating basis functions are 0.10s & 0.08s.

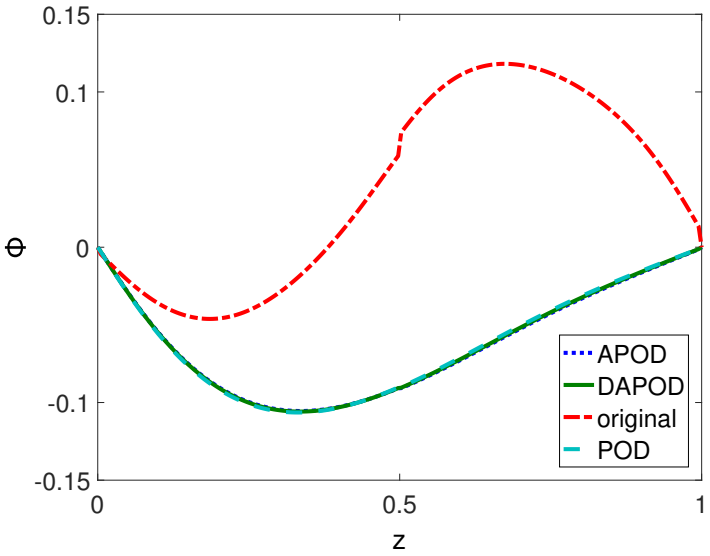


Figure 5.3a: First basis function of old and new snapshots matrices and those estimated by APOD and DAPOD.

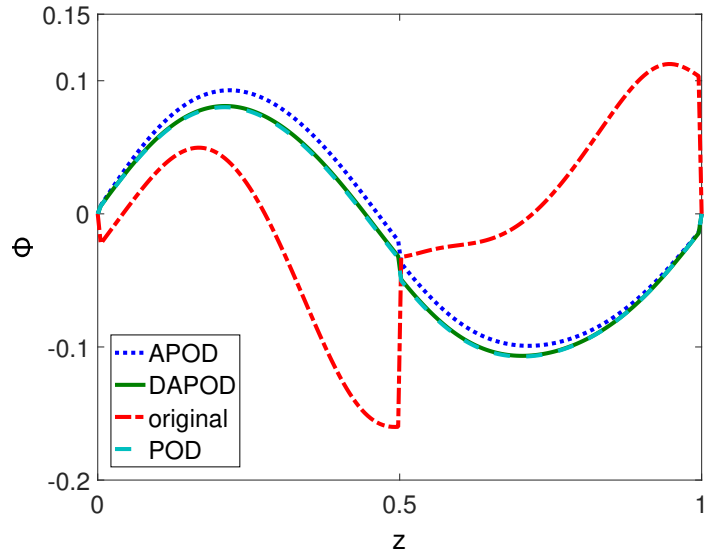


Figure 5.3b: Second basis function of old and new snapshots matrices and those estimated by APOD and DAPOD.

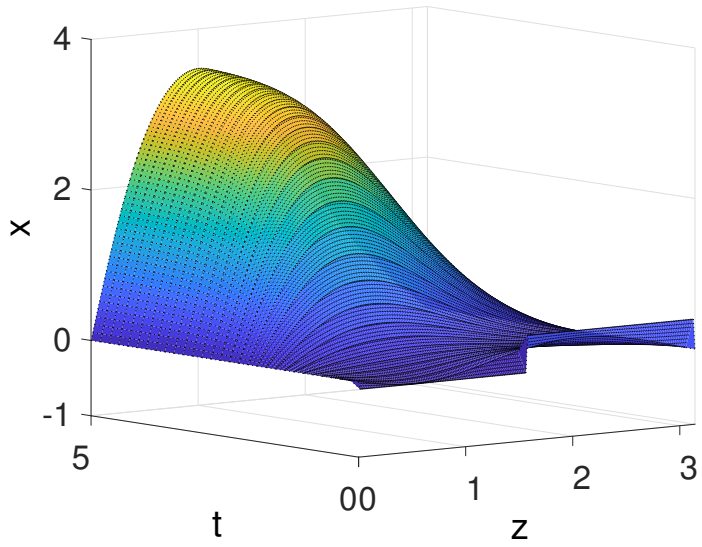


Figure 5.4: snapshots used to construct basis functions off-line.

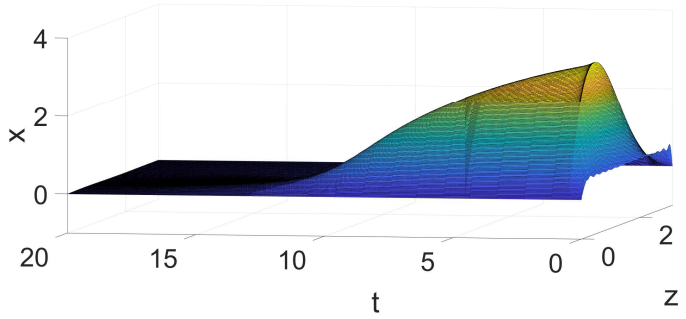


Figure 5.5a: closed-loop temperature profile of the system based on DA-POD.

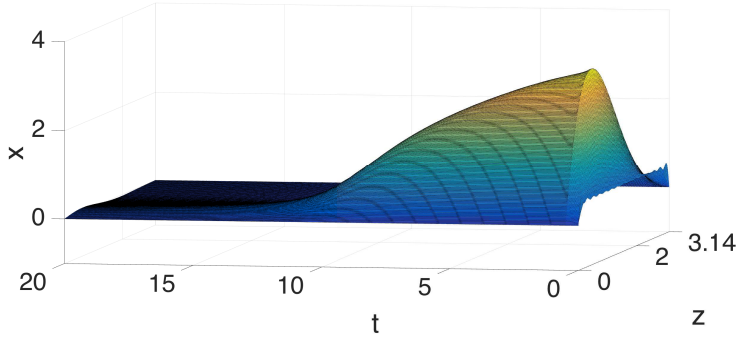


Figure 5.5b: closed-loop temperature profile of the system based on APOD.

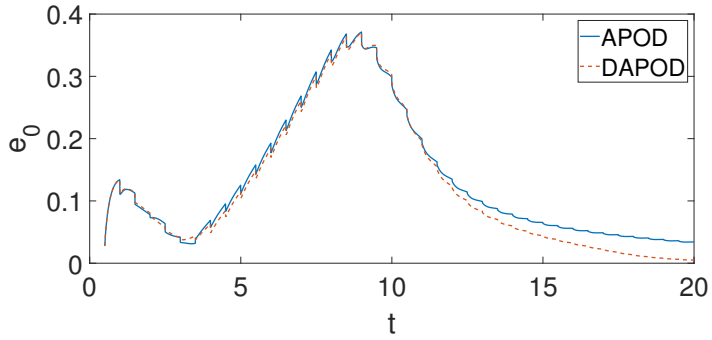


Figure 5.6: The error of observers in systems based on DAPOD and APOD.

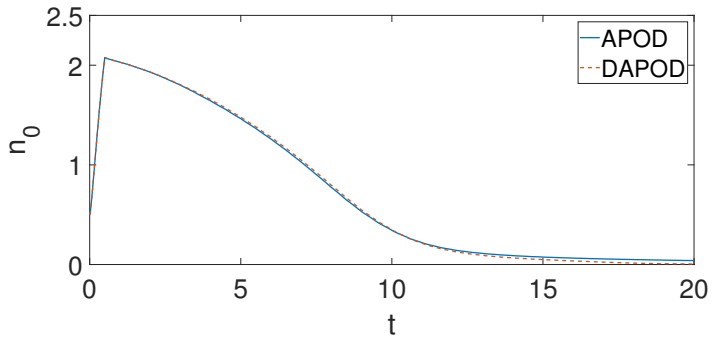


Figure 5.7: norm of temperature in systems based on DAPOD and APOD.

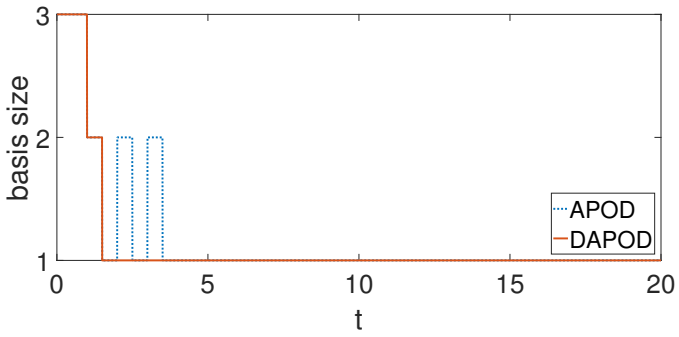


Figure 5.8: dimension of ROM in systems based on DAPOD and APOD.

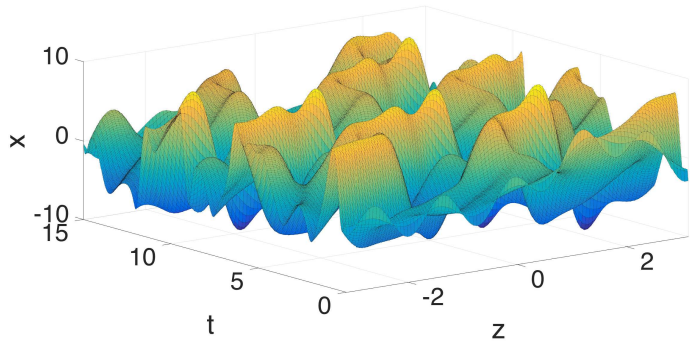


Figure 5.9: open loop profile of the state

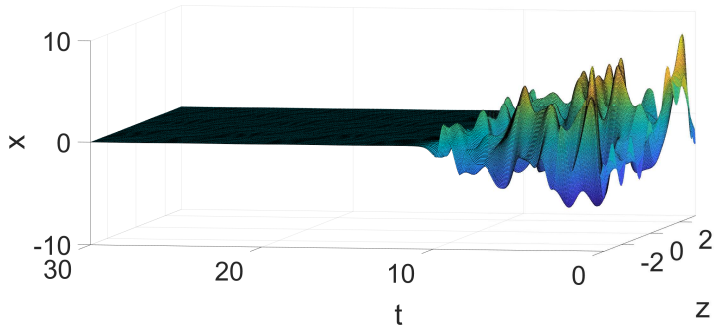


Figure 5.10a: closed-loop profile of the system based on DAPOD (30 sensors).

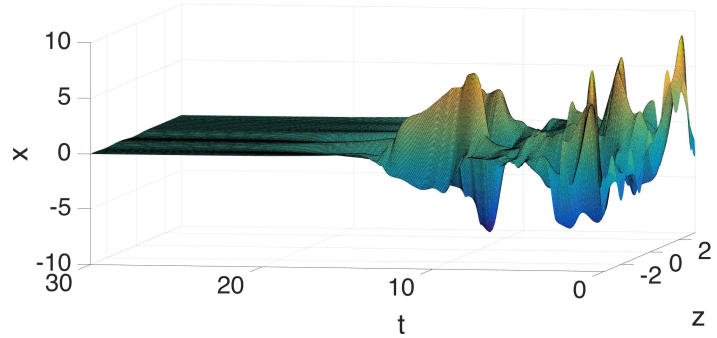


Figure 5.10b: closed-loop profile of the system based on APOD (30 sensors).

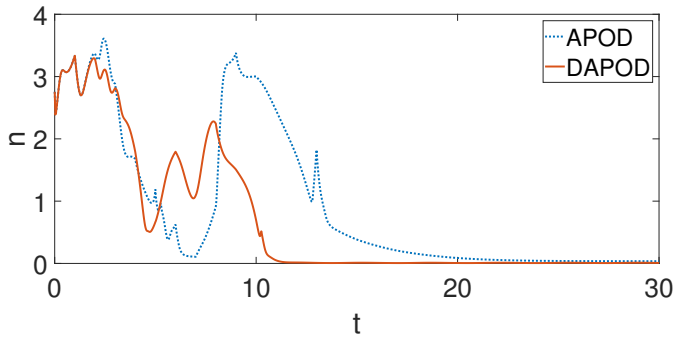


Figure 5.11: norm of the profile of systems based on DAPOD and APOD (30 sensors).

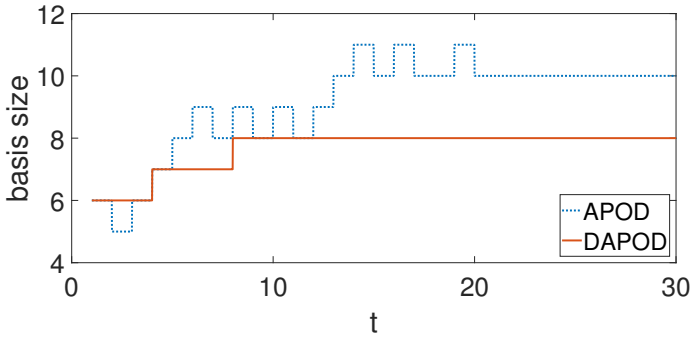


Figure 5.12: basis size in systems based on DAPOD and APOD (30 sensors).

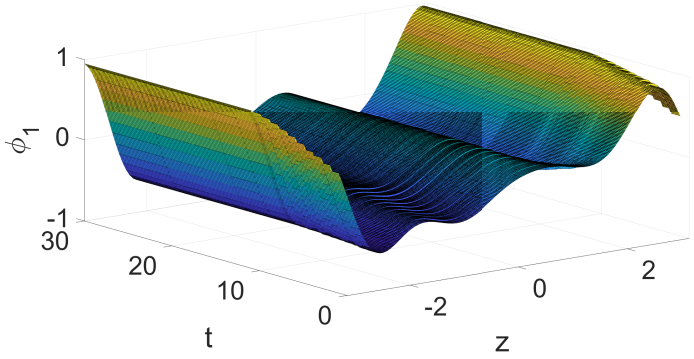


Figure 5.13a: temporal profile of the first basis function in DAPOD (30 sensors).

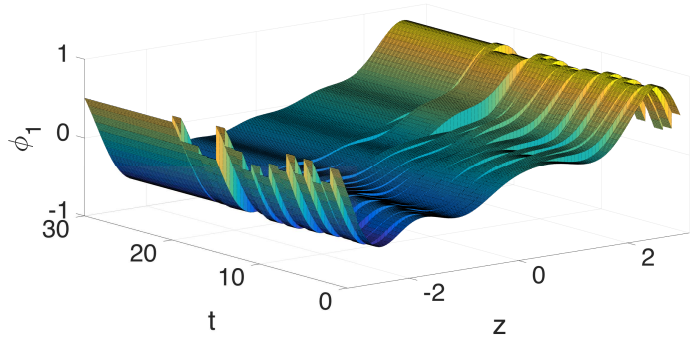


Figure 5.13b: temporal profile of the first basis function in APOD (30 sensors).

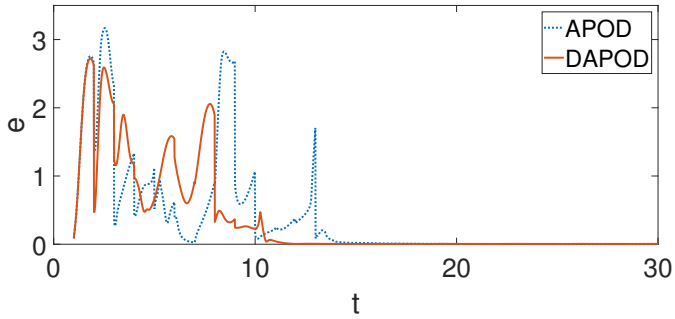


Figure 5.14: error of the observer for systems based on DAPOD and APOD (30 sensors).

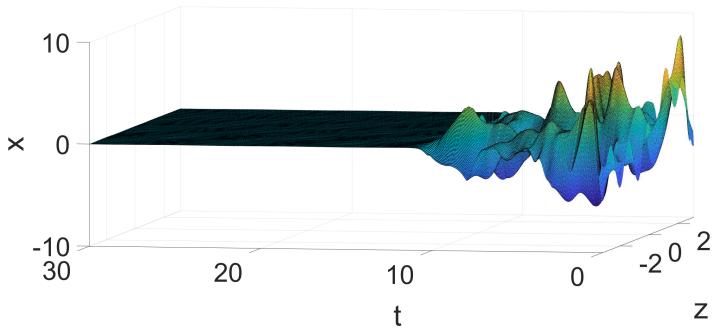


Figure 5.15a: closed-loop profile of the system based on DAPOD (20 sensors).

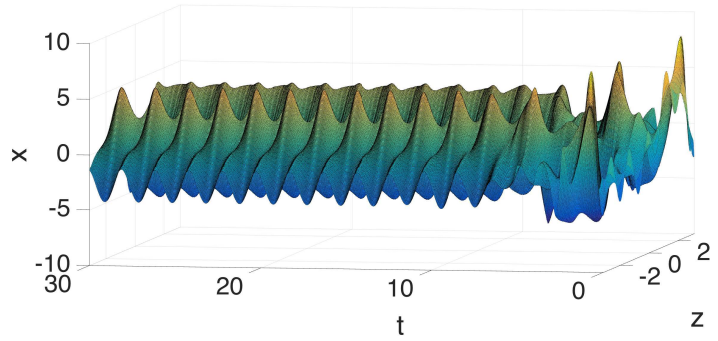


Figure 5.15b: closed-loop profile of the system based on APOD (20 sensors).

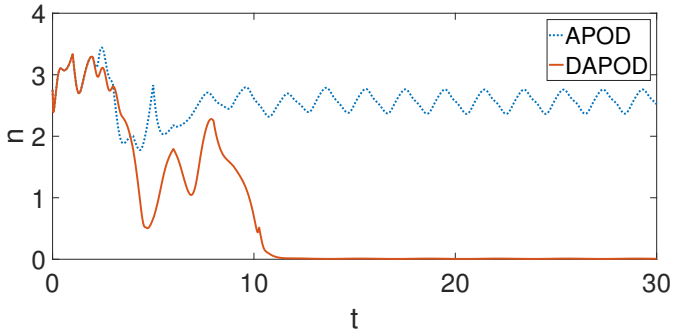


Figure 5.16: norm of the profile of systems based on DAPOD and APOD (20 sensors).

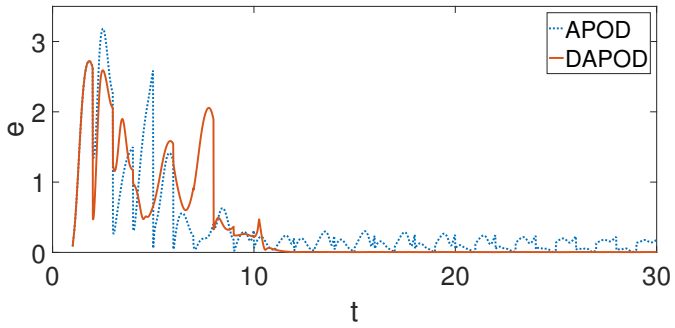


Figure 5.17: error of the observer for systems based on DAPOD and APOD
(20 sensors).

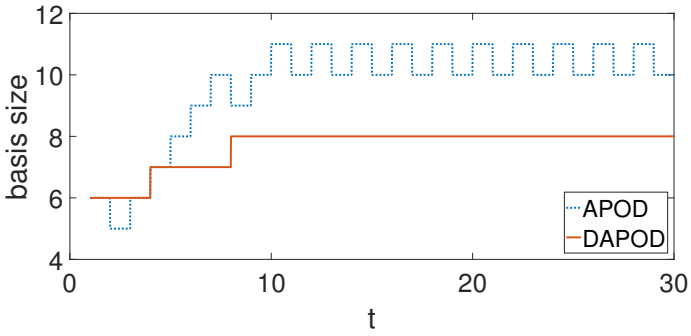


Figure 5.18: basis size in systems based on DAPOD and APOD (20 sensors).

Chapter 6
Control of Dissipative PDE Systems with Strong Convective Phenomena Based on Model Reduction

Actuation in PDEs can be divided into 3 categories: source terms/interior control, boundary control and coefficients [56]. Most of the studies focus on source terms and boundary control, while control via coefficients has been rarely discussed. To improve the process controllability, this work focuses on control by manipulating the convection coefficient. However, when a system has strong convective phenomenon, the accuracy of the corresponding reduced order model may suffer.

Conventional controller designs are based on spatial discretization techniques. Finite difference method (FDM) can be used to construct a system of ordinary differential equations (ODEs). In the work of Dejan Bošković *et al.* [57], backstepping is utilized to control a tubular chemical reactor based on the finite dimensional model generated by FDM. Despite its wide usage, this approach usually leads to a high order system. Since the dynamic behavior of systems described by parabolic PDEs can be approximated by finite dimensional systems [10, 11], low order models can be obtained via method of weighted residuals (MWR) [7, 9]. However, as convection becomes more and more dominant compared with diffusion, the accuracy of the reduced order model of a certain order decreases since the energy captured by each mode becomes closer. In the extreme case in which convection is dominant and

diffusion is negligible, this approach is no longer applicable, since all the modes have almost the same energy [58, 59].

To deal with the ROM accuracy issue, we propose to use adaptive proper orthogonal decomposition (APOD) [31] to update the ROM when new observation becomes available. To generate ROM, MWR approximates the state variable by superposition of basis functions multiplied by time dependent coefficients. Proper orthogonal decomposition (POD) [12] is a standard method to construct basis functions numerically by using an ensemble of data of state profile collected from experiment or simulation. Based on whether basis functions are updated by incorporating new observations, POD can be divided into two categories: batch POD [37] and incremental POD. Most research about POD method focuses on batch POD. Compared with standard POD [12], which process all the data in a batch, one of the advantages of incremental POD is it can mitigate the impact of historic observations by successively updating basis function and enabling dimension increase or decrease when new observation becomes available. Since applying POD iteratively is computational intensive, a significant amount of research has focused on designing incremental POD methods with less computational cost, including APOD, adaptive POD (aPOD) [38], recursive POD [39], incremental POD (iPOD) [40], and incremental SVD/POD method [30]. To reduce the computational cost, different assumptions are made in these methods. In our work, we focus on APOD.

In this work, the performance of APOD in systems with strong convective phenomena is evaluated through a tubular reactor example. First, a ROM of a system with time-varying velocity is constructed via APOD to demonstrate the improvement in the accuracy of ROM. Then velocity is used as actuator to control the system. It is also demonstrated that APOD can improve the accuracy of the ROM when unexpected disturbance occurs.

6.1 Approach and Techniques

We focus on processes described by the following semi-linear partial differential equation.

$$\frac{\partial x}{\partial t} = \theta_1 \frac{\partial x}{\partial z} + \theta_2 \frac{\partial^2 x}{\partial z^2} + f(x) + b(z)u \tag{6.1}$$

Here θ_1 is a convection parameter, θ_2 is a diffusion parameter

6.2 Adaptive Model Reduction

6.2.1 Discrete Adaptive Proper Orthogonal Decomposition

In this section, we briefly review DAPOD. Compared with standard POD that used a predetermined set of snapshots to construct basis functions, DAPOD updates basis functions by updating the snapshot ensemble when new snapshots become available. It can be summarized into the following steps:

- construct basis functions off-line
- incorporate new snapshots
- determine basis function size
- check the accuracy of ROM
- update basis functions
- eliminate old snapshot(s)

In the off-line initial basis function construction step, we can use singular value decomposition (SVD) to construct the basis function Φ, and initial basis function size is determined by energy captured by each basis function.

In the on-line step, snapshot matrix A is updated when a new snapshot becomes available. Both importance and "freshness" of snapshots are considered.

As new snapshots become available, the number of basis functions needed to construct ROM may increase or decease. We change the basis function size accordingly based on the energy captured by the basis functions truncated in previous steps.

After calculating the energy captured by the basis functions, we check the accuracy of basis functions if the basis function size is unchanged. When basis functions size changes or current basis functions are not accurate enough, basis functions are updated.

6.2.2 DAPOD-based model reduction

We use method of weighted residual (MWR) to approximate the PDE by a set of ODEs. State variable x is approximated by

$$x = \sum_{i=1}^{m} c_i(t)\phi_i(z) \tag{6.2}$$

where c_i are called modes, weight functions are chosen to be the same with basis functions $\phi_i(z)$. Both of them are determined by DAPOD. By multiplying the both sides of the PDE and integrating over the spatial domain, we obtain the following ROM

$$\frac{\mathrm{d}C}{\mathrm{d}t} = \theta_1 A_1 C + \theta_2 A_2 C + F(C) + Bu \tag{6.3}$$

where $C = [c_1, c_2, \cdots, c_m]^T \in \mathbb{R}^{m\times 1}$, $A_1,\ A_2 \in \mathbb{R}^{m\times m}$ and $F = [f_1, f_2, \cdots, f_m]^T$ The ROM will be updated when basis functions Φ are updated by DAPOD to increase the accuracy of the model.

6.2.3 DAPOD-based observer design

In this paper, we present two dynamic observes to estimate the state of the system. The dynamic observer is designed based on the ROM (Eq. 6.3)

Assumption 3 *F is a Lipschitz nonlinear vector function:*

$$\|F(c_p) - F(c_q)\| \leq K_l \|c_p - c_q\| \tag{6.4}$$

The Luenberger-type dynamic observer is formulated as

$$\frac{\mathrm{d}\tilde{C}}{\mathrm{d}t} = \theta_1 A_1 \tilde{C} + \theta_2 A_2 \tilde{C} + F(\tilde{C}) + Bu + G_m(y_m - \tilde{y}) \tag{6.5}$$

where $\tilde{C}$ refers to the estimated state. The gain matrix G_m is determined using LQR theory [42] to ensure stability of the estimation error.

When the periodical measurement y_p is available, estimated state is reset:

$$\tilde{C}_i = (y_p, \phi_i) \tag{6.6}$$

6.2.4 DAPOD-based controller design

In this section, we use Lyapunov-based control to regulate the systems. To improve the process controllability, control via coefficients is considered: both θ_1 and u are manipulated input in Eq. 6.3. Since Peclet number should be positive, input constraint is applied.

For a Lyapunov function $\mathscr{V}$, we design the following feedback regulator based on refs 21, 22. First, we use F_0 to denote $\theta_2 A_2 C + F(C)$ in Eq. 6.3; $\hat{B}\hat{u}$ refers to $\theta_1 A_1 C + Bu$, $\hat{u}$ contains both θ_1 and u

$$\dot{C} = F_0(C) + \hat{B}\hat{u} \tag{6.7}$$

To reduce chattering, we modify the method for the case $\|(L_{\hat{B}}\mathscr{V})'(C)\| = 0$. The following bounded control law is employed:

$$\begin{cases} \hat{u} = -k(C)(L_{\hat{B}}\mathscr{V})^T(C) & \|(L_{\hat{B}}\mathscr{V})^T(C)\| > 0 \\ \dot{\hat{u}} = 0 & \|(L_{\hat{B}}\mathscr{V})^T(C)\| = 0 \end{cases} \tag{6.8}$$

where

$$k(C) = \frac{L^*_{F_0}\mathscr{V}(C) + \sqrt{\left(L^*_{F_0}\mathscr{V}(C)\right)^2 + (\hat{u}_{max}\|(L_{\hat{B}}\mathscr{V})^T(C)\|)^4}}{\|(L_{\hat{B}}\mathscr{V})^T(C)\|^2\left[1 + \sqrt{1 + (\hat{u}_{max}\|(L_{\hat{B}}\mathscr{V})^T(C)\|)^2}\right]} \tag{6.9}$$

and

$$L^*_{F_0}\mathscr{V}(C) = L_{F_0}\mathscr{V}(C) + \rho_0\mathscr{V} \tag{6.10}$$

ρ_0 is a positive number. It can be proved the Lyapunov function $\mathscr{V}(C)$ is negative definite when the initial condition satisfies $\{C \in C_0 : L^*_{F_0}\mathscr{V}(C) \leq \hat{u}_{max}\|(L_{\hat{B}}\mathscr{V})^T(C)\|\}$.

6.3 Application to Tubular Reactor Example

We apply DAPOD to a tubular reactor example [60], in which a first order reaction takes place. Both temperature and concentration of reactant vary in space.

6.3.1 ROM for System with Time-varying Velocity

The governing equation is as follows:

$$\begin{aligned}\frac{\partial T}{\partial t} =& - v\frac{\partial T}{\partial z} + \frac{k}{\rho c_p}\frac{\partial^2 T}{\partial z^2} + \frac{-\Delta H}{\rho c_p}k_0 \mathrm{e}^{-\frac{E}{RT}}C_A - \frac{hA_s}{\rho c_p}(T - T_c)\\ \frac{\partial C_A}{\partial t} =& - v\frac{\partial C_A}{\partial z} + D_A\frac{\partial^2 C_A}{\partial z^2} - k_0 \mathrm{e}^{-\frac{E}{RT}}C_A\end{aligned} \tag{6.11}$$

The boundary condition of this problem is

$$\begin{aligned} z = 0 : T &= T_0\\ C_A &= C_{A0}\\ z = L : \frac{\partial T}{\partial z} &= 0\\ \frac{\partial C_A}{\partial z} &= 0\end{aligned} \tag{6.12}$$

After defining new variables, the governing equation 6.11 is simplified to

$$\begin{aligned}\frac{\partial x_1}{\partial t} =& u_1\frac{\partial x_1}{\partial z} + \beta_{12}\frac{\partial^2 x_1}{\partial z^2} + \beta_{13}\mathrm{e}^{-\frac{\beta_e}{1+x_1}}(1 + x_2) + \beta_T(u_2 - x_1)\\ \frac{\partial x_2}{\partial t} =& u_1\frac{\partial x_2}{\partial z} + \beta_{22}\frac{\partial^2 x_2}{\partial z^2} + \beta_{23}\mathrm{e}^{-\frac{\beta_e}{1+x_1}}(1 + x_2)\end{aligned} \tag{6.13}$$

where

$$\begin{aligned} &x_1 = \frac{T - T_0}{T_0}, x_2 = \frac{C_A - C_{A0}}{C_{A0}}, \tilde{z} = \frac{z}{L}\\ &u_1 = -\frac{v}{L}, u_2 = \frac{T_c - T_0}{T_0}, \beta_{12} = \frac{k}{\rho c_p L^2}, \beta_{13} = \frac{-\Delta H k_0 C_{A0}}{T_0 \rho c_p}\\ &\beta_{22} = \frac{D_A}{L^2}, \beta_{23} = \frac{D_A}{L^2}, \beta_e = \frac{E}{RT_0}, \beta_T = \frac{hA_s}{\rho c}\end{aligned} \tag{6.14}$$

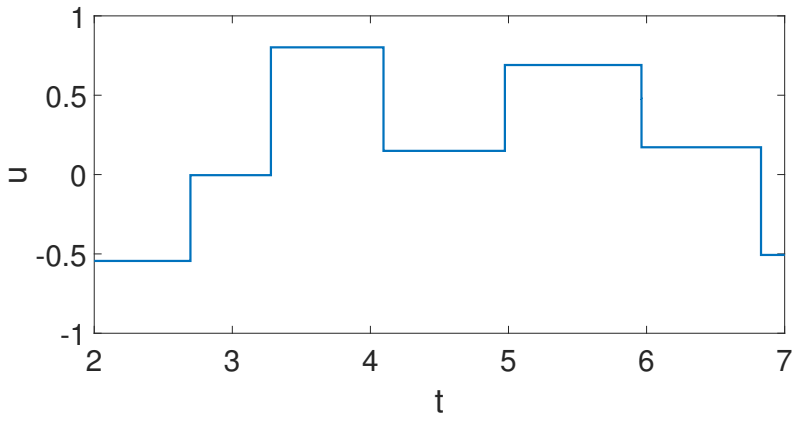


Figure 6.1: temporal profile of the random manipulated input

The corresponding boundary condition is

$$\begin{aligned} z = 0 : x_1 &= 0 \\ x_2 &= 0 \\ z = 1 : \frac{\partial x_1}{\partial z} &= 0 \\ \frac{\partial x_2}{\partial z} &= 0 \end{aligned} \tag{6.15}$$

First, to demonstrate that strong convective phenomena may make the standard model reduction method fail, we use the same snapshots to construct basis functions, and then apply the basis functions to systems with different θ_1. The systems are excited by the same random manipulated input (Fig. 6.1). Fig. 6.2 and 6.3 give the system response in full model when $\theta_1 = 3$. The system response calculated by reduced order model is displayed in Fig. 6.4 and 6.5. The error of the ROM is defined as follows:

$$e = \sqrt{\frac{\int_0^L (x - \tilde{x})^2 \mathrm{d}z}{L}} \tag{6.16}$$

where $\tilde{x}$ represents the system response calculated by ROM and L is the length of reactor, which is 1 in this case. We compare the error of the ROM when $\theta_1 = 3, 10, 20$ (Fig.6.6-6.7). It can be seen that as θ_1 increases or the convective phenomena become stronger, the error of the ROM increases.

Then we demonstrate updating basis function using new snapshots can improve

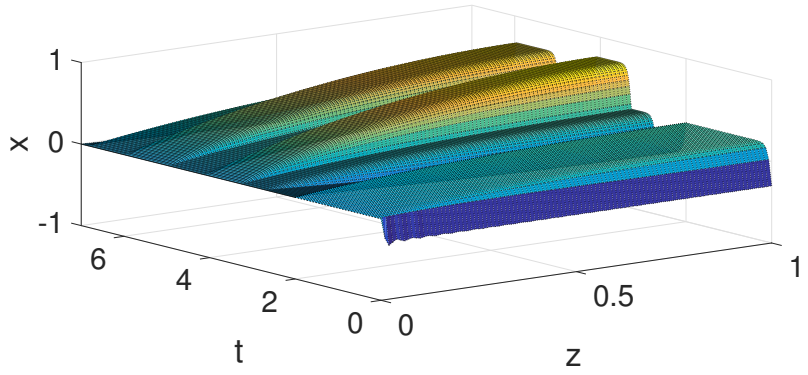


Figure 6.2: system response of x_1

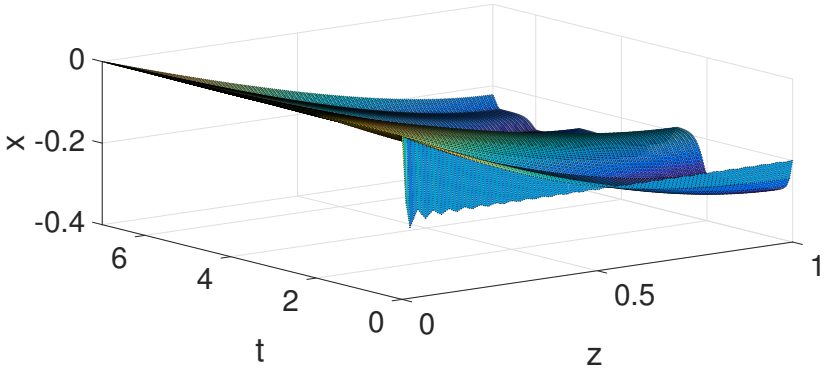


Figure 6.3: system response of x_2

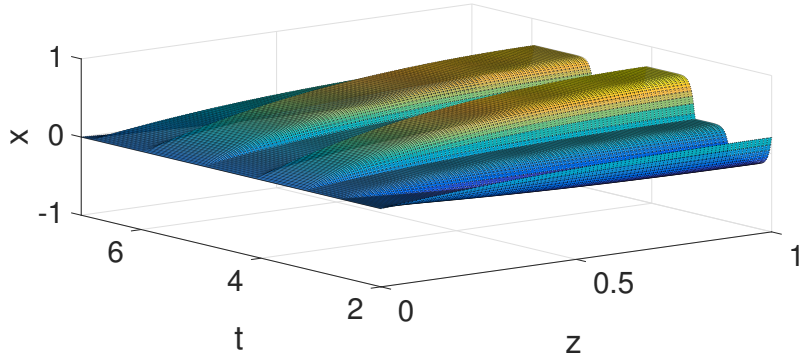


Figure 6.4: x_1 in ROM

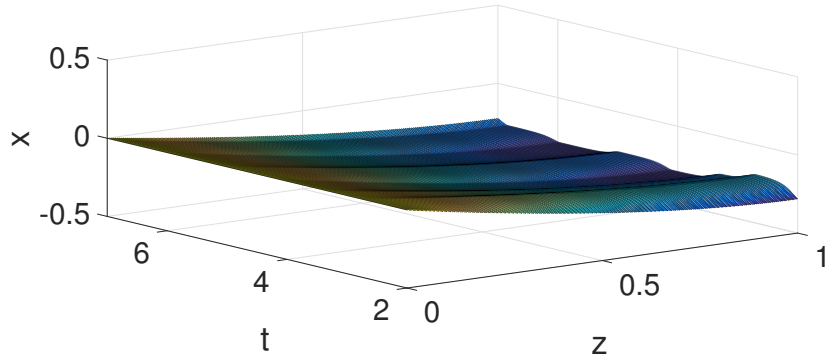


Figure 6.5: x_2 in ROM

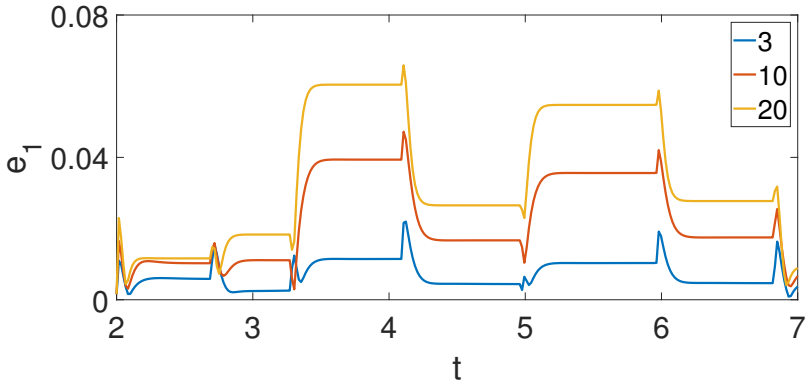


Figure 6.6: error of x_1 in ROM

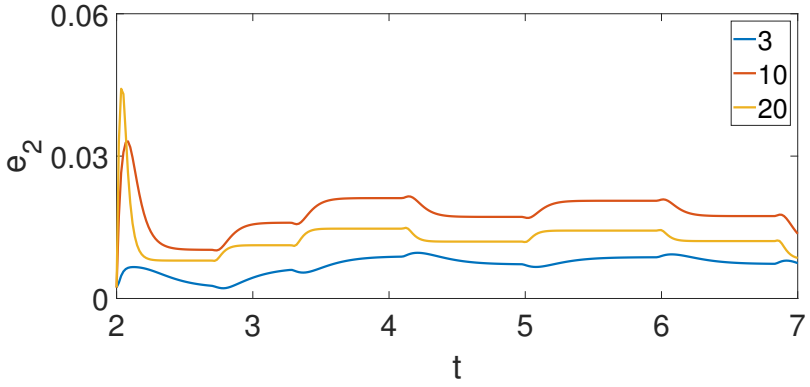


Figure 6.7: error of x_2 in ROM

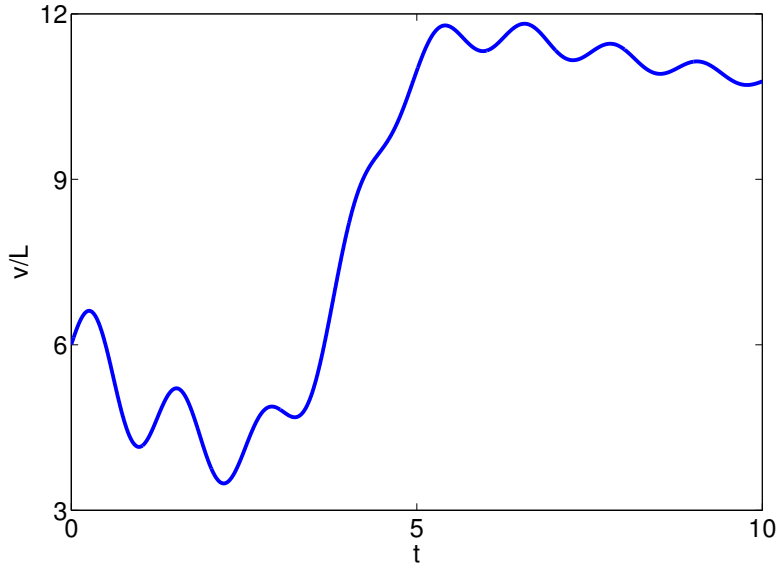


Figure 6.8: temporal profile of $-\theta_1$

the accuracy of ROM. We use DAPOD to construct ROM for the process with unknown time-varying velocity (displayed in Fig.6.8, $\theta_1 = v/L$ in this example.) Standard POD is used as a comparison. Since model accuracy depends on model dimension, the dimension of ROM when using standard POD is the same with that in DAPOD. Dimension of ROMs or numbers of basis functions are provided in Fig. 6.9. The basis functions in standard POD are also used as the initial basis functions in DAPOD. Because the snapshots are collected from a process with constant velocity, updating basis functions using new online snapshots can better capture the current behavior of the system.

The errors of ROMs are displayed in Fig. 6.10 and 6.11, where red dots indicate basis functions in DAPOD are updated. This result demonstrates that DAPOD can improve the accuracy of ROM.

Note that in DAPOD, the estimated state of the ROM in the dynamic observer is reset every time a new snapshot is available, to eliminate the effect of the resetting step, the state of the ROM in standard POD is also reset. As a result, the error of ROM reduces every 1 unit of time.

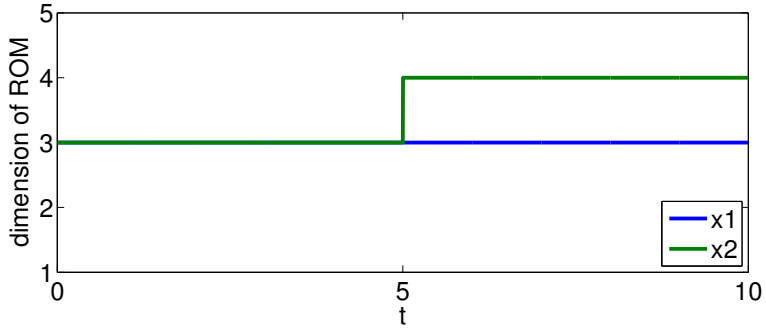


Figure 6.9: dimension of ROM

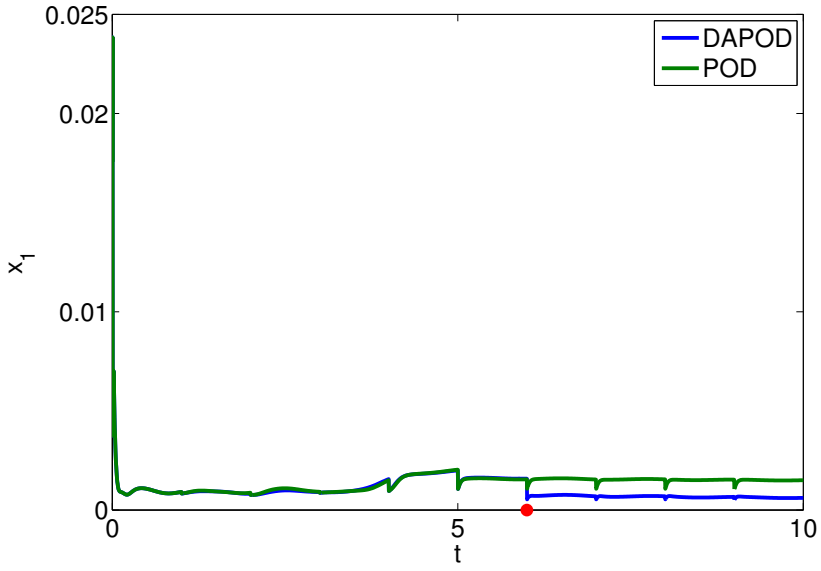


Figure 6.10: error of x_1 in ROMs based on DAPOD and POD

6.3.2 Lyapunov-based Control

In this section we use both cooling jacket (u_2 in Eq. 6.13) and velocity ($-u_2 L$ in Eq. 6.13 or $-\theta_1 L$ in Eq. 6.1 where L is a constant) to control the system. The objective is to maintain the temperature at $x_1(z) = 0$ and the concentration of A at outlet is $x_2(1) = -0.1$. Snapshots or profilse over the whole domain are assumed to be available periodically and one sensor is lo located at $z = 1$ to measure the temperature continuously. State variables of ROM are estimated by Luenberger observer [61] and controller is designed based on Lyapunov function [52, 53]. The limits for control actions are set to be: $u_1 \in [-10, 0]$ and $u_2 \in [-0.5, 0.5]$. The

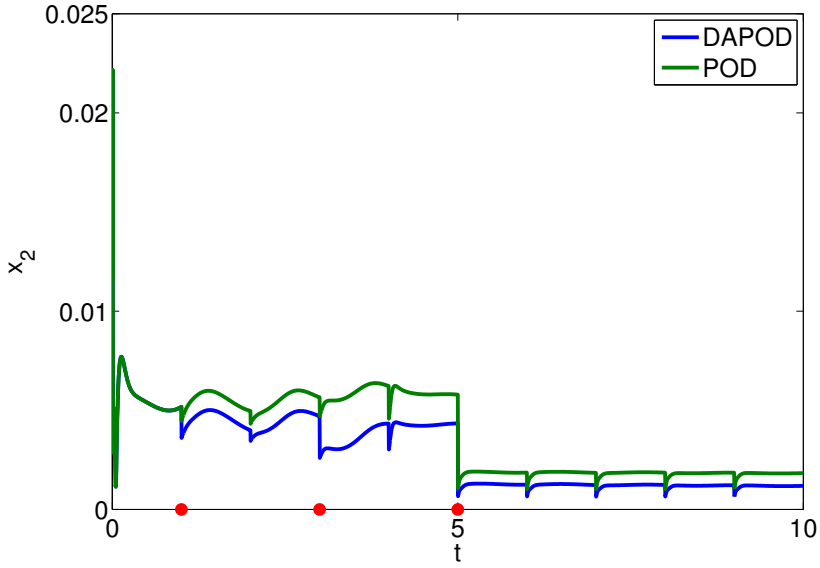


Figure 6.11: error of x_2 in ROMs based on DAPOD and POD

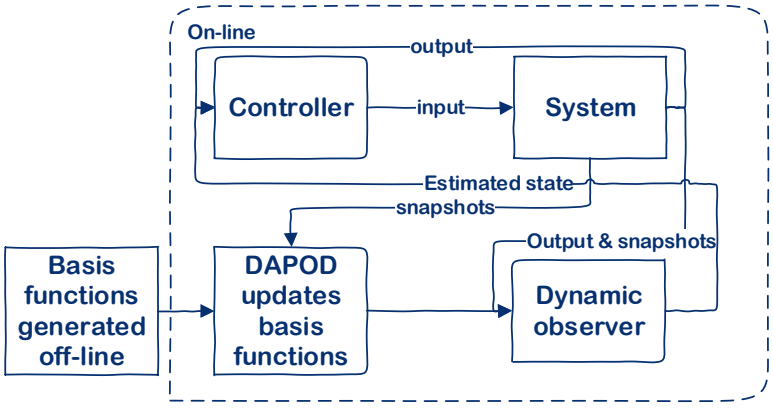


Figure 6.12: process operation block diagram

process operation block diagram is provided in Fig. 6.12.

The control actions are set to $u_1 = -7$ and $u_2 = 0$ at first 2 units of time. Then the controller becomes active. At $t = 6.6$, concentration on the boundary C_{A0} increases by 10%; as a result the boundary condition and coefficients that involves C_{A0} all change accordingly. This disturbance is not known by controller or observer and the new trend induced by this disturbance is not captured in the

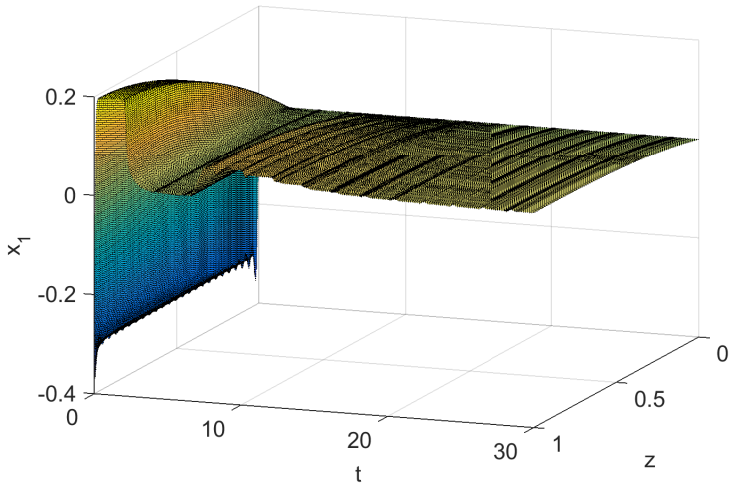


Figure 6.13: system response of x_1

initial snapshots ensemble. System response based on DAPOD for x_1 and x_2 are displayed in Fig. 6.13 and 6.14. It is clear that the controller stabilize the system at the set point $x_1(z) = 0$ and $x_2(1) = -0.1$. The error of the observer based on DAPOD and POD is compared in Fig.6.15 and 6.16. The disturbance leads to large error in dynamic observer which is based on ROM because the new trend of system behavior is not captured by ROM based on POD, while DAPOD captures the new behavior using new snapshots and the error of observer decreases gradually. Since the objectives are $x_1 = 0$ and $x_2(1) = -0.1$, we compared the norm of x_1 with set point in Fig. 6.17 and $x_2(1)$ is given in Fig. 6.18. It can be seen that both x_1 and x_2 have smaller offset in system based on DAPOD. Control action u_1 and u_2 are displayed in Fig. 6.19 and 6.20. It is clear that both manipulated variables are within the limits.

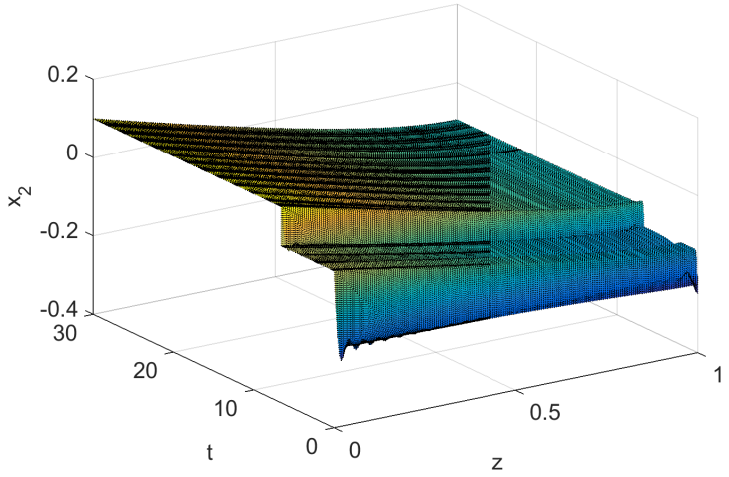


Figure 6.14: system response of x_2

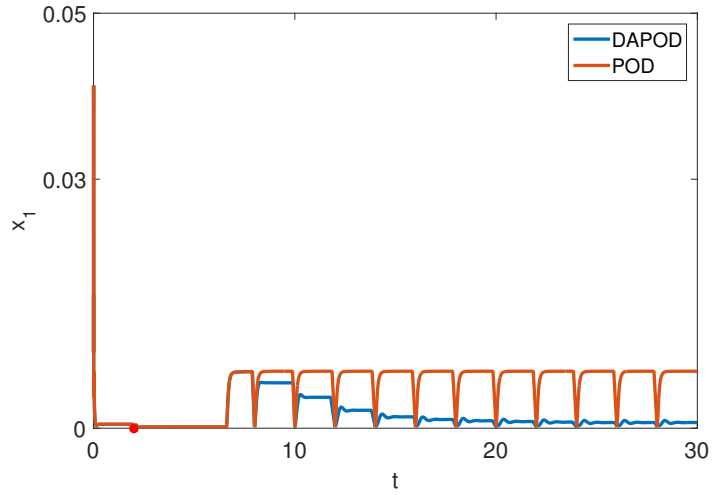


Figure 6.15: error of x_1 in ROMs based on DAPOD and POD

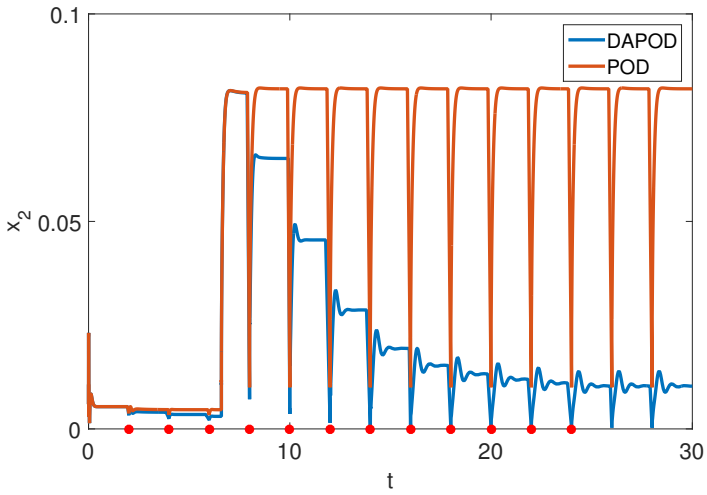


Figure 6.16: error of x_2 in ROMs based on DAPOD and POD

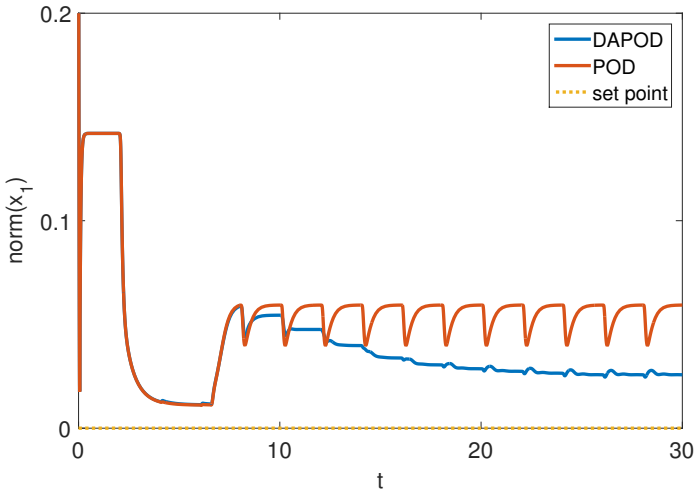


Figure 6.17: norm of x_1 in ROMs based on DAPOD and POD

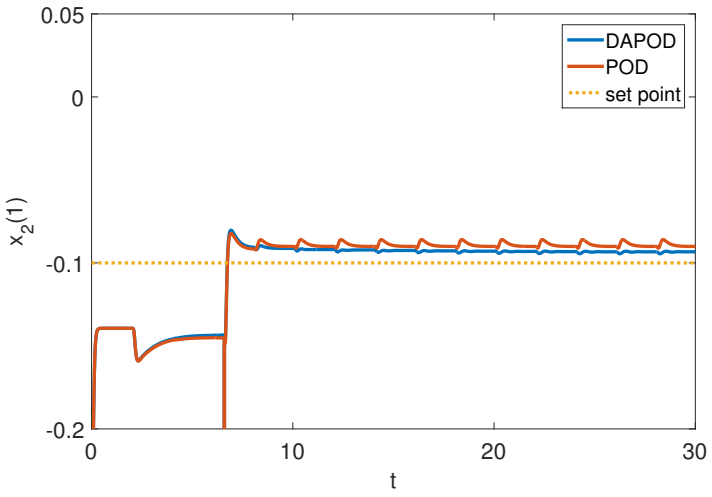


Figure 6.18: temporal profile of $x_2(1)$ in ROMs based on DAPOD and POD

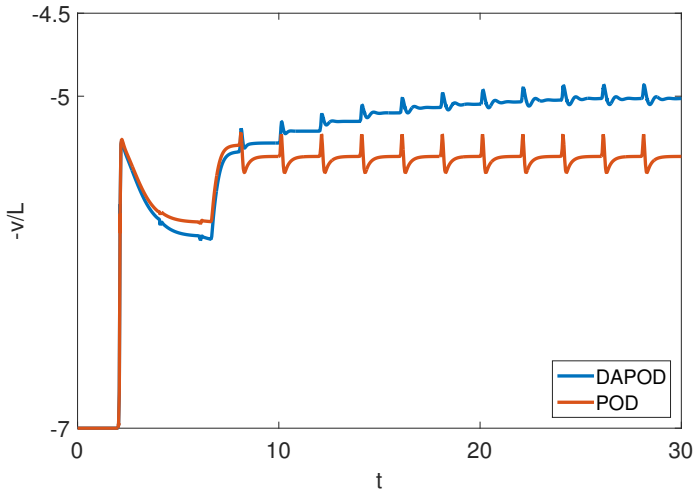


Figure 6.19: control action u_1 in ROMs based on DAPOD and POD

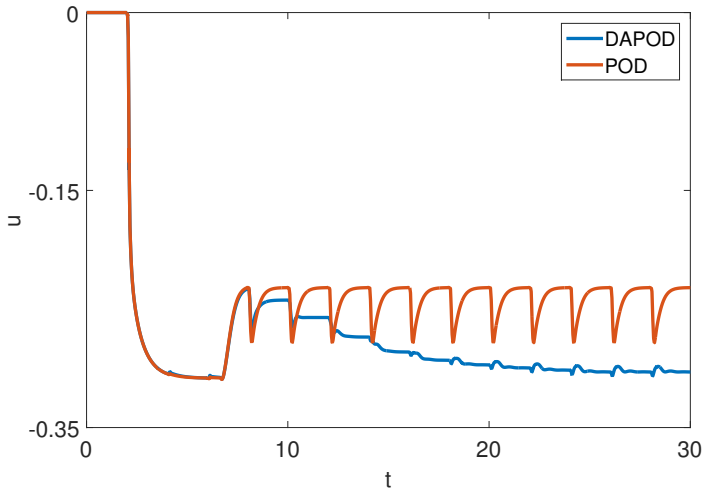


Figure 6.20: control action u_2 in ROMs based on DAPOD and POD

Chapter 7
On the Design of Equation-free Controllers for Dissipative PDEs via DEIM

Typical control methods for nonlinear distributed parameter systems include feedback linearization, the Lyapunov based method and the back stepping method [57]. A common characteristic of these methods is that they require the availability of the system state. However, distributed parameter systems are infinite dimensional in function space, making the online measurement of the complete state infeasible in practice. Although Luenberger-based observers have been widely used to estimate the state of distributed parameter systems [62, 63]. The implementation of a Luenberger-based observer requires the knowledge of the mathematical model of the system. Nevertheless, model uncertainty always exist in controlling processes. As a result, significant efforts have been devoted to control with uncertainty [64] and data-driven control [65, 66].

In most data-driven control methods, identification is utilized as the basis of control. System dynamics (local or global), or the input-output mapping is identified first. Some of the methods include divide-and-conquer techniques [67] (such as memory based local modeling [68]), subspace identification [69, 70], neural network [71, 72] and autoregressive moving average with exogenous inputs (ARMAX) [73, 74]. In divide-and-conquer techniques, linearized models at different operating points are estimated and a controller is constructed accordingly. One of the disadvantages of this algorithm is each linearized model can only correctly predict the system response in the vicinity of the corresponding operating point. Therefore, a large

number of local models are required.

Another important data-driven control method is subspace identification. Subspace identification identifies a state space model for artificial state variables by using input output data from a dynamical system. Popular algorithms include N4SID [69], canonical variate analysis (CVA) [75] and MOESP. This technique does not require any knowledge of the model, but suffers from high computational cost. To reduce the computational burden, modified versions, such as Verdult's work [76] and recursive subspace identification [77] have been developed. In addition to the excessive computation burden, another disadvantage is subspace identification applies to linear system only. To circumvent this restriction, carleman linearization is used to construct a bilinear system [78–80], which sets the basis for system identification. A shortcoming of this approach is the resulting bilinear model may be computationally expensive to implement online.

Neural network method has also been widely used to identify input-output mapping in various fields including chemical engineering over many years [81,82]. It has great promise in control nonlinear system since it can represent any continuous mapping [83]. To robustify this method in terms of the disturbance in data, prediction interval-based neural network [84, 85] has been proposed. However, how to determine the structure of the neural network and the demanding requirement in the availability of training set to tune parameters remain open questions.

ARMAX is also widely used to approximate nonlinear systems. This method has a given structure for the system it will approximate and the parameters in polynomials need to be estimated through an identifying process [86].

In this manuscript, we propose another approach to control systems in which the knowledge of the chemical and physical law that describes the systems is not available or complete. DEIM, an algorithm which is used to reduce computational cost, will be modified to estimate the state and the dynamics of the system by using the continuous measurement of a finite number of point sensors. The estimation will be employed for controller implementation. The idea is the proposed method and DEIM share a similar goal: determining the "optimal" locations based on snapshot profiles so that the complete profile of the interested term can be approximated by using the values of the term at finite locations.

7.1 Preliminaries

We consider a dissipative process described by the following PDE:

$$\frac{\partial x}{\partial t} = f(x) + b(z)u \tag{7.1}$$

$$y_i = \int_{\Omega_z} \delta(z - z_i) x \mathrm{d}z \ \ (i = 1, 2, \cdots, k_1) \tag{7.2}$$

$$y_j = \int_{\Omega_z} \delta(z - z_j) \dot{x} \mathrm{d}z \ \ (j = 1, 2, \cdots, k_2) \tag{7.3}$$

subject to the boundary condition

$$g_b(x, \frac{\partial x}{\partial z}) = 0 \ on \ \Gamma \tag{7.4}$$

and the initial condition

$$x(z, 0) = x_0(z) \tag{7.5}$$

In this system, $f(x)$ is a unknown functional which contains a semi-linear spatial operator and a nonlinear function. $u \in \mathbb{R}^{s \times 1}$ denotes manipulated variables, where s refers to the number of manipulated inputs. $b(z) \in \mathbb{R}^{1 \times s}$ describes how the manipulated variables u control the system spatially and is known. y_i and y_j denotes the i^{th} and j^{th} measured output, and $\delta(\cdot)$ represents the Dirac delta function. $\dot{x}$ refers to the time derivative of x. In this work, we assume only a finite number of point sensors are available. We have 2 sets of point sensors: the first set of sensors is called state sensors. They measure the state (such as temperature) at certain points directly. The other set of sensors is "velocity sensors" since they estimate or measure directly the time derivative of the state at certain points, which is used for $\{f(x_i)\}$. z_i denotes the location of ith state sensor and z_j denotes the location of ith velocity sensor. g is a function of x and its spatial derivative. The control objective is to maintain the state variable at a desired steady state. Without loss of generality, we assume $x(z, t) \in \mathbb{R}$ and the desired steady state is the zero spatially uniform profile.

Assumption 4 *We assume a set of snapshots of x and $f(x)$ are available before the process operation starts.*

Remark 4 *The snapshots of $f(x)$ can be calculated by using x without the explicit*

knowledge of $f(x)$ (This will be explained more in section 8.1).

7.2 Equation-free Control Based on DEIM

Typical control methods for nonlinear distributed parameter systems include feedback linearization. To implement this method, the knowledge of the state and the time derivative of the state are required. After the ROM is constructed, the requirements convert to the knowledge of c_q and$\int_{\Omega_z} \varphi_j f(\sum_{q=1}^{n} c_q \phi_q)\mathrm{d}z$.

7.2.1 basis functions construction

We assume a set of snapshots $\{x_i\}$ are sampled from an open-loop process via experiment. $x_i = x(z, t_i)$ and $\{t_i\}$ is known. Then we use $\{x_i\}$ and $\{t_i\}$ to generate the spatial profile of the time derivative of the state $\frac{\partial x}{\partial t}(z, t_i)$. Since in the open-loop process

$$\frac{\partial x}{\partial t} = f(x) \tag{7.6}$$

each profile of $\frac{\partial x}{\partial t}(z, t_i)$ is denoted by $f(x_i)$. Please note that the explicit knowledge of $f(x)$ is not required to generate $\{f(x)\}$ (which is different from DEIM) since $f(x)$ is equal to the time derivative of x when the control action is set to zero. When zero-control-action does not hold, $f(x)$ can be calculated from $\dot{x} - bu$ as long as spatial profile of $\dot{x}$ can be calculated from x_i and t_i. Then we apply POD to $\{x_i\}$ and $\{f(x_i)\}$ respectively to generate 2 sets of basis functions: $\{\phi_q\}$ for $\{x_i\}$ and $\{\tilde{\phi}_q\}$ for $\{f(x_i)\}$.

Remark 5 *In DEIM, basis functions are constructed for $g(x)$ only, which is the nonlinear part in $f(x)$, while in our method, the linear part and nonlinear part are not differentiated. This modification has an advantage: it does not require the knowledge of the linear operator $\mathcal{L}$ and the nonlinear function $g(x)$, while DEIM requires the explicit information of the governing equation.*

To the best of the authors' knowledge, DEIM is mainly used in reducing computational cost [87, 88]. This is the first time DEIM is used in equation free control.

Assumption 5 *The nonlinear function g can be represented by basis functions*

Remark 6 *In theory, a general nonlinear function in the governing equation of a dissipative process might be infinite dimensional, and therefore cannot be approximated by a linear combination of basis functions. This problem may be solved by applying localized DEIM [88] or adaptive DEIM [89] to improve the accuracy. Since DEIM has been widely used in reducing computational cost in different systems with enough accuracy, we can assume for many dissipative processes, the nonlinear part g can be approximated by Eq.4.1 (assumption 5). In the linear part,*

$$\mathcal{L}(z)x \approx \mathcal{L}(z)\sum_{q=1}^{n} c_q(t)\phi_q(z) = \sum_{q=1}^{n} c_q(t)\Big(\mathcal{L}(z)\phi_q(z)\Big) \tag{7.7}$$

Hence, the difference between the dimension of the f and g is at most n for a certain level of accuracy.

Remark 7 (snapshots) *How to select snapshots and how many of them should be used remain an open question in POD, which is out of the scope of this paper. Because of the limit of space, we only present the work in which snapshots of x and $f(x)$ are generated in the same process and the same number of them are used.*

7.2.2 determine sensor locations by using DEIM algorithm

After we generate basis functions for $\{x_i\}$ and $\{f(x_i)\}$, point sensor locations are determined by linking the sensor locations to the interpolation indices in the DEIM algorithm.

First we put the basis functions $\{\phi_q\}$ in order based on the corresponding eigenvalues of the covariance matrix. Then we calculate those indices inductively. p_1 is the position of the peak of the first nonlinear basis function ϕ_1. We define 2 matrices $\Phi' = [\phi_1]$ and $P' = [e_{p_1}]$, where e_{p_1} denotes the p_1^{th} column in an identity matrix. Starting from the second iteration, we solve

$$(P'^T U')d = P'^T \phi_q \tag{7.8}$$

for d to obtain residual r

$$r = \phi_q - \Phi' d \tag{7.9}$$

the q^{th} interpolation index is the position of the peak of the residual. Then we update Φ' to $[\Phi', \phi_q]$ and P' to $[P', \boldsymbol{e}_{p_i}]$. We repeat this process until we find out

all the interpolation indices. The locations of first set of sensors correspond to the interpolation indices for $\{\phi_q\}$. Then we repeat the above process for $\{\tilde{\phi}_q\}$ to determine the locations of the second set of sensors.

7.2.2.1 Observer & Controller

We design static observer for the system. Based on Eq. 4.3, we have

$$x \approx \Phi(P^T\Phi)^{-1}X \tag{7.10}$$

where $X = [x(z_1)\ x(z_2)\ \cdots\ x(z_n)]^T$ are continuously measured by the first set of sensors, and P is determined by the DEIM algorithm described in section 8.1.2. To obtain the state in the ROM, Φ is multiplied at both sides:

$$\Phi^T x \approx \Phi^T\Phi(P^T\Phi)^{-1}X \tag{7.11}$$

We obtain

$$c \approx (P^T\Phi)^{-1}X \tag{7.12}$$

Similarly, we can estimate $f(x)$ by using the same approach:

$$f(x) \approx U'(P'^TU')^{-1}X' \tag{7.13}$$

where $U' = [\tilde{\phi}_1, \cdots, \tilde{\phi}_k]$, P' is obtained based on U' and

$$X' = [\dot{x}(z'_1) - b(z'_1)u,\ \ \dot{x}(z'_2) - b(z'_2)u,\ \cdots\ \dot{x}(z'_n) - b(z'_n)u]^T \tag{7.14}$$

where $\dot{x}(z'_i)$ is measured or estimated by the second set of sensors.

The controller will be designed using feedback linearization. With the estimated state (Eq.8.4) and $f(x)$ (Eq. 8.5), these control methods can be implemented without the explicit information of $f(x)$.

7.3 Diffusion-reaction process

To illustrate how the estimated information is used in controller design and evaluate the performance of the proposed static observer, we consider a diffusion reaction

process. In this process, the temperature along the surface of a catalytic rod is affected by an elementary exothermic reaction and diffusion [31]. The governing equation is as follows:

$$\rho C\frac{\partial T}{\partial \tilde{t}} = k_1\frac{\partial^2 T}{\partial z^2} - Ak_2(z)\mathrm{e}^{-\frac{E}{RT}}c_0\Delta H + \frac{k_3 l_s}{\Delta A}(T_c(z) - T) \tag{7.15}$$

where concentration c_0 and parameters ρ, C, ΔA, E, R, A, k_1, k_3, l_s are all constants. The objective is to force the temperature to a uniform unstable steady state $T = T_0$. To regulate the temperature, cooling (or heating), T_c is employed. T_c contains 2 components:

$$T_c = T_{c0} + T_c' \tag{7.16}$$

where T_{c0} is a cooling system determined during process design, chosen so that $T = T_0$ is a viable (albeit unstable) open-loop steady state and T_c' is the manipulated input.

New variables are defined to simplify Eq. 7.15. We obtain:

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_T(z)(\mathrm{e}^{-\gamma/(1+x)} - \mathrm{e}^{-\gamma}) + \beta_u(b(z)u - x) \tag{7.17}$$

where

$$x = \frac{T - T_0}{T_0} \tag{7.18}$$

The values of new parameters are $\beta_u = 2, \gamma = 2$. Because of a spatial distribution on the activity of the catalyst, we have

$$\beta_T(z) = 16[\cos(z) + 1] \tag{7.19}$$

We consider 2 boundary conditions:

1)

$$x(0,t) = 0,\ x(\pi,t) = 0 \tag{7.20}$$

2)

$$x(0,t) = 0,\ \frac{\partial x}{\partial z}(\pi,t) = 0 \tag{7.21}$$

In both cases the initial condition is

$$x(z,0) = 0.7 \tag{7.22}$$

Without control, the system evolves to a spatially varying steady state for both boundary conditions.

To facilitate observer and controller design, POD is utilized to construct a ROM for this process. As a result, the control objective $x = 0$ becomes $c = 0$. We use feedback linearization to regulate the unstable state variables in the ROM, which is c_1 in this case. Hence, only one actuator is required to regulate this process. We consider an actuator that affects the interval $[0.3\pi, 0.7\pi]$ uniformly.

$$b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \tag{7.23}$$

where $H(\cdot)$ denotes Heaviside step function. There are 2 sets of point sensors to continuously measure the temperature of the rod at points z_i and the time derivative of temperature at z_i'.

$$y_i = \int_0^\pi \delta(z - z_i)x(z)\mathrm{d}z \tag{7.24}$$

$$y_i' = \int_0^\pi \delta(z - z_i')\dot{x}(z)\mathrm{d}z \tag{7.25}$$

7.3.1 Snapshots

We assume a set of snapshots is available before the process operation starts. In this example, 40 snapshots of x are collected from an open-loop process with the initial condition

$$x(z, t_0) = 0.3\sin(4.3z) - 0.11\cos(1.38z) + 1 \tag{7.26}$$

which is displayed in Fig. 7.1. These 40 snapshots are also used to estimate the spatial profiles of the time derivative of x, which are the snapshots of $f(x)$ since $u = 0$.

Remark 8 *In POD method, whether the basis functions can capture the major behavior of a system hinges on the quality and quantity of snapshots. Since this example is very simple, snapshots from a single process are sufficient to stabilize the system. For more information about snapshots sampling, please refer to [15, 90].*

7.3.2 Observer & Controller

We use the static observer (Eq. 8.4) to estimate the state c. As to the controller, we design the control action so that

$$\dot{c_1} + kc_1 = 0 \tag{7.27}$$

where k is a positive constant. Based on the governing equation for c_1

$$\dot{c_1} = \int_{\Omega_z} \phi_1 f(\sum_{q=1}^{n} c_q \phi_q) \mathrm{d}z + \int_{\Omega_z} b(z) u \phi_1 \mathrm{d}z \tag{7.28}$$

we obtain the following control action

$$u = \frac{-kc_1 - \int_{\Omega_z} \phi_1 f(\sum_{q=1}^{n} c_q \phi_q) \mathrm{d}z}{\int_{\Omega_z} b(z) \phi_1 \mathrm{d}z} \tag{7.29}$$

The unknown term $\int_{\Omega_z} \phi_1 f(\sum_{q=1}^{n} c_q \phi_q) \mathrm{d}z$ is estimated by Eq. 8.5. The first state variable c_1 is estimated by the observer (Eq. 8.4).

Remark 9 *To implement the above controller, the only information we need is the actuator effect $b(z)$ and the continuous measurement by state and velocity sensors. The proposed method takes advantage of the information in snapshots to estimate the complete profile of x and $f(x)$ based on point measurement.*

Remark 10 *Since the uniform initial condition is arbitrarily chosen and shows different behavior from the snapshots in open loop, the observer has large error for a small period of time at the beginning; as a result, the control action is very aggressive at the beginning and then decays very fast. To eliminate this aggressive control action, we force the control action to be in the region $[-3, 3]$.*

7.3.3 Results

We use linear Galerkin method with analytical eigenfunctions to simulate the plant. 100 eigenfunctions are used to guarantee the accuracy of the simulation.

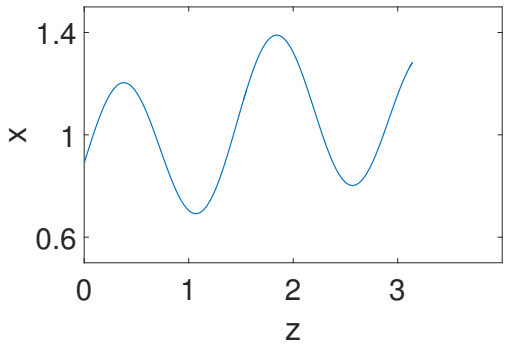


Figure 7.1: initial condition of the open-loop process.

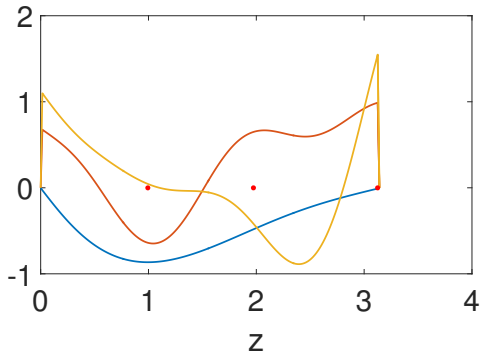


Figure 7.2a: basis functions for x and the corresponding sensor locations.

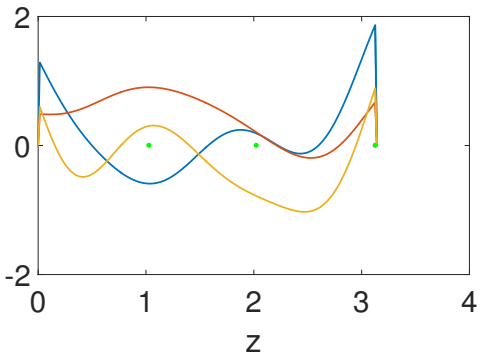


Figure 7.2b: basis functions for $f(x)$ and the corresponding sensor locations.

7.3.3.1 Dirichlet boundary condition

In the system with Dirichlet boundary condition, we first use 2 sets of 3 point sensors (in total 6 sensors) to measure the temperature. The basis functions and the corresponding sensor locations are given in Fig. 7.2a and 7.2b, where the red and green dots indicate sensor locations. Fig. 7.3 shows the system response of the closed-loop system and the temperature estimated by the observer is given in Fig. 7.4.

To analyze the impact of the number of sensors, we increase the number of sensors in each set from 3 to 4, 5, 6 and 7. The errors of observers are compared in Fig. 7.5. It shows that as the number of sensors increases, the error of observer decreases. The only exception is the error in the system with 2 sets of 4 sensor decays slower than that in 3 sensors. The reason is the control action (Fig. 7.6) is

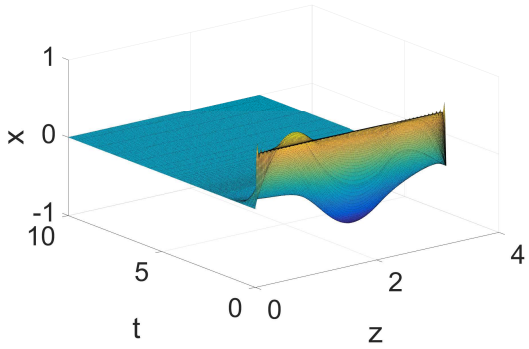


Figure 7.3: closed-loop temperature profile of the system.

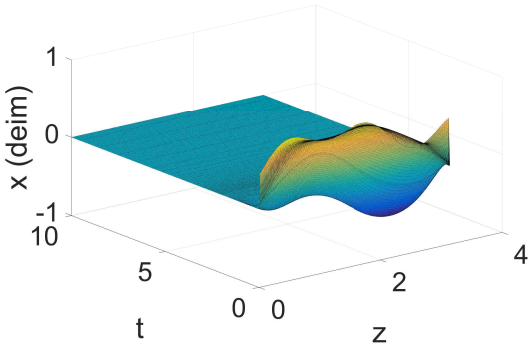


Figure 7.4: temperature estimated by the observer.

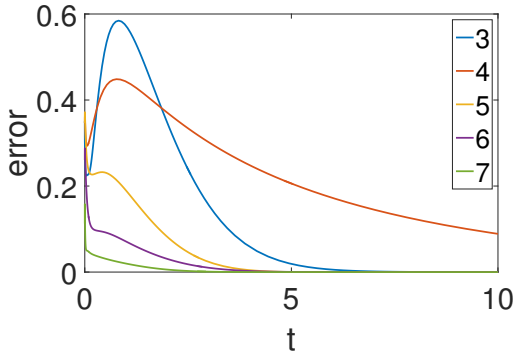


Figure 7.5: error of observer.

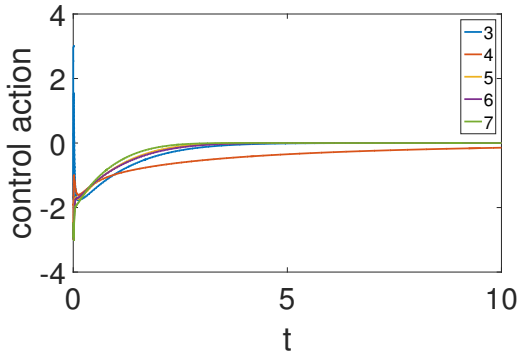


Figure 7.6: control action.

less aggressive in the system with 4 sensors.

Remark 11 *Since the dynamic behavior of dissipative PDEs can be approximated by finite dimensional systems [10, 11], the estimation of x may require less sensors than $f(x)$ to achieve the same level of accuracy. We only present the result in which x and $f(x)$ have the same number of sensors in the example because of the limit of space.*

7.3.3.2 Dirichlet-Neumann boundary condition

To evaluate the effectiveness of the method in a different circumstance, we apply the same method to the system with Dirichlet-Neumann boundary condition (Eq. 7.21). Fig. 7.7 shows the system response of the closed-loop system and the temperature estimated by the observer is given in Fig. 7.8. The error of the observers and

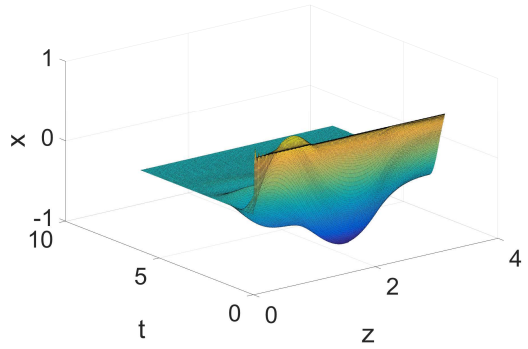


Figure 7.7: closed-loop temperature profile of the system.

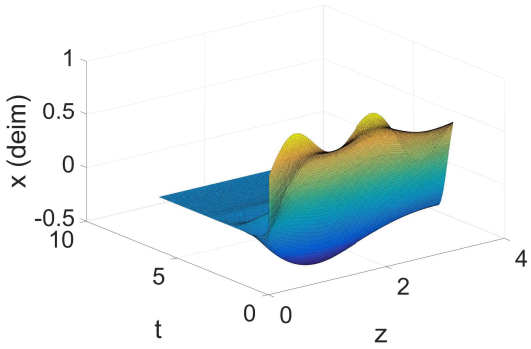


Figure 7.8: temperature estimated by the observer.

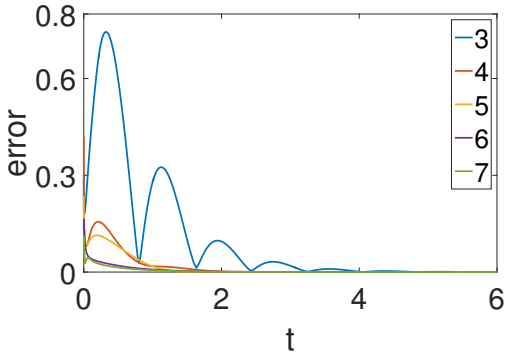


Figure 7.9: error of observer.

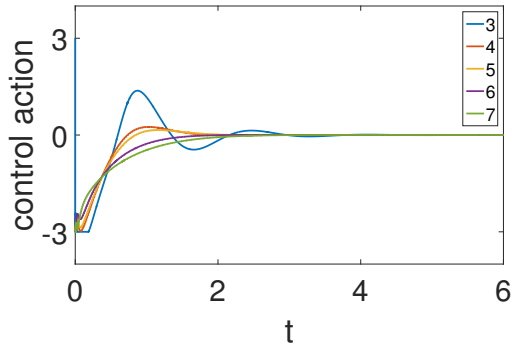


Figure 7.10: control action

the control actions in systems with a different number of sensors are compared in Fig. 7.9 and 7.10. The result shows that the controller can also regulate the system with Dirichlet-Neumann boundary condition and the results corresponding to different numbers of sensors present the similar trend with those in the Dirichlet boundary condition. The only difference is the system oscillates when it has 2 sets of 3 sensors.

7.3.3.3 Closed-loop System Evolution under Disturbances

Here we consider a system with disturbance. We assume that the activity of the catalyst changes uniformly in space over time. The temporal profile of the ratio $\frac{\beta_T(t)}{\beta_T(t_0)}$ is depicted in Fig. 7.11. We also consider the case in which the activity of the catalyst decreases. Since the decreasing of the nonlinear term makes the system easier to control, we only present the former one here.

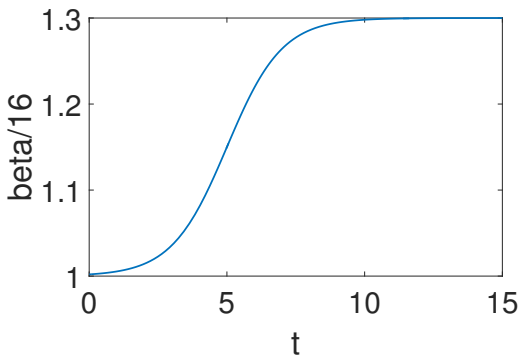


Figure 7.11: disturbance in β_T.

In the system with Dirichlet boundary condition, the system response of the closed-loop system with 2 sets of 3 sensors and the temperature estimated by the observer are given in Fig. 7.12 and 7.13. We can see that the regulator can still achieve the goal. Fig. 7.14 compares the error of the observers and the control action is given in Fig. 7.15. They show the similar trend with that in systems without disturbances.

In the system with Dirichlet-Neumann boundary condition, the method is less robust than that with Dirichlet boundary condition. When β_T increases by 20%, the controller still works (Fig. 7.16 gives the error of the observers and Fig. 7.17 displays the control action), although the errors of the observers are larger. However, when β_T increases by 30%, the second mode explodes after the system gets close to the origin; since this controller is designed based on the assumption that only the first mode is unstable, the controller is not able to regulate the system.

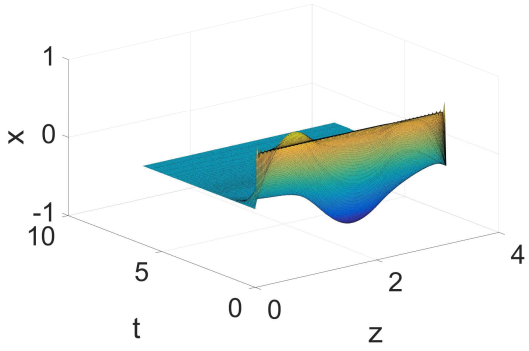


Figure 7.12: closed-loop temperature profile of the system when disturbance exists.

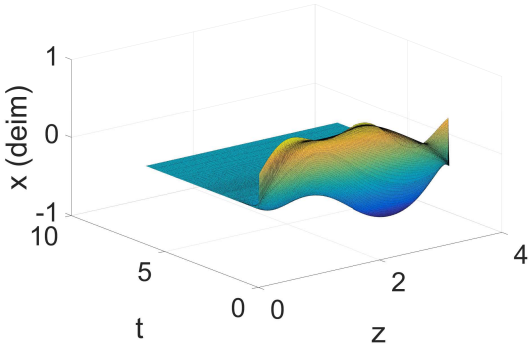


Figure 7.13: temperature estimated by the observer when disturbance exists.

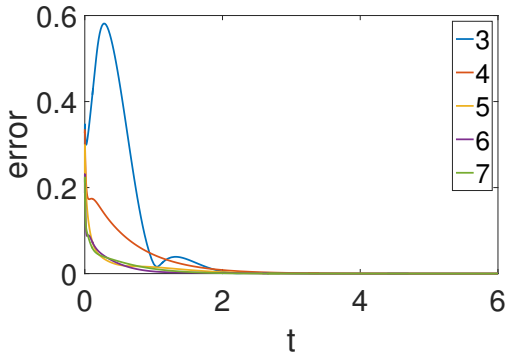


Figure 7.14: error of observer when disturbance exists. (Dirichlet boundary condition)

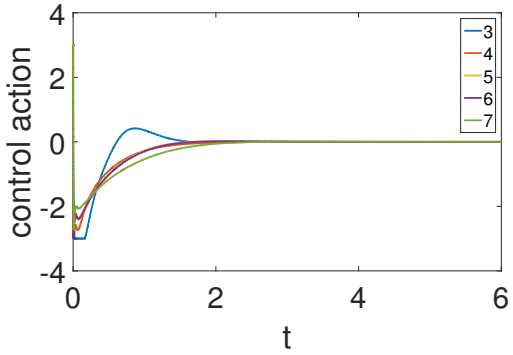


Figure 7.15: control action. (Dirichlet boundary condition)

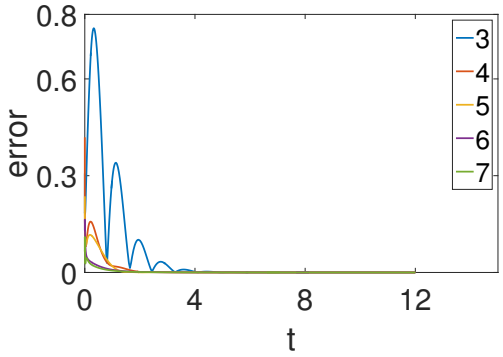


Figure 7.16: error of observer when disturbance exists. (Dirichlet-Neumann boundary condition)

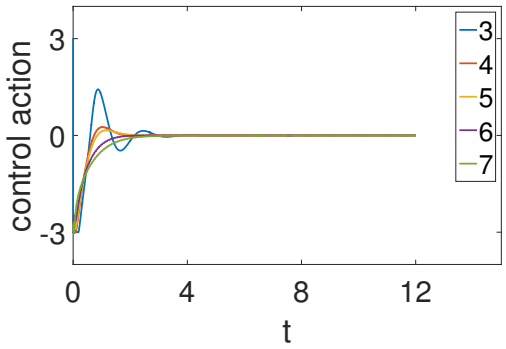


Figure 7.17: control action. (Dirichlet-Neumann boundary condition)

Chapter 8
Modified Equation-free Control Method

In previous work [91], we have described another equation-free method to control systems when the knowledge of the governing law is not available or complete and the actuator effect is. This method is motivated by the feature of discrete empirical interpolation method (DEIM) [47]: the selection of the interpolation indices can limit the growth of the error of the approximation by DEIM. To take advantage of this feature, sensor locations in this method are determined by the interpolation indices. While there have been many contributions about sensor network designs [92–95], this approach has the advantage that no information about the governing equation is required. With the continuous measurement of these sensors, the state in the ROM is estimated by a static observer. In the dynamics estimation part, the mapping from the outputs onto the projection of the dynamics is generated using a similar approach as the static observer. By using this information, the explicit governing law becomes superfluous.

In this chapter, we try to improve the accuracy of the estimation of the dynamics by using the information of estimated state. A correction term is applied to compensate the error brought by the "static" part. The difference between the expected value of the state and the actual value is used to update the correction term. The modified version is applied to a diffusion reaction process. The assumption of the availability of the constant regulation term in the previous work is relaxed.

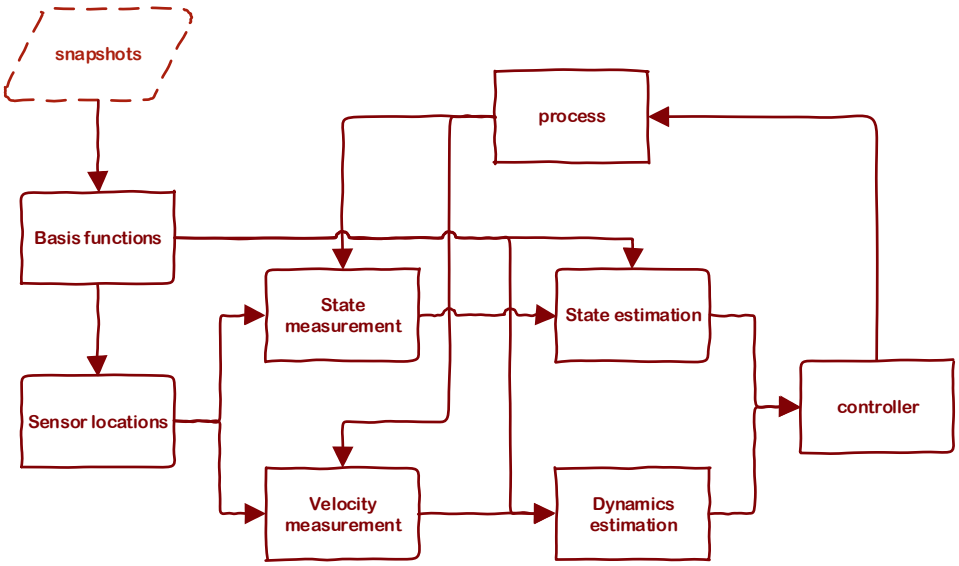


Figure 8.1: flow chart of equation free control framework.

8.1 Data-driven Control Based on DEIM

In this section, we review how the equation-free control is used to regulate systems. The flow chart of the proposed method is given in Fig. 8.1.

This method makes the following assumptions:

Assumption 6 *A representative set of snapshots of x and a set of snapshots of f, $\{x_i\}$ and $\{f_i\}$ respectively, are available before the process operation starts, or $\{f_i\}$ can be obtained from processing $\{x_i\}$.*

Assumption 7 *The effect of the actuators, $b(z)$, on the state is known.*

Assumption 8 (state space partition) *[61] The infinite dimensional system of Eq.1.1 can be partitioned into a finite dimensional slow (stable or unstable) subsystem and an infinite dimensional fast stable subsystem.*

Please note that the availability of offline f snapshots does not necessarily require the knowledge of the governing equation. As long as the sampling time corresponding to each x or $\dot{x}$ can be measured directly, f can be calculated by $f = \frac{\partial x}{\partial t} - b(z)u$.

8.1.1 basis functions construction

First, POD or SVD is applied to $\{x_i\}$ and $\{f_i\}$ respectively to generate 2 sets of basis functions: $\{\phi_i\}$ for $\{x_i\}$ and $\{\psi_i\}$ for $\{f_i\}$. To take full advantage of the continuous point measurement, the modal representation dimension for x (i.e. n) is the same with the number of state sensors k_s and the mode size for f is k_v.

8.1.2 determine sensor locations by using DEIM algorithm

During system evolution, the continuous measurement from point sensors (both state sensors and velocity sensors) are used for state and dynamics estimation. The locations of these sensors are predetermined by applying DEIM algorithm to the basis functions corresponding to $\{x_i\}$ and $\{f_i\}$. We choose DEIM to determine sensor locations because it can improve the observability when the dynamics of the system is unknwon.

Take the basis functions for $\{x_i\}$ for example: The interpolation indices for each basis functions are determined inductively using DEIM. Each interpolation index corresponds to the location of a sensor. The first interpolation index is the position of the peak of the first basis function. We define 2 matrices $\Phi' = [\phi_1]$ and $P' = [e_{p_1}]$, where p_1 is the index position of the peak of the first nonlinear basis function ϕ_1 and e_{p_1} denotes the p_1^{th} column in an identity matrix. Starting from the second one $q = 2$, the interpolation indices are the peaks of "residuals" r_0. To obtain them, first we solve

$$(P'^T\Phi')d = P'^T\phi_q \tag{8.1}$$

for d. The residual r_0 is constructed by subtracting the components of previous basis functions from the current basis function

$$r_0 = \phi_q - \Phi' d \tag{8.2}$$

The q^{th} interpolation index is the position of the peak of the residual. Then we update Φ' to $[\Phi', \phi_q]$ and P' to $[P', \boldsymbol{e}_{p_i}]$ and repeat the process for the next $q = 3$. This process is repeated until all the interpolation indices are determined and the final P' is obtained. The final P' is called P_x and P_f in next section; the subscripts show that the interpolation indices are different for x and f. Note that the basis functions ϕ_i have been sorted so that the corresponding eigenvalues in

the covariance matrix (Eq. 2.22) are in descending order. The index position p_i is nonrepeated [47], i.e. all index positions are different.

8.1.3 State and Dynamics Estimation

A static observer is designed for the system. By using the basis functions Φ and their corresponding interpolation indices, an output to state mapping that is similar to Eq. 4.3 can be obtained

$$x \approx \Phi(P_x^T \Phi)^{-1} X \tag{8.3}$$

where $X = [y_1 \; y_2 \; \cdots \; y_{k_s}]^T$ are outputs measured by state sensors, and P_x contains the interpolation indices. The state in the ROM can be obtained by multiplying x by Φ:

$$c \approx \tilde{c} := (P_x^T \Phi)^{-1} X \tag{8.4}$$

Similarly, f can be estimated by using the same approach:

$$f \approx \tilde{f} := \Psi(P_f^T \Psi)^{-1} Y \tag{8.5}$$

where P_f is obtained based on Ψ and

$$Y = [y_{1+k_s} - b(z_{1+k_s})u \quad y_{2+k_s} - b(z_{2+k_s})u \; \cdots \; y_{k_s+k_v} - b(z_{k_s+k_v})u]^T \tag{8.6}$$

Remark 12 *Although Eq. 8.5 provides an estimate of the nonlinear term of the governing equation, static observer is used instead of Luenberger-type dynamic observer despite the fact that dynamic observer can dramatically relax the requirement on the number of sensors. The first reason is the estimate of f may not be accurate enough to estimate the state since f is unknown. The second reason is the observer gain is not easy to determine when the governing law is not available.*

8.1.4 Controller Based on Feedback Linearization

The controller constitutes of 2 parts: the first part u_1 is designed using feedback linearization and the second part comes from the estimation of the error of the nonlinearity observer (Eq. 8.5). For a given set point x_0, the corresponding set

point c_0 in the ROM is calculated by

$$c_0 = \Phi^T x_0 \tag{8.7}$$

We define the distance $e \in \mathbb{R}^n$ from the set point as

$$e = c - c_0 \tag{8.8}$$

u_1 is designed so that

$$\dot{e} + K_d e = 0 \tag{8.9}$$

where $K_d \in \mathbb{R}^{n_a \times n_a}$ is a diagonal matrix with positive entries and n_a is the number of actuators.

In discretized context, we express the estimated ROM in the following form:

$$\dot{\tilde{c}} = \Phi^T \tilde{f} + B u_1 \tag{8.10}$$

We can obtain u_1 from

$$u_1 = B^{-1}\Big(- K_d(c - c_0) - \Phi^T \tilde{f}\Big) \tag{8.11}$$

With the estimated state (Eq.8.4) and f (Eq. 8.5), the implementation of u_1 doesn't require the explicit information of the governing equation.

8.2 Limitations of Equation-free Method

In previous work [91], a predetermined term is assumed to be available so that a steady state is translated to the operation point when the manipulated input is 0. This term not only reduces the number of actuators required to regulate the system, but also relaxes the requirement on the accuracy of the estimation. Hence, the number of sensors used for estimation can be reduced. However, this transformation method may not always be feasible, especially when the governing law of the system is unknown. Motivated by this limitation, we try to regulate the system without translating the steady state. In this situation, a more accurate estimation of the term f is required, otherwise the error will lead to off set (which will be demonstrated in the numerical results). Since the accuracy of the estimation

hinges on the snapshot ensemble, one way to solve the problem is to enrich the snapshot ensemble by using various initial conditions and operating conditions. However, there is no guarantee that the enriched snapshots can improve the accuracy. Here we propose another approach to reduce the estimation error of f by using the information of x.

8.3 Modified Equation-free Method

In previous work [91], a predetermined term is assumed to be available so that a steady state is translated to the operation point when the manipulated input is 0. This term not only reduces the number of actuators required to regulate the system, but also relaxes the requirement on the accuracy of the estimation. Hence, the number of sensors used for estimation can be reduced. However, this transformation method may not always be feasible, especially when the governing law of the system is unknown. Motivated by this limitation, we try to regulate the system without translating the steady state. In this situation, a more accurate estimation of the term f is required, otherwise the error will lead to off set (which will be demonstrated in the numerical results). Since the accuracy of the estimation hinges on the snapshot ensemble, one way to solve the problem is to enrich the snapshot ensemble by using various initial conditions and operating conditions. However, there is no guarantee that the enriched snapshots can improve the accuracy. Here we propose another approach to reduce the estimation error of f by using the information of x.

8.3.1 Motivation

By properly choosing the sensor locations via DEIM, the state x can be estimated by using the information of a restricted number of sensors with enough accuracy; however, the estimation for f usually cannot achieve the same level of accuracy. This is because f does not necessarily satisfy the state space partition assumption that x does. This can be demonstrated in a simple example as follows:

We collect 40 snapshots of x and 40 snapshots of f from the same open loop process (Fig. 8.2) of a diffusion reaction process. We apply POD to each snapshot ensemble. The energy captured by first n basis functions for x and f are compared

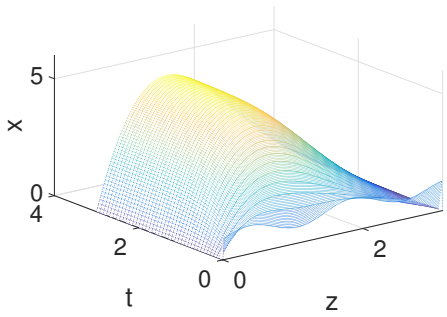


Figure 8.2: system response of the open-loop process.

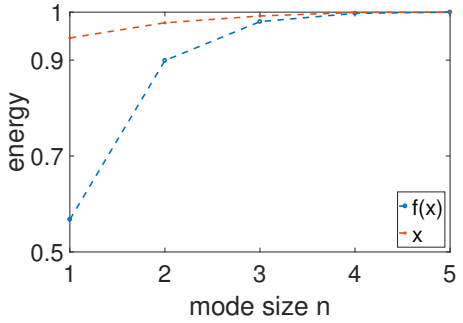


Figure 8.3: energy captured by first n basis functions.

in Fig. 8.3. It can be seen from this result that more basis functions are required for f than x to capture a certain amount of energy. Motivated by this observation, we will use the information of x to correct the estimate of f so that the total number of sensors can be reduced.

8.3.2 Updating Dynamics Estimation

In the modified equation free method, we use a correction term to improve the accuracy of the estimation for f. A correction term $\Delta f \in \mathbb{R}^{n_a}$ for $\Phi^T f$ in Eq. 8.10 is defined as

$$\Delta f = \Phi^T f - \Phi^T \tilde{f} \tag{8.12}$$

Unfortunately, Δf cannot be evaluated. We estimate Δf by $\widetilde{\Delta f}$ that is updated every Δt units of time at $t = t_0, t_1, \cdots, t_v, t_{v+1}, \cdots$, where $t_v = v\Delta t$.

The estimated ROM becomes:

$$\dot{\tilde{c}} = \Phi^T \tilde{f} + \widetilde{\Delta f} + Bu \tag{8.13}$$

Assuming $\widetilde{\Delta f}$ is constant in the time interval $[t_v, t_{v+1}]$ and $\widetilde{\Delta f} = \widetilde{\Delta f}_v$, the control action in that interval is adjusted accordingly:

$$u = B^{-1}\Big(- K_d(c - c_0) - \Phi^T \tilde{f} - \widetilde{\Delta f}_v\Big) \tag{8.14}$$

The state estimated by the observer at $t = t_v$ is denoted by $\tilde{c}_v$. If the correction term $\widetilde{\Delta f}_v$ is accurate, the state during $[t_v, t_{v+1}]$ should obey the rule determined

by Eq. 8.9. Therefore,

$$\begin{aligned} \tilde{e}_{v+1,1} &= \tilde{e}_{v,1}\mathrm{e}^{-K_{d,1}\Delta t} \\ \tilde{e}_{v+1,2} &= \tilde{e}_{v,2}\mathrm{e}^{-K_{d,2}\Delta t} \\ &\vdots \end{aligned} \tag{8.15}$$

where $\tilde{e}_{i,j}$ indicates the difference between j^{th} estimated state variable in the ROM at $t = t_i$ and the objective and $K_{d,i}$ denotes the i^{th} entry on the diagonal of K_d. The difference between the "actual" estimated distance $\tilde{e}_{v+1}$ and the "theoretical" distance in Eq. 8.15 indicates the error of the correction term $\widetilde{\Delta f}$.

At $t = t_0$, $\widetilde{\Delta f}$ is chosen to be 0, at $t = t_{v+1}$, $\widetilde{\Delta f}$ is updated using the estimated distance $\tilde{e}$ at $t = t_v$ and $t = t_{v+1}$. Take the first state variable in the ROM as an example, the dynamics during $[t_v, t_{v+1}]$ is

$$\dot{\tilde{c}}_1 = \phi_1^T \tilde{f} + \widetilde{\Delta f}_{v,1} + e_{\Delta,1} + B_1 u \tag{8.16}$$

With the control action in Eq. 8.14, we can obtain

$$\dot{\tilde{e}_1} = -K_{d,1}\tilde{e_1} + e_{\Delta,1} \tag{8.17}$$

Assumption 9 *We assume the variance of the error e_Δ is negligible during each time interval $[t_v, t_{v+1}]$, and e_Δ can be approximated by a constant.*

Based on assumption 9, it can be derived that

$$e_{\Delta,1} = \frac{-\mathrm{e}^{-K_{d,1}\Delta t}K_{d,1}\widetilde{e_{v,1}} + K_{d,1}\widetilde{e_{v+1,1}}}{1 - \mathrm{e}^{-K_{d,1}\Delta t}} \tag{8.18}$$

The errors for other state variables are calculated using the same approach. Then $\widetilde{\Delta f}$ is updated to

$$\widetilde{\Delta f}_{v+1} = \widetilde{\Delta f}_v + e_\Delta \tag{8.19}$$

Remark 13 *The information used to correct the estimation of f is from the sensor network used to estimate x. Hence, no additional sensors are required in this updating part.*

Remark 14 *Since the estimation of the system dynamics in this equation-free method is associated with POD, the accuracy of the estimation hinges on the quality*

Table 8.1: algorithm of equation-free method

	offline	
1	collect snapshots of x and f	
2	generate basis functions for x and f using POD	Eq. 2.24
3	determine sensor location by applying DEIM to basis functions	
4	calculate the matrices for output-state mapping and output-dynamics mapping	Eq. 8.4-8.5
5	design controller	
6	set $\widetilde{\Delta f}_0 = 0$	
	online	
6	continuous measurement of state sensors $\rightarrow x$	Eq. 8.4
7	continuous measurement of velocity sensors $\rightarrow f$	Eq. 8.5
8	determine control action u	Eq. 8.14
9	update correction term $\widetilde{\Delta f}$ for $\Phi^T f$ using estimated x	Eq. 8.18-8.19

of the basis function which depends on how the snapshot ensemble is collected. As a result, the number of sensors for both x and f can be reduced with a better collected snapshot ensemble or more snapshots to achieve the same level of accuracy.

The modified equation-free control method is summarized in Table 8.1.

Remark 15 (error bound) *In previous work [91], the error bound for f is proved to be*

$$\|\tilde{f} - f\|_2 \leq (1 + \sqrt{2M})^{k_v - 1} \|\psi_1\|_\infty^{-1} \|f - \Psi\Psi^T f)\|_2$$

where $\|f - \Psi\Psi^T f\|$ is small based on the assumption 6 that implies the nonlinear function g can be represented by basis functions Ψ. The updating step in the proposed method relaxes this assumption.

Remark 16 *In this work, we try to regulate a process of which model information is not available, control objective is known, and offset is inevitable (all the actuators need be to used to achieve the best regulating performance). Plantwide control [96] can be used by combining the proposed control method with a system identification algorithm to gather information about the system and thus achieve better performance in terms of economic benefit, production efficiency and safety.*

8.3.3 Original Equation-free Method

In this section, we regulate the diffusion reaction process via the original equation-free method without shifting the steady state. The snapshots used for generating basis functions are collected from an open-loop process with the following initial condition:

$$x(t_0) = 0.3\sin(4.3z) - 0.11\cos(1.38z) + 1 \tag{8.20}$$

40 snapshots are collected and displayed in Fig. 8.2.

Since x can be accurately estimated with a small number of sensors, we use 5 state sensors (more than 99.99% is captured) to estimate the state and vary the number of velocity sensors. It is found that at least 23 velocity sensors are required. The system response in systems with 21, 23, and 30 velocity sensors are displayed in Fig. 8.4a, 8.4b and 8.4c, respectively. To better compare the results, we define the distance from the set point as:

$$d = \sqrt{\int (x - x_o)^2 \mathrm{d}z} \tag{8.21}$$

The distance from the set point in Fig. 8.4a, 8.4b and 8.4c are compared in Fig.8.6. It can be seen that as the number of velocity sensors increase, the system response approaches to the set point. In this case, as the total number of sensors increases, the error of the observer (for the state) also decreases (Fig. 8.7). Fig. 8.5a, 8.5b and 8.5c give the corresponding control actions. While none of them has chattering, the larger error that results from decreasing the number of sensors leads to actuator saturation.

8.3.4 Original Equation-free Method with Enriched Snapshots Ensemble

To show that a better collected set of snapshots can increase the quality of the estimation, in this section, we use 2 different initial conditions to collect offline snapshots. The first initial condition is the same with the last section (Eq. 8.20), the second initial condition is Eq. 8.22.

$$x(t_0) = \cos(z) \tag{8.22}$$

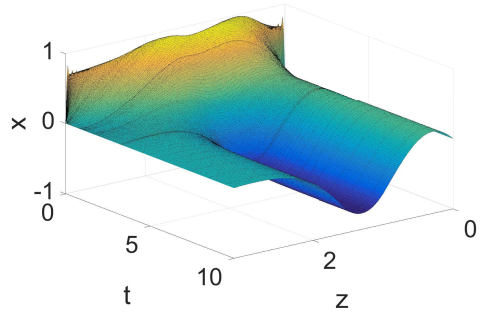


Figure 8.4a: system response of the closed-loop process with 5 plus 21 sensors.

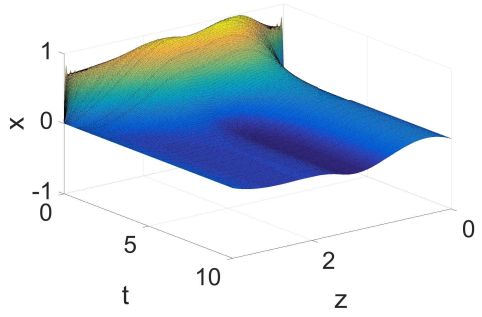


Figure 8.4b: system response of the closed-loop process with 5 plus 23 sensors.

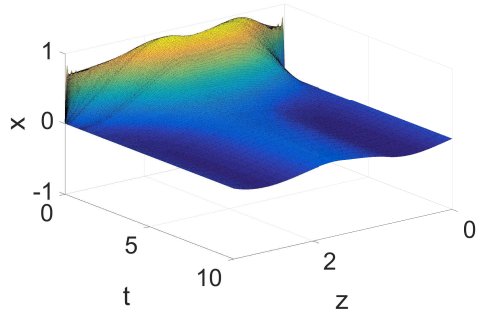


Figure 8.4c: system response of the closed-loop process with 5 plus 30 sensors.

80 snapshots are collected, 40 from the open-loop process with the first initial condition and the other 40 from the second one. We still use 5 state sensors; yet the number of velocity sensors can be decreased from 23 to 12. The conclusion can be drawn from the system response from the closed-loop systems with 10, 12 and 15 velocity sensors (Fig. 8.8a, 8.8b and 8.8c) and the distance from the set point (Fig. 8.9).

8.3.5 Modified Equation-free Method

In this section, we use the correction term to improve the quality of the estimation. The value of the correction term is updated every 0.5 units of time. The same snapshot ensemble as section "Original Equation-free Method" is used. The simulation results show that the number of velocity sensors can be decreased to 6

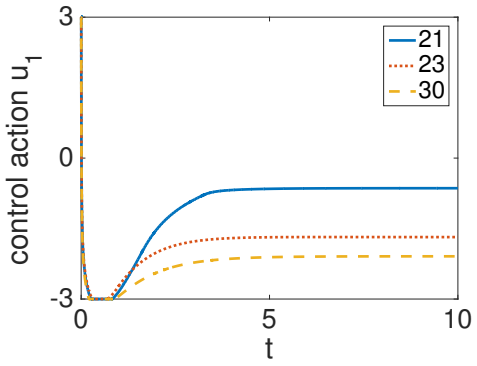


Figure 8.5a: comparison of the control action of the 1st actuator.

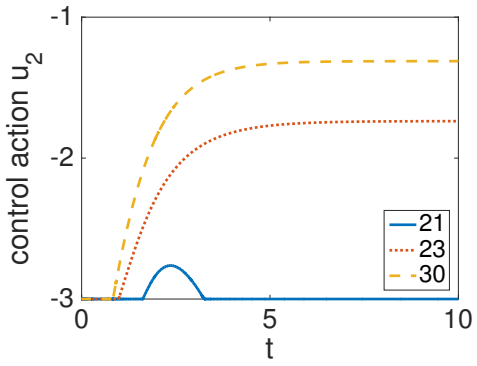


Figure 8.5b: comparison of the control action of the 2nd actuator.

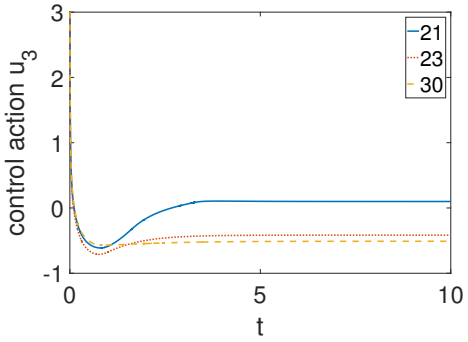


Figure 8.5c: comparison of the control action of the 3rd actuator.

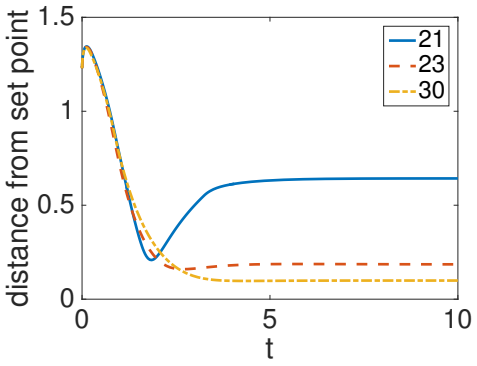


Figure 8.6: temporal profile of the distance of the state from the set point.

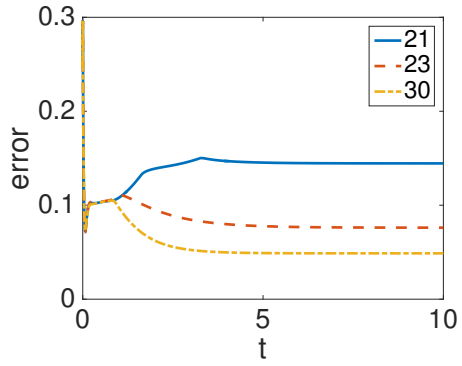


Figure 8.7: comparison of the error of the observer.

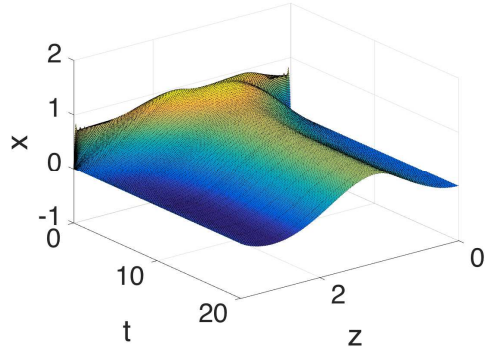


Figure 8.8a: system response of the closed-loop process with 5 plus 10 sensors.

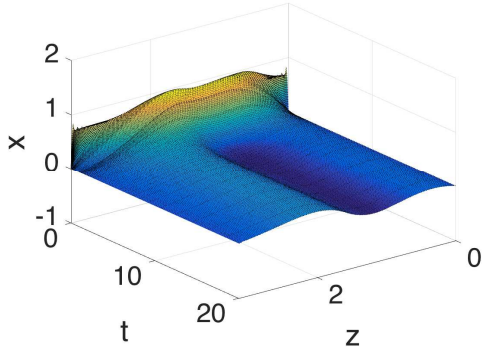


Figure 8.8b: system response of the closed-loop process with 5 plus 12 sensors.

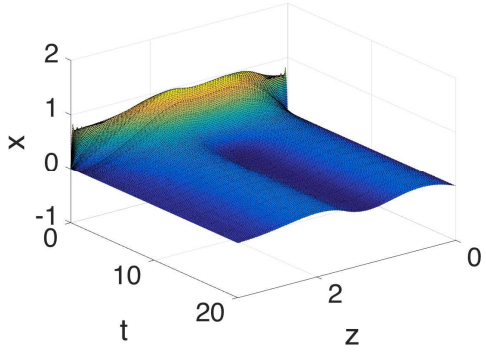


Figure 8.8c: system response of the closed-loop process with 5 plus 15 sensors.

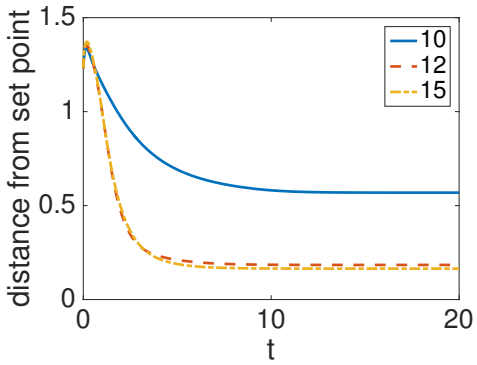


Figure 8.9: temporal profile of the distance of the state from the set point.

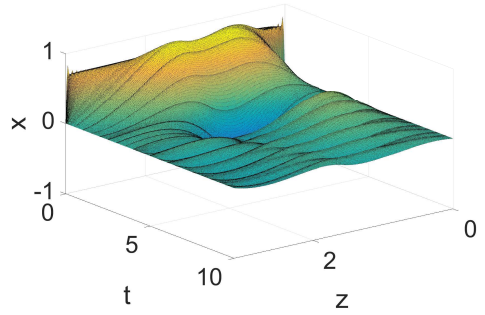


Figure 8.10a: system response of the closed-loop process with 5 plus 5 sensors.

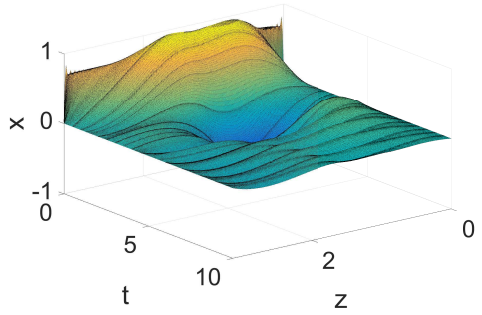


Figure 8.10b: system response of the closed-loop process with 5 plus 6 sensors.

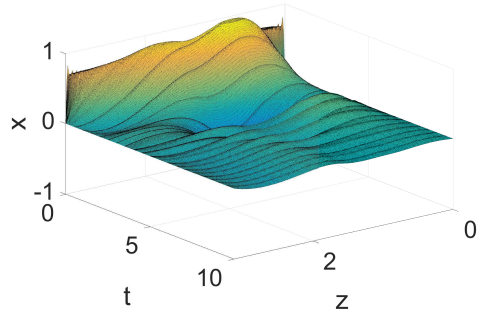


Figure 8.10c: system response of the closed-loop process with 5 plus 8 sensors.

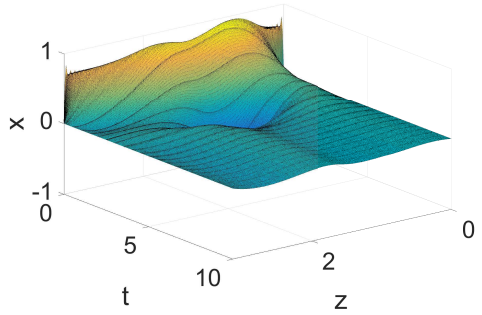


Figure 8.10d: system response of the closed-loop process with 5 plus 10 sensors.

in this example. Fig. 8.10a, 8.10b, 8.10c and 8.10d illustrate the system response in systems with 5, 6, 8, and 10 velocity sensors respectively. The temporal profile of the distance from the set point is depicted in Fig. 8.11 and the control actions are given in Fig. 8.12a, 8.12b and 8.12c. The overall performance of the controller is better than the one without the updating step. Note that when there are not enough sensors, the error in the correction term may lead to oscillation in the control actions.

8.3.6 sensor noise & model mismatch

In this section, we consider the sensor noise and model mismatch. We use the same 40 snapshots as section "Original Equation-free Method" and "Modified Equation-free Method". These snapshots are collected from a process in which β_T is

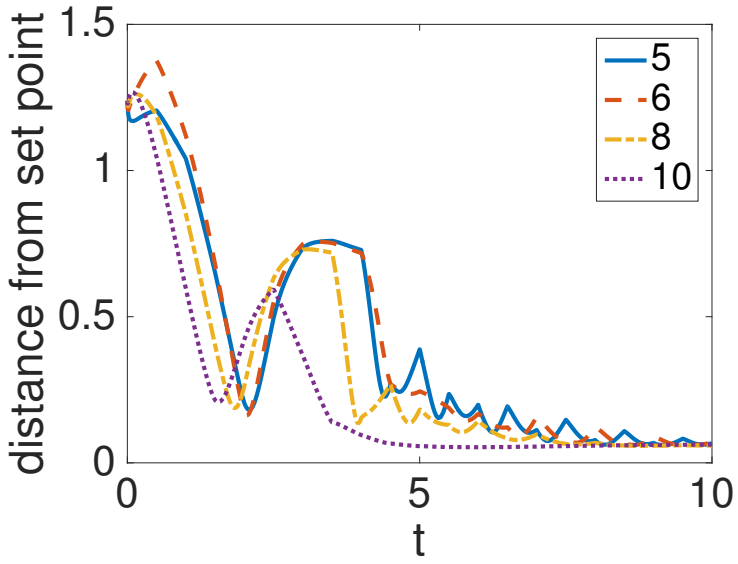


Figure 8.11: temporal profile of the distance of the state from the set point.

time-invariant (Eq. 3.30). We assume in the real process, β_T becomes time-variant. The spatial temporal profile of β_T is given in Fig. 8.13. We also assume 2 state sensors and 2 velocity sensors have noise, specifically $y_i = y_{real}(1 + \theta_i)$, where θ_i is random uniform noise of amplitude 0.3, 0.1, 0.2 and 0.3 for $i = 1, 2, 7, 8$. The noise of the first sensor is displayed in Fig. 8.14. We also use 5 state sensors as previous sections. Because of space limitation, we only present the result for 6 and 10 velocity sensors. Fig. 8.15a and 8.15b depict the system response. Although the state oscillates because of the time-varying coefficient and sensor noise, the controller still manages to regulate the system. The distance from the set point and the observer error are give in Fig. 8.16a, 8.16b, 8.17a and 8.17b. Both the distance and error are smaller in system with 10 velocity sensors. This is expected since the signal-to-noise ratio is higher with 10 sensors than 6 sensors.

8.3.7 Discussion

To further increase the quality of the modified equation-free method, there are at least 3 approaches we can investigate:

1. Enrich the snapshot ensemble

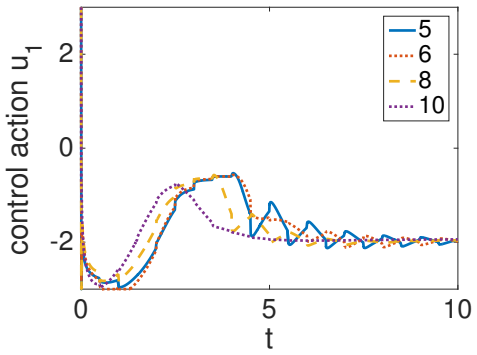


Figure 8.12a: comparison of the control action of the 1st actuator.

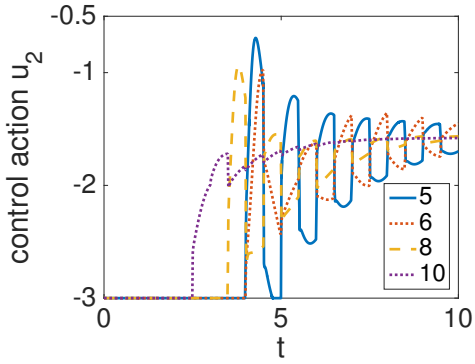


Figure 8.12b: comparison of the control action of the 2nd actuator.

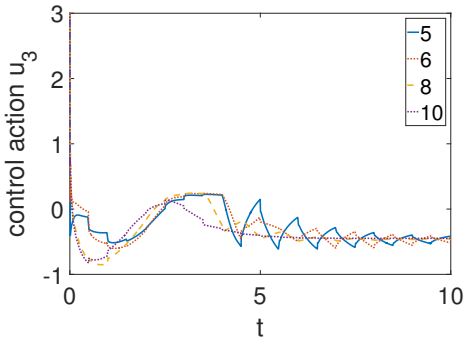


Figure 8.12c: comparison of the control action of the 3rd actuator.

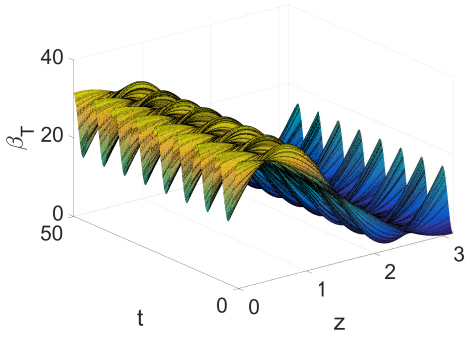


Figure 8.13: spatial temporal profile of β_T.

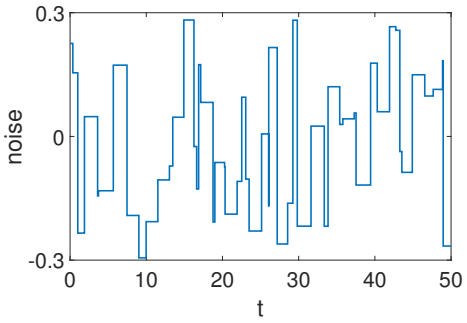


Figure 8.14: temporal profile of the sensor noise.

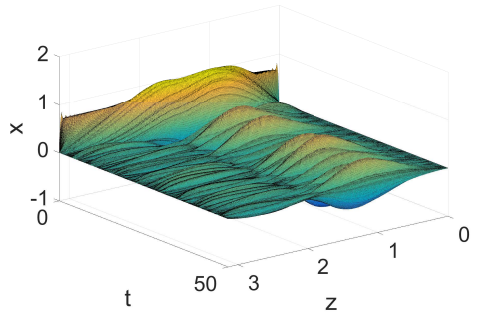


Figure 8.15a: system response of the closed-loop process with 5 plus 6 sensors in presence of sensor noise and model mismatch.

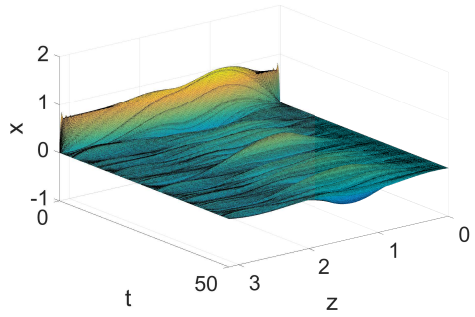


Figure 8.15b: system response of the closed-loop process with 5 plus 10 sensors in presence of sensor noise and model mismatch.

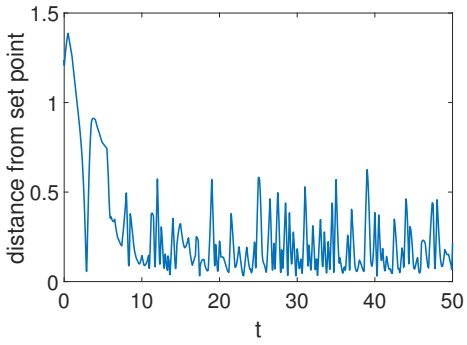


Figure 8.16a: distance of the state from the set point in system with 5 plus 6 sensors in presence of sensor noise and model mismatch.

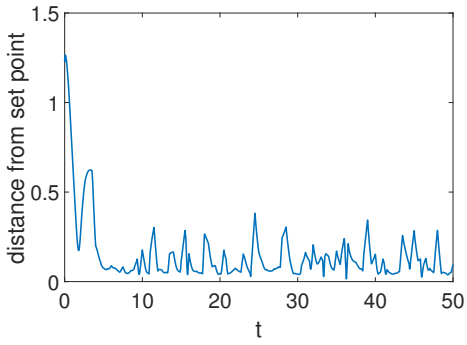


Figure 8.16b: distance of the state from the set point in system with 5 plus 10 sensors in presence of sensor noise and model mismatch.

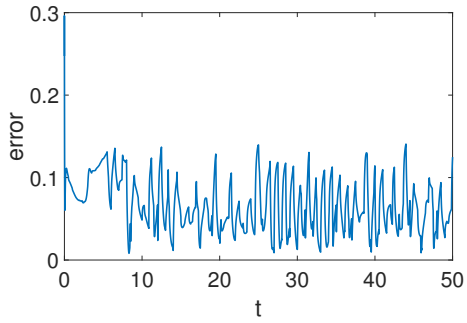


Figure 8.17a: error of observer in system with 5 plus 6 sensors in presence of sensor noise and model mismatch.

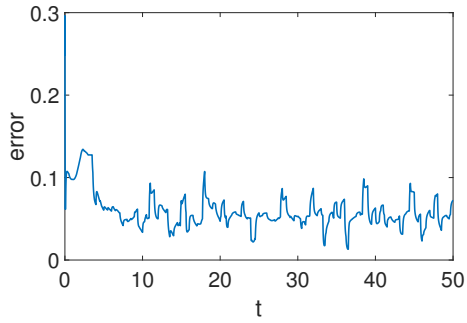


Figure 8.17b: error of observer in system with 5 plus 10 sensors in presence of sensor noise and model mismatch.

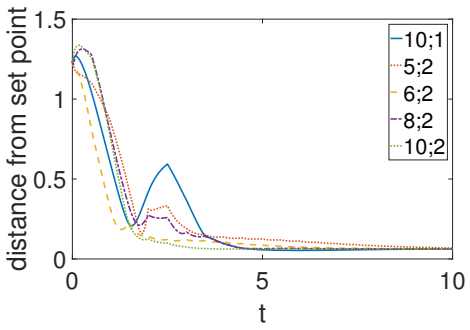


Figure 8.18: temporal profile of the distance of the state from the set point.

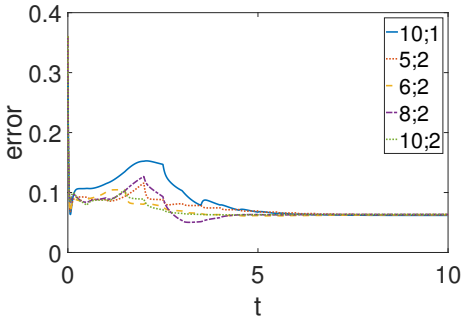


Figure 8.19: error of the observer.

2. Increase the frequency of updating the correction term

3. Change the timing of activating the correction term

8.3.7.1 snapshot ensemble

To show that a better collected set of snapshots can increase the quality of the estimation, we use 2 different initial conditions to collect off-line snapshots. The two initial conditions are the same with those in section "Original Equation-free Method with Enriched Snapshots Ensemble" (Eq. 8.20, Eq. 8.22). In total, 80 snapshots are collected. Results of systems using different number of velocity sensors and different snapshot ensembles are compared in Fig. 8.18 and 8.19. They are denoted by "$a; b$", in which a refers to the number of velocity sensors and b is the number of initial conditions. Fig. 8.18 illustrates the temporal profile of the distance from the set point. We can see that the oscillation in the system response is mitigated (the amplitudes of the peaks at t=2.5 decrease). The errors of the observers are compared in Fig. 8.19. The errors in systems with enhanced snapshot ensemble are significantly smaller. The error of the observer decreases and the system oscillates less as the number of velocity sensors increases except the case of 8 velocity sensors

Despite the difference between the standard POD and the proposed equation free method, we are optimistic that the research results on improved snapshot sampling, such as Smith's work [15] and Graham's work [90], might be extended to this equation free method.

8.3.7.2 frequency of updating

The second approach is to improve the accuracy of the correction term by varying the frequency of updating. Here we present the results for updating the correction term every 0.25 units of time. The results for updating every 0.5 units of time with 5 and 10 velocity sensors are denoted by "lf5" and "lf 10" in Fig. 8.21-8.20c, where "lf" means lower frequency. The distances from the set point are compared in Fig. 8.21 and Fig. 8.20a-8.20c depict the control actions. Similar to the first approach (section "snapshot ensemble"), increasing the updating frequency can also improve the accuracy of the estimation of the dynamics and mitigate the oscillation of the control actions. The effect of the frequency of updating is specially obvious with fewer sensors. However, this approach does not reduce much observer error.

Although the information used in the correction term updating are measured continuously, the correction term is not updated "continuously" due to the computational cost. On the other hand, the updating frequency should be high enough to satisfy assumption 9. A more advanced way to deal with it is to identify criteria to determine the updating frequency. The research about varying the snapshot acquisition frequency in adaptive proper orthogonal decomposition [97] might be useful.

8.3.7.3 activation timing for the correction term

The third approach is to activate the correction term after the transient period. The idea is the system in the fast stable subspace has not decayed to zero in the transient period. Hence, the large error in the ROM in the transient period will enter the correction term and may make the estimation worse before the correction term converges, which explains the peak in Fig. 8.11. In this section, the correction term is not activated until $t = 0.5$. The results in section 8.3.5 with 5 and 10 velocity sensors are denoted by "m5" and "m10" in Fig. 8.22-8.23c. Fig. 8.22 depicts the distance from the set point and Fig 8.23a-8.23c present the control actions. It can be observed that the performance of the controller is improved, especially when the number of velocity sensors is increased to 10, where the peak in Fig.8.22 disappears. The only exception is the control action in the system with 6 velocity sensors has a larger amplitude of oscillation. The "waiting time" should be long enough for the fast dynamics to relax, which depends on the ratio of the energy

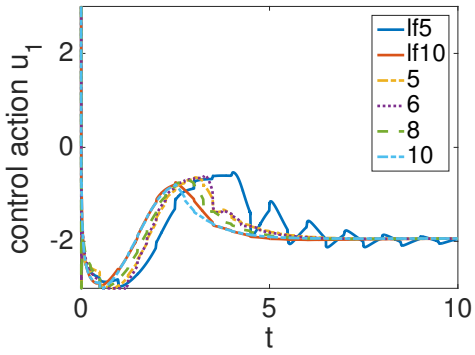


Figure 8.20a: comparison of the control action of the 1st actuator.

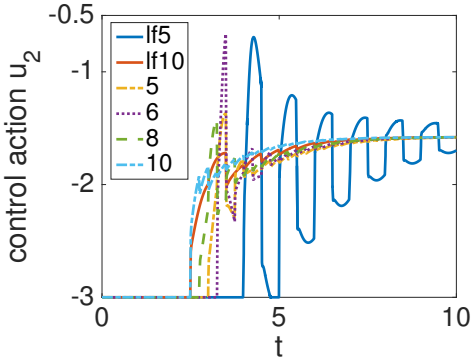


Figure 8.20b: comparison of the control action of the 2nd actuator.

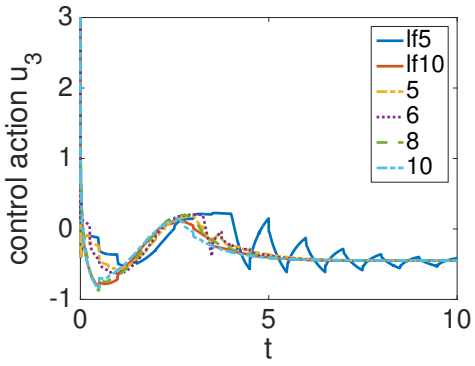


Figure 8.20c: comparison of the control action of the 3rd actuator.

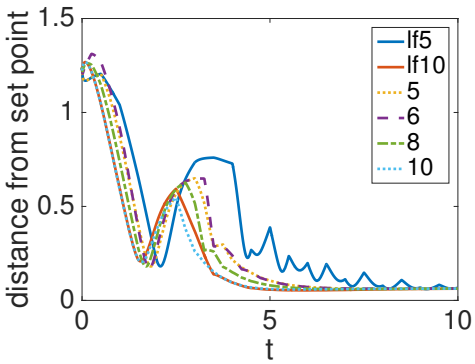


Figure 8.21: temporal profile of the distance of the state from the set point with different frequencies of updating.

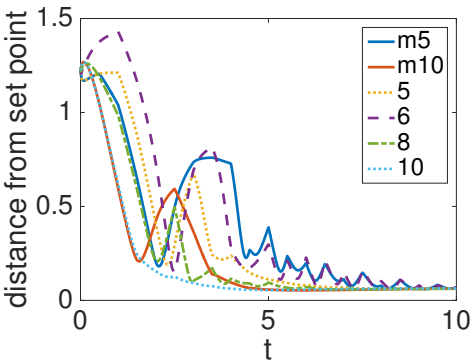


Figure 8.22: temporal profile of the distance of the state from the set point with different activation timing.

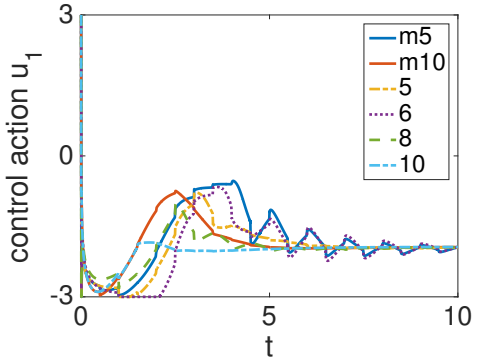


Figure 8.23a: comparison of the control action of the 1st actuator.

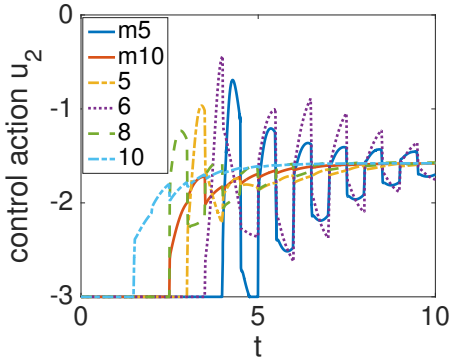


Figure 8.23b: comparison of the control action of the 2nd actuator.

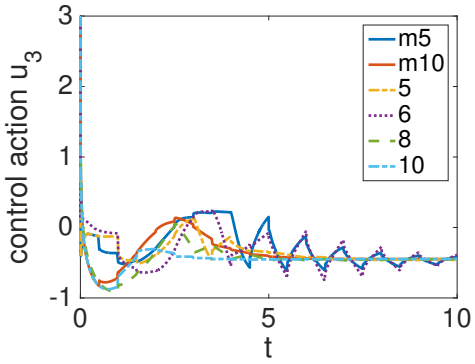


Figure 8.23c: comparison of the control action of the 3rd actuator.

captured by the basis functions in the ROM. On the other hand, this "waiting time" cannot be too long since the equation free method without the updating step is not able to stabilize the system with a small number of sensors during this waiting period.

To better present the improvement of above ideas, the amplitudes of the peak of the distance from the set point (Fig. 8.18, 8.21 and 8.22) are compared in table 8.2. It can be seen that except changing activation timing when using 6 sensors, all the 3 approaches either eliminates the oscillation or reduces the amplitude of the peak.

Table 8.2: the amplitude of the peak of the distance from the set point

		modified		further improvement			
number of sensors		5	10	5	6	8	10
mehtod	snapshots ensemble	0.760	0.594	0.333	0.276	N/A	N/A
	frequency of updating			0.65	0.648	0.627	0.546
	activation timing			0.677	0.815	0.497	N/A

Chapter 9
Conclusions and Future Research

This dissertation focuses on feedback control of distributed parameter system based on model reduction. Several methods are proposed to relax the assumptions/improve performance of existing methods. The performance of these methods are evaluated by simulation.

9.1 Conclusions

Chapter 1 introduces the background of our research and the general form of PDEs that describes the process we focus on. The objective of the research and the structure of the dissertation are also given.

In Chapter 2, the algorithm of MWR, Galerkin method, POD and APOD is introduced. These methods are the basis of our research.

In Chapter 3, we propose a modified POD method called passthrou POD. Using singular perturbation, we showed passthoru POD can decompose a distributed parameter system into slow subspace and fast one, which makes controller and observer designed based on slow part more robust than that using standard POD. To circumvent the difficulty of analytically solving eigenfunction problem of an operator, we didn't use the approach of solving eigenfunction of linearized operator. Instead, we construct standard POD basis functions first. Although standard POD basis functions can be affected by the number of snapshots and whether all modes are properly excited, we showed that snapshots have much less impact on passthrou POD than standard POD. Therefore, passthrou POD relaxes standard POD's requirements on snapshots ensemble. Besides, passthrou captures the dominant modes better than standard POD, which is demonstrated by the result

that controller and observer based on passthrou POD is more robust in the presence of disturbance.

In Chapter 4, we propose to use APOD/DEIM combination to reduce the computational cost in controlling dissipative distributed parameter system. By evaluating the performance of the proposed method in a diffusion-reaction process and Kuramoto-Sivashinsky equation, we conclude that using DEIM can reduce the computational cost and has negligible impact on the performance of the controller and observer compared with systems using APOD only. As the number of interpolation indices k increases, the error introduced by DEIM decreases, but the computational cost may increase.

In Chapter 5, a refined APOD called discrete adaptive proper orthogonal decomposition (DAPOD) is proposed. Compared with modified APOD, it provides a more accurate alternative to estimate the eigenvalues of the covariance matrix and eliminate the resulting basis size oscillation. The problem that the newest snapshot may be continually eliminated is also addressed. And the basis functions are guaranteed to be orthogonal.

In Chapter 6, the control problem in distributed parameter system with strong convective phenomena is investigated. Strong convective phenomena has a significant effect on the accuracy of the ROM. We use DAPOD to update the ROM so that the accuracy can be improved. The performance of the proposed strategy is demonstrated in a tubular reactor example.

In Chapter 7, we propose an equation-free control method based on DEIM. The requirement for knowledge of the governing equation in DEIM is circumvented so that it can be used for estimation. Only a finite number of point sensors are required in this method. The location of sensors are determined by linking the interpolation indices in DEIM to sensor locations. We demonstrate the performance of the proposed method by applying it to a diffusion-reaction process. The result shows that the observer can provide an estimation of the system profile with enough accuracy for controller design and it's robust in the presence of disturbance.

In Chapter 8, a modified version of equation-free control method is proposed and successfully applied to a diffusion reaction process. The assumption made in translating the steady state in the previous work is relaxed. The result illustrates that the method is robust in the presence of noise and model mismatch. Three approaches to further improve the performance of the proposed method are

discussed.

9.2 Future Research

The research work can be extended in the following directions.

- A major assumption of DAPOD/APOD is the snapshot is available periodically, which may not be feasible in many cases. If the state of a system can be scanned, the obtained information is a function of both time and space. A procedure is needed to incorporate this kind of information in updating of basis functions and the reduced order model.
- In Equation-free control method, we assume the state in all the locations in space can be measured and choose the best locations based on DEIM. A procedure is needed to determine the locations when some area cannot be measured.

Appendix A
Proof of Stability

To prove the stability of Eq. 6.3, first we prove the stability of the slow subsystem without involving the separation principle since the difference between the full order model and ROM is of order ϵ (section 6.2.2). Then we prove the stability of the hybrid system that results from the updating of ROM. The following Lyapunov function [46, 61] is considered

$$\mathcal{V}_t = \mathcal{V}_{\tilde{c}} + \mathcal{V}_o = \frac{\zeta_{\tilde{c}}}{2}\tilde{c}^T\tilde{c} + \frac{\zeta_o}{2}e^T e \tag{A.1}$$

where e denotes the error of the observer

$$e = c - \tilde{c} \tag{A.2}$$

With the static observer

$$y_m = \Phi_m c \rightarrow \Phi_m^T\Phi_m c = \Phi_m^T y_m \tag{A.3}$$

$$\tilde{c} = (\Phi_m^T\Phi_m)^{-1}\Phi_m^T y_m \tag{A.4}$$

So

$$\tilde{c} = c \tag{A.5}$$

Therefore

$$\frac{\mathrm{d}\mathcal{V}_t}{\mathrm{d}t} = \zeta_{\tilde{c}}\dot{\tilde{c}}^T\tilde{c} \tag{A.6}$$

A.1 feedback linearization

To prove the system under feedback linearization is stable, we use the Lemma 9.1 in [98]. We inject Eq. 5.30 into Eq. A.21 and 5.29, we obtain

$$\begin{aligned}\frac{\mathrm{d}c'}{\mathrm{d}t} &= -K_f c' \\ \frac{\mathrm{d}c_u}{\mathrm{d}t} &= -K_u c_u + F_u(c', c_u) + B_u B'^{-1}\Big[- K_f c' - F_l'(c) + F_n'(c)\Big]\end{aligned} \tag{A.7}$$

where K_u is positive definite. We consider $G_u = \begin{bmatrix} 0 \\ g_u \end{bmatrix} = \begin{bmatrix} 0 \\ F_u(c', c_u) + B_u B'^{-1}\Big[- K_f c' - F_l'(c) + F_n'(c)\Big] \end{bmatrix}$. To use the lemma, we first prove when $G_u = 0$, for some positve constants s_1, s_2, s_3, s_4, s_5

$$s_1\|c\|^2 \leq \mathcal{V}_t \leq s_2\|c\|^2 \tag{A.8}$$

$$\frac{\partial \mathcal{V}_t}{\partial t} + \frac{\partial \mathcal{V}_t}{\partial x} f_0 \leq -s_3\|c\|^2 \tag{A.9}$$

$$\|\frac{\partial \mathcal{V}_t}{\partial x}\| \leq s_4\|c\| \tag{A.10}$$

$$\|G_u\| \leq s_5\|c\| \tag{A.11}$$

where $f_0 = \begin{bmatrix} -K_f c' \\ -K_u c_u \end{bmatrix}$. It is obvious Eq. A.8 is satisfied. For Eq. A.9,

$$\frac{\partial \mathcal{V}_t}{\partial t} + \frac{\partial \mathcal{V}_t}{\partial x} f_0 = -\frac{\zeta_{\tilde{c}}}{2}\Big[c'^T K_f c' + c_u^T K_u c_u\Big] \leq -s_3\|c\|^2 \tag{A.12}$$

For Eq. A.10,

$$\|\frac{\partial \mathcal{V}_t}{\partial x}\| = \|\frac{\zeta_{\tilde{c}}}{2} c\| \leq s_4\|c\| \tag{A.13}$$

For Eq. A.11, based on assumption 1

$$\begin{aligned}\|G_u\| =& \|g_u\| \leq \text{ł}F_u(c', c_u)\| + \|B_u B'^{-1}\Big[- K_f c' - F_l'(c) + F_n'(c)\Big]\| \\ \leq& p_1\|c\| + p_2\|B_u B'^{-1}\| \cdot \|c\| \leq s_5\|c\|\end{aligned} \tag{A.14}$$

where p_1 and p_2 are positive constants.

Assumption 10

$$s_5 < \frac{s_3}{s_4} \tag{A.15}$$

Based on assumption 10, we can prove the system is stable. Note that if the unstable subsystem has slow dynamics, assumption 10 will be violated and the controller design has to consider both the stable subsystem and the unstable subsystem.

A.2 Lyanpunov based control

Here we prove the stability of the system using Lyanpunov based control

$$\begin{aligned}\frac{\mathrm{d}\mathcal{V}_t}{\mathrm{d}t} =& L_F\mathcal{V}(c) - \|(L_B\mathcal{V})^T(c)\|^2 \frac{L_F^*\mathcal{V}(c) + \sqrt{\left(L_F^*\mathcal{V}(c)\right)^2 + \left(u_{max}\|(L_B\mathcal{V})^T(c)\|\right)^4}}{\|(L_B\mathcal{V})^T(c)\|^2\left[1+\sqrt{1+\left(u_{max}\|(L_B\mathcal{V})^T(c)\|\right)^2}\right]} \\ =& \frac{-\rho_0\mathcal{V} + L_F\mathcal{V}\sqrt{1+\left(u_{max}\|(L_B\mathcal{V})^T(c)\|\right)^2} - |L_F^*\mathcal{V}|\sqrt{1+\left(\frac{u_{max}\|(L_B\mathcal{V}\|}{L_F^*\mathcal{V}}\right)^2(u_{max}\|(L_B\mathcal{V}\|)^2}}{1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}}\end{aligned} \tag{A.16}$$

if $L_F\mathcal{V} < 0$, $\frac{\mathrm{d}\mathcal{V}_t}{\mathrm{d}t} < 0$, otherwise, based on the assumption $L_F^*\mathcal{V}(c) \leq u_{max}\|(L_B\mathcal{V})^T(c)\|$

$$\frac{\mathrm{d}\mathcal{V}_t}{\mathrm{d}t} \leq \frac{-\rho_0\mathcal{V} - \rho_0\mathcal{V}\sqrt{1+\left(u_{max}\|(L_B\mathcal{V})^T(c)\|\right)^2}}{1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}} = -\rho_0\mathcal{V} \leq 0 \tag{A.17}$$

So the time derivative of the Lyapunov function is negative definite. Next we prove the input satisfies the constraint.

$$u = -\frac{L_B\mathcal{V}}{\|(L_B\mathcal{V})^T(c)\|^2}\frac{L_F^*\mathcal{V}(c) + \sqrt{\left(L_F^*\mathcal{V}(c)\right)^2 + (u_{max}\|(L_B\mathcal{V})^T(c)\|)^4}}{1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}} \tag{A.18}$$

$$
\begin{aligned}
\|u\| =& \frac{L_F^*\mathcal{V}(c) + \sqrt{\Big(L_F^*\mathcal{V}(c)\Big)^2 + (u_{max}\|(L_B\mathcal{V})^T(c)\|)^4}}{\|(L_B\mathcal{V})^T(c)\|\Big[1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}\Big]} \\
=& \frac{L_F^*\mathcal{V}(c) + u_{max}\|(L_B\mathcal{V})^T(c)\|\sqrt{\Big(\frac{L_F^*\mathcal{V}}{u_{max}\|(L_B\mathcal{V}\|}\Big)^2 + (u_{max}\|(L_B\mathcal{V}\|)^2}}{\|(L_B\mathcal{V})^T(c)\|\Big[1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}\Big]} \\
\leq& \frac{L_F^*\mathcal{V}(c) + u_{max}\|(L_B\mathcal{V})^T(c)\|\sqrt{1+(u_{max}\|(L_B\mathcal{V}\|)^2}}{\|(L_B\mathcal{V})^T(c)\|\Big[1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}\Big]} \\
\leq& u_{max}\frac{\frac{L_F^*\mathcal{V}(c)}{u_{max}\|(L_B\mathcal{V})^T(c)\|} + \sqrt{1+(u_{max}\|(L_B\mathcal{V}\|)^2}}{1+\sqrt{1+(u_{max}\|(L_B\mathcal{V})^T(c)\|)^2}} \leq u_{max}
\end{aligned}
\tag{A.19}
$$

Next we prove the stability for systems using dynamic observer. It has been proven in [61] that the time derivative of the following Lyapunov function is negative definite

$$
\mathcal{V}_o = \frac{\zeta_o}{2} e^T P_0 e \tag{A.20}
$$

So we only consider the time derivative of $\mathcal{V}_{\tilde{c}}$.

For feedback linearization, the estimated state variables $\tilde{c}$ satisfy

$$
\begin{aligned}
\frac{\mathrm{d}\tilde{c}'}{\mathrm{d}t} =& -K_f\tilde{c}' + G_m'(y_m - \tilde{y}) \\
\frac{\mathrm{d}\tilde{c}_u}{\mathrm{d}t} =& -K_u\tilde{c}_u + F_u(\tilde{c}', \tilde{c}_u) + B_u B'^{-1}\Big[-K_f\tilde{c}' - F_l'(c) + F_n'(\tilde{c})\Big] \\
&+ G_{um}(y_m - \tilde{y})
\end{aligned}
\tag{A.21}
$$

It is assumed that the Luenberger term $G_m(y_m - \tilde{y}) = \begin{bmatrix} G_m'(y_m - \tilde{y}) \\ G_{um}(y_m - \tilde{y}) \end{bmatrix}$ can be considered as a perturbation [61].

$$
G_m(y_m - \tilde{y}) = \sum_{i=1}^{s} w_i(\tilde{c}')\theta_i(t) \tag{A.22}
$$

By incorporating $G_m(y_m - \tilde{y})$ into G_u, we can use the same lemma to prove the stability.

The assumption can also be used in the proof for Lyapunov based control.

Finally, to prove the stability of switching system, ζ_o and ζ_c can be adjusted each time the hybrid system stability criteria is violated. [99]

Appendix B
Error Bound of Equation Free Control Method

The error bound of the estimate of f can be obtained by using the method provided in [47]. We denote the estimate of f by $\tilde{f}$

$$\tilde{f} = U'(P'^T U')^{-1} X' \tag{B.1}$$

When the error of the velocity sensor is negligible,

$$\tilde{f} = \underbrace{U'(P'^T U')^{-1} P'^T}_{\mathsf{P}} f = \mathsf{P} f \tag{B.2}$$

The projection of f in the subspace spanned by the basis functions U is considered

$$\hat{f} = U' U'^T f = \underbrace{U'(P'^T U')^{-1}(P'^T}_{\mathsf{P}} \underbrace{U')U'^T f}_{\hat{f}} = \mathsf{P}\hat{f} \tag{B.3}$$

We can obtain

$$\tilde{f} - f = \mathsf{P}f - \mathsf{P}\hat{f} + \hat{f} - f = (\mathsf{P} - I)(f - \hat{f}) \tag{B.4}$$

The selection of the sensor locations using DEIM can limit the bound of $(\mathsf{P} - I)$. It can be proved that (see [47])

$$\|\mathsf{P} - I\|_2 \leq (1 + \sqrt{2M})^{k-1} \|\tilde{\phi_1}\|_\infty^{-1} \tag{B.5}$$

and $(f - \hat{f})$ is negligible when the assumption 5 is satisfied.

Bibliography

[1] Pourkagar, D. and A. Armaou (2013) "Modification to adaptive model reduction for regulation of distributed parameter systems with fast transients," *AIChE Journal*, **59**(12), pp. 4595–4611.

[2] Armaou, A. and P. Christofides (1999) "Plasma enhanced chemical vapor deposition: Modeling and control," *Chemical engineering science*, **54**, pp. 3305–3314.

[3] Ye, Y. and K. Y. Choi (2008) "Dynamic modeling of a moving-packed-bed reactor for the solid-state polymerization of bisphenol-A polycarbonate," *Industrial and Engineering Chemistry Research*, **47**(10), pp. 3687–3699.

[4] Armaou, A., J. Baker, and P. D. Christofides (2001) "Feedback control of plasma etching reactors for improved etching uniformity," *Chemical engineering science*, **56**, pp. 257–265.

[5] Subramanian, S. and V. Balakotaiah (1996) "Classification of steady-state and dynamic behavior of distributed reactor models," *Chemical Engineering Science*, **51**(3), pp. 401–421.

[6] May, G. S. and C. J. Spanos (2006) *Fundamentals of Semiconductor Manufacturing and Process Control*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

[7] Balas, M. (1983) "The Galerkin method and feedback control of linear distributed parameter systems," *Journal of Mathematical Analysis and Applications*, pp. 527–546.

[8] Armaou, A. and P. Christofides (2001) "Robust control of parabolic PDE systems with time-dependent spatial domains," *Automatica*, **37**, pp. 61–69.

[9] Baker, J. and P. D. Christofides (1999) "Output Feedback Control of Parabolic PDE Systems with Nonlinear Spatial Differential Operators," *Industrial & Engineering Chemistry Research*, **38**(11), pp. 4372–4380.

[10] Balas, M. (1979) "Feedback control of linear diffusion processes," *International Journal of Control*, (January 2014), pp. 37–41.

[11] Christofides, P. and P. Daoutidis (1997) "Finite-dimensional control of parabolic PDE systems using approximate inertial manifolds," *Proceedings of the 36th IEEE Conference on Decision and Control*, **2**, pp. 1068–1073.

[12] Sirovich, L. (1987) "Turbulence and the dynamics of coherent structures .1. coherent structures," *Quarterly of Applied Mathematics*, **45**(3), pp. 561–571.

[13] Berkooz, G., P. Holmes, and J. Lumley (1993) "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid ...*, (1971).

[14] Choi, H., W.-P. Jeon, and J. Kim (2008) "Control of Flow Over a Bluff Body," *Annual Review of Fluid Mechanics*, **40**(1), pp. 113–139.

[15] Smith, T. R., J. Moehlis, and P. Holmes (2005) "Low-Dimensional Modelling of Turbulence Using the Proper Orthogonal Decomposition: A Tutorial," *Nonlinear Dynamics*, **41**(1-3), pp. 275–307.

[16] Bergmann, M. and L. Cordier (2008) "Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models," *Journal of Computational Physics*, **227**(16), pp. 7813–7840.

[17] Rowley, C. W., T. Colonius, and R. M. Murray (2004) "Model reduction for compressible flows using POD and Galerkin projection," *Physica D: Nonlinear Phenomena*, **189**(1-2), pp. 115–129.

[18] Ravindran, S. (2000) "A reduced-order approach for optimal control of fluids using proper orthogonal decomposition," *international journal for numerical methods in fluids*, (January), pp. 425–448.

[19] Scott Collis, S., R. D. Joslin, A. Seifert, and V. Theofilis (2004) "Issues in active flow control: theory, control, simulation, and experiment," *Progress in Aerospace Sciences*, **40**(4-5), pp. 237–289.

[20] Ilak, M. and C. W. Rowley (2008) "Modeling of transitional channel flow using balanced proper orthogonal decomposition," *Physics of Fluids*, **20**(3), p. 034103.

[21] Kerschen, G., J.-c. Golinval, A. F. Vakakis, and L. a. Bergman (2005) "The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview," *Nonlinear Dynamics*, **41**(1-3), pp. 147–169.

[22] Kerschen, G. and J. Golinval (2002) "Physical Interpretation of the Proper Orthogonal Modes Using the Singular Value Decomposition," *Journal of Sound and Vibration*, **249**(5), pp. 849–865.

[23] Lenaerts, V., G. Kerschen, and J.-C. Golinval (2003) "Identification of a continuous structure with a geometrical non-linearity. Part II: Proper orthogonal decomposition," *Journal of Sound and Vibration*, **262**(4), pp. 907–919.

[24] Druault, P., P. Guibert, and F. Alizon (2005) "Use of proper orthogonal decomposition for time interpolation from PIV data," *Experiments in Fluids*, **39**(6), pp. 1009–1023.

[25] Lucia, D. J., P. S. Beran, and W. a. Silva (2004) "Reduced-order modeling: new approaches for computational physics," *Progress in Aerospace Sciences*, **40**(1-2), pp. 51–117.

[26] Cao, Y., J. Zhu, I. Navon, and Z. Luo (2007) "A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition," *International Journal for ...*, (October 2006), pp. 1571–1583.

[27] Pitchaiah, S. and A. Armaou (2010) "Output feedback control of distributed parameter systems using adaptive proper orthogonal decomposition," *Industrial & Engineering Chemistry Research*, **49**, pp. 10496–10509.

[28] Tissot, G., L. Cordier, N. Benard, and B. Noack (2014) "Model reduction using Dynamic Mode Decomposition," *Comptes Rendus Mécanique*, **342**(6-7), pp. 410–416.

[29] Amsallem, D. and C. Farhat (2008) "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, **46**(7), pp. 1803–1813.

[30] Braconnier, T., M. Ferrier, J. C. Jouhaud, M. Montagnac, and P. Sagaut (2011) "Towards an adaptive POD/SVD surrogate model for aeronautic design," *Computers and Fluids*, **40**(1), pp. 195–209.

[31] Varshney, A., S. Pitchaiah, and A. Armaou (2009) "Feedback control of dissipative PDE systems using adaptive model reduction," *AIChE journal*, **55**(4).

[32] Solsvik, J. and H. Jakobsen (2012) "Solution of the pellet equation by use of the orthogonal collocation and least squares methods: Effects of different orthogonal Jacobi polynomials," *International Journal of Chemical Reactor Engineering*, **10**(1).

[33] Armaou, A. and P. Christofides (2002) "Dynamic optimization of dissipative PDE systems using nonlinear order reduction," *Chemical Engineering Science*, **57**, pp. 5083–5114.

[34] Volkwein, S. (2012) "Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling," *Lecture Notes, University of Konstanz*.

[35] Armaou, A. and A. Varshney (2004) "Dynamic optimization of dissipative PDEs using control vector parameterization: Application to GaN thin film epitaxy," *Proceeding of the 2004 American Control Conference*, (1), pp. 279–286.

[36] Liang, Y., H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu (2002) "Proper Orthogonal Decomposition and Its ApplicationsâĂŤPart I: Theory," *Journal of Sound and Vibration*, **252**(3), pp. 527–544.

[37] Hall, P., D. Marshall, and R. Martin (2002) "Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition," *Image and Vision Computing*, **20**(13-14), pp. 1009–1016.

[38] Singer, M. A. and W. H. Green (2009) "Using adaptive proper orthogonal decomposition to solve the reactionâĂŞdiffusion equation," *Applied Numerical Mathematics*, **59**(2), pp. 272–279.

[39] Xu, C., L. Luo, and E. Schuster (2010) "On recursive proper orthogonal decomposition via perturbation theory with applications to distributed sensing in cyber-physical systems," in *American Control Conference (ACC), 2010*, pp. 4905–4910.

[40] Xu, C. and E. Schuster (2013) "low-dimensional modeling of linear heat transfer systems using incremental the proper orthogonal decomposition method," *Asia-Pacific Journal of Chemical Engineering*, (8), pp. 473–482.

[41] Kokotovic, P. (1984) "Applications of singular perturbation techniques to control problems," *SIAM review*, **26**(4), pp. 501–550.

[42] Friedland, B. (2005) *Control system design: an introduction to state-space methods*, Dover Publications, Inc.

[43] Armaou, A., I. G. Kevrekidis, and C. Theodoropoulos (2004) "The gaptooth scheme, patch dynamics and equation-free controller design for distributed complex/multiscale processes," *Proceeding of the 2004 American Control Conference*, pp. 926–932.

[44] Ravindran, S. (2002) "Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition," *SIAM Journal on Scientific Computing*, **23**(6), pp. 1924–1942.

[45] Singer, M. a. and W. H. Green (2009) "Using adaptive proper orthogonal decomposition to solve the reaction-diffusion equation," *Applied Numerical Mathematics*, **59**(2), pp. 272–279.

[46] Pitchaiah, S. and A. Armaou (2013) "Output feedback control of dissipative PDE systems with partial sensor information based on adaptive model reduction," *AIChE Journal*, **59**(3), pp. 747–760.

[47] Chaturantabut, S. and D. Sorensen (2010) "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, **32**(5), pp. 2737–2764.

[48] Sivashinsky, G. I. and D. M. Michelson (1980) "On irregular wavy flow of a liquid film down a vertical plane," *Progress of Theoretical Physics*, **63**(6), pp. 2112–2114.

[49] Lee, C. H. and H. T. Tran (2005) "Reduced-order-based feedback control of the Kuramoto-Sivashinsky equation," *Journal of Computational and Applied Mathematics*, **173**(1), pp. 1–19.

[50] Sivashinsky, G. I. (1980) "on Flame Propagation Under Conditions of Stoichiometry," *Siam Journal on Applied Mathematics*, **39**(1), pp. 67–82.

[51] Sontag, E. D. (1989) "A universal construction of Artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, **13**(2), pp. 117–123.

[52] Mhaskar, P., N. H. El-Farra, and P. D. Christofides (2006) "Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control," *Systems & Control Letters*, **55**(8), pp. 650–659.

[53] Lin, Y. and E. D. Sontag (1991) "A universal formula for stabilization with bounded controls," *Systems & Control Letters*, **16**(6), pp. 393–397.

[54] Dubljevic, S., P. D. Christofides, and I. G. Kevrekidis (2004) "Distributed nonlinear control of diffusionâĂŞreaction processes," *International Journal of Robust and Nonlinear Control*, **14**(2), pp. 133–156.

[55] Christofides, P. and A. Armaou (2000) "Global stabilization of the KuramotoâĂŞSivashinsky equation via distributed output feedback control," *Systems & Control Letters*, **39**, pp. 283–294.

[56] Xu, C., Y. Ou, and E. Schuster (2011) "Sequential linear quadratic control of bilinear parabolic PDEs based on POD model reduction," *Automatica*, **47**(2), pp. 418–426.

[57] Bošković, D. M., a. Balogh, and M. Krstić (2003) "Backstepping in infinite dimension for a class of parabolic distributed parameter systems," *Mathematics of Control, Signals, and Systems (MCSS)*, **16**(1), pp. 44–75.

[58] Christofides, P. D. and P. Daoutidis (1996) "Feedback control of hyperbolic PDE systems," *AIChE Journal*, **42**(11), pp. 3063–3086.

[59] ——— (1996) "Robust control of hyperbolic PDE systems," *Chemical Engineering Science*, **53**(11), pp. 3063–3086.

[60] Antoniades, C. and P. Christofides (2001) "Studies on nonlinear dynamics and control of a tubular reactor with recycle," *Nonlinear Analysis: Theory, Methods & Applications*, **47**(9), pp. 5933–5944.

[61] Pourkargar, D. B. and A. Armaou (2014) "Design of APOD-based switching dynamic observers and output feedback control for a class of nonlinear distributed parameter systems," *Chemical Engineering Science*, **136**, pp. 62–75.

[62] Vernieres-Hassimi, L., D. Seguin, M. a. Abdelghani-Idrissi, and N. Mouhab (2015) "Estimation and Localization of Maximum Temperature in a Tubular Chemical Reactor By Luenberger State Observer," *Chemical Engineering Communications*, **202**(1), pp. 70–77.

[63] Pourkagar, D. B., A. Armaou, D. B. Pourkargar, and A. Armaou (2014) "Geometric output tracking of nonlinear distributed parameter systems via adaptive model reduction," *Chemical Engineering Science*, **116**, pp. 418–427.

[64] Schuëller, G. I. and H. a. Jensen (2008) "Computational methods in optimization considering uncertainties - An overview," *Computer Methods in Applied Mechanics and Engineering*, **198**(1), pp. 2–13.

[65] Fliess, M. and C. Join (2013) "Model-free control," *International Journal of Control*, **86**(12), pp. 2228–2252, 1305.7085.

[66] Hou, Z. S. and Z. Wang (2013) "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, **235**, pp. 3–35.

[67] Bontempi, G. and M. Birattari (2005) "From Linearization to Lazy Learning : A Survey of Divide-and-Conquer Techniques for Nonlinear Control (Invited Paper)," *International Journal*, **3**(1), pp. 56–73.

[68] Schaal, S. and C. G. Atkeson (1994) "Robot juggling: an implementation of memory-based learning," *Control Systems Magazine*, **14**(1), pp. 57–71.

[69] Overschee, P. V. and B. D. Moor (1994) "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems," *Automatica*, **311**(1), pp. 75–93.

[70] Qin, S. J. (2006) "An overview of subspace identification," *Computers & Chemical Engineering*, **30**(10-12), pp. 1502–1513.

[71] Chen, F.-c. (1990) "Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control," *IEEE Control Systems Magazine*, pp. 44–48.

[72] Bartecki, K. (2012) "PCA-based approximation of a class of distributed parameter systems: classical vs. neural network approach," *Bulletin of the polish academy of sciences, technical sciences*, **60**(3), pp. 651–660.

[73] Hu, J. and K. Kumamaru (1998) "A hybrid quasi-ARMAX modeling scheme for identification of nonlinear systems," in *Proceedings of the 35th Conference on Decision and Control*, December, pp. 1413–1418.

[74] Rachad, S., B. Nsiri, and B. Bensassi (2015) "System Identification of Inventory System Using ARX and ARMAX Models," *International Journal of Control and Automation*, **8**(12), pp. 283–294.

[75] Larimore, W. E. (2005) "Maximum likelihood subspace identification for linear, nonlinear, and closed-loop systems," *Proceedings of the 2005 American Control Conference 2005*, pp. 2305–2319.

[76] Verdult, V. and M. Verhaegen (2002) "Subspace identification of multivariable linear parameter-varying systems," *Automatica*, **38**(5), pp. 805–814.

[77] Weng, J.-H. and C.-H. Loh (2011) "Recursive subspace identification for on-line tracking of structural modal parameter," *Mechanical Systems and Signal Processing*, **25**(8), pp. 2923–2937.

[78] Favoreel, W., B. de Moor, and P. v. Overschee (1997) "Subspace Identification of Balanced Deterministic Bilinear Systems Subject to White Inputs," *Proc. European Control Conference, ECC 97, July 1-4, 1997, Brussels, Belgium*, **44**(6), pp. FR–A F2.

[79] Varshney, A. and A. Armaou (2008) "Low-order ODE approximations and receding horizon control of surface roughness during thin-film growth," *Chemical Engineering Science*, **63**(5), pp. 1246–1260.

[80] Pitchaiah, S. and A. Armaou (2008) "Online system-identification using subspace algorithms for the control of microscopic processes," *Proceedings of the American Control Conference*, (814), pp. 4413–4418.

[81] Hussain, M. A. (1999) "Review of the applications of neural networks in chemical process control - simulation and online implementation," *Artificial Intelligence in Engineering*, **13**(1), pp. 55–68.

[82] Pirdashti, M., S. Curteanu, M. H. Kamangar, M. H. Hassim, and M. A. Khatami (2013) "Artificial neural networks: applications in chemical engineering," *Reviews in Chemical Engineering*, **29**(4), pp. 205–239.

[83] Funahashi, K.-I. (1989) "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, **2**(3), pp. 183–192.

[84] Hosen, M. A., A. Khosravi, S. Nahavandi, and D. Creighton (2014) "Prediction interval-based neural network modelling of polystyrene polymerization reactor âĂŞ A new perspective of data-based modelling," *Chemical Engineering Research and Design*, **92**(11), pp. 2041–2051.

[85] Quan, H., D. Srinivasan, and A. Khosravi (2014) "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Transactions on Neural Networks and Learning Systems*, **25**(2), pp. 303–315.

[86] Wang, J. H., J. S. H. Tsai, J. S. Huang, S. M. Guo, and L. S. Shieh (2013) "A low-order active fault-tolerant state space self-tuner for the unknown sampled-data nonlinear singular system using OKID and modified ARMAX model-based system identification," *Applied Mathematical Modelling*, **37**(3), pp. 1242–1274.

[87] Hochman, A., B. N. Bond, and J. K. White (2011) "A stabilized discrete empirical interpolation method for model reduction of electrical, thermal, and microelectromechanical systems," *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 540–545.

[88] Peherstorfer, B., D. Butnaru, K. Willcox, and H.-j. Bungartz (2014) "Localized discrete empirical interpolation method," *Journal of scientific computing*, **36**(1), pp. 168–192.

[89] Peherstorfer, B. and K. Willcox (2015) "Online Adaptive Model Reduction for Nonlinear Systems via Low-Rank Updates," *SIAM Journal on Scientific Computing*, **37**(4), pp. 1–36.

[90] Graham, M. and I. Kevrekidis (1996) "Alternative approaches to the Karhunen-Loeve decomposition for model reduction and data analysis," *Computers & chemical engineering*, **20**(5), pp. 495–506.

[91] Yang, M. and A. Armaou "Synthesis of equation-free control systems for dissipative distributed parameter systems using proper orthogonal decomposition and discrete empirical interpolation method," in *Conference on Decision and Control*.

[92] Ramirez-castelan, C. E., J. Moguel-castañeda, and E. Puebla, Hector Hernandez-Martinez (2016) "A study of temperature sensor location based on fractal analysis for cascade control schemes in tubular reactors," *Chemical Engineering Science*, **141**, pp. 195–204.

[93] Demetriou, M. (2010) "Guidance of mobile actuator-plus-sensor networks for improved control and estimation of distributed parameter systems," *Automatic Control, IEEE Transactions on*, **55**(7), pp. 1570–1584.

[94] Demetriou, M. a. (2008) "Guidance of a moving collocated actuator/sensor for improved control of distributed parameter systems," *IEEE Conference on Decision and Control (CDC)*, **1**(1), pp. 215–220.

[95] UcinÌĄski, D. (2005) *Optimal measurement methods for distributed parameter system identification*, Systems and control series, CRC Press.

[96] Downs, J. J. and S. Skogestad (2011) "An industrial and academic perspective on plantwide control," *Annual Reviews in Control*, **35**(1), pp. 99–110.

[97] Pourkargar, D. B. and A. Armaou (2015) "APOD based control of linear distributed parameter systems under sensor controller communication bandwidth limitations," *AIChE journal*, **61**(2), pp. 434–447.

[98] Khalil, H. K. (2002) *Nonlinear systems*, Prentice Hall.

[99] Christofides, P. D. and N. H. El-Farra (2005) *Control of nonlinear and hybrid process systems : designs for uncertainty, constraints and time-delays*, Springer.

Vita
Manda Yang

EDUCATION

THE PENNSYLVANIA STATE UNIVERSITY | Ph.D. in Chemical Engineering
2013 - 2018 | University Park, PA • Cum. GPA: 3.87/4.0

THE PENNSYLVANIA STATE UNIVERSITY | Master in Chemical Engineering
2013 - 2015 | University Park, PA • Cum. GPA: 3.84/4.0

TSINGHUA UNIVERSITY | B.S. in Environmental Engineering
2009 - 2013 | Beijing, China • Cum. GPA: 89.7/100, Rank: 1/79

AWARDS AND HONORS

One of the 5 finalists of Nittany Watson Challenge
Apr 2017 | University Park, PA

Scholarship for excellence in academic performance
Oct 2012 | Beijing, China

Meishang International Scholarship of Tsinghua University
Oct 2011 | Beijing, China

Meishang International Scholarship of Tsinghua University
Oct 2010 | Beijing, China

SELECTED PUBLICATIONS

Manda Yang and Antonios Armaou, 2018, "Dissipative distributed parameter systems on-line reduction and control using DEIM/APOD combination", Proceedings of the American Control Conference

Manda Yang and Antonios Armaou, 2017, "Revisiting APOD accuracy for nonlinear control of transport reaction processes: a spatially discrete approach", Chemical Engineering Science

Manda Yang and Antonios Armaou, 2017, "Synthesis of equation-free control structures for dissipative distributed parameter systems using proper orthogonal decomposition and discrete empirical interpolation methods", Ind. Eng. Chem. Res

Manda Yang and Antonios Armaou, 2017, "On the design of equation-free controllers for dissipative PDEs via DEIM", Proceedings of the American Control Conference

M. Yang and A. Armaou, "Feedback control of semi-linear distributed parameter systems using advanced POD method," 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, 2015, pp. 4680-4687.