FEEDBACK CONTROL AND OPTIMIZATION USING ADAPTIVE
REDUCED ORDER MODELS

A Dissertation in
Chemical Engineering

by
Sivakumar Pitchaiah

© 2011 Sivakumar Pitchaiah

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2011
The dissertation of Sivakumar Pitchaiah 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

Costas Maranas
Professor of Chemical Engineering
Donald B. Broughton Professor of Chemical Engineering

Jack W. Langelaan
Assistant Professor of Aerospace Engineering

Andrew Zydney
Professor of Chemical Engineering
Walter L. Robb Chair and Head of the Department of Chemical Engineering

*Signatures are on file in the Graduate School.
ABSTRACT

The development of efficient control and optimization schemes for processes described by nonlinear parabolic partial differential (PDE) equations is a fundamental problem with a variety of industrially important applications. Typical examples of such processes range from reactive distillation in petroleum processing to plasma enhanced chemical vapor deposition, etching and metallorganic vapor phase epitaxy (MOVPE) in semiconductor manufacturing. The key difficulty in designing control schemes for such PDE systems arises from the “infinite-dimensional” nature of the distributed process model. Consequently, ideas for reducing the complexity of these models are considered through the formulation of the reduced order models (ROMs). A properly formulated ROM can replace the existing complex model thus greatly facilitating in the further design of control and optimization schemes for the above PDE processes.

One of the widely used methodologies for developing ROMs is the proper orthogonal decomposition (POD). Initially, one collects the experimental data or data from detailed numerical simulations (snapshots) in a “dataset”. POD then
extracts the characteristic basis (shape) functions from the collected data set. The computed basis functions are subsequently used in the framework of “method of weighted residuals” to compute the low-dimensional ROMs. The effectiveness of the POD methodology, however, is dependent on the quality of the collected dataset. Consequently, the ROMs are sufficiently accurate only in a restricted neighborhood around the state space where they are constructed. On the other hand, no well defined methodology exists for constructing ROMs with a larger region of validity as this requires a “representative” dataset which contains all the possible spatial modes (including those that might appear during closed-loop evolution of the PDE system). Thus there arises a need for the development of efficient methodologies for the systematic update of these ROMs during the closed-loop process evolution.

_Thus this doctoral thesis attempts to resolve fundamental computational issue associated with the formulation of the ROMs, specifically tailored for controller design, by developing a computationally efficient methodology called the adaptive proper orthogonal decomposition (APOD), that utilizes the closed-loop process information for ROM updates._ The updated ROMs are utilized for synthesis of high-performance nonlinear & optimal feedback controllers using Lyapunov and geometric control techniques. The effect of the developed methodology on the closed-loop system is analyzed by a study of stability and other performance properties of the closed-loop system (distributed PDE process model along with the controller). The developed adaptive data-reduction methodology would also have applications in various other fields such as efficient process monitoring and fault diagnosis of chemical processes.
<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Introduction</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1 Focus of the thesis</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1.1 Previous work &amp; Challenges</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1.2 Structure of thesis</td>
<td>9</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Adaptive proper orthogonal decomposition</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.1 Computation of initial basis function</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>2.2 Adaptive Proper Orthogonal Decomposition</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2.2.1 Initial basis Construction</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>2.2.2 Online basis refinement</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>2.2.2.1 Increasing the size of the basis</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>2.2.2.2 Decreasing the size of the basis</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>2.2.2.3 Maintaining the accuracy of the basis</td>
<td>27</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>State feedback control</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>3.1 Introduction</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>3.2 Mathematical Preliminaries</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>3.3 Problem formulation and solution methodology</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>3.3.1 Formulation of reduced order model</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>3.3.2 Controller design using feedback linearization</td>
<td>36</td>
</tr>
</tbody>
</table>
3.3.3 Recursive update of empirical basis functions and controller reconfiguration ......................... 38
3.4 Application to diffusion-reaction process ......................... 40
3.4.1 Linear spatial operator .................................. 40
3.4.2 Numerical results ........................................ 42
3.4.3 Spatially distributed actuation with nonlinear spatial operator and a spatially varying coefficient .................................................. 47
3.4.4 Numerical results ........................................ 49
3.4.5 Effect of parametric uncertainty .............................. 54

Chapter 4
Output feedback control 57
4.1 Introduction .................................................. 57
4.2 Mathematical Formulation ...................................... 58
4.3 Galerkin’s method .............................................. 61
4.4 Nonlinear static Output feedback control ......................... 63
4.4.1 Recursive update of empirical basis functions and controller reconfiguration ......................... 68
4.5 Applications .................................................. 68
4.5.1 Kuramoto-Sivashinsky equation ............................... 68
4.5.2 The FitzHugh-Nagumo Equation ............................... 85
4.6 Conclusions .................................................. 96

Chapter 5
Predictive control 98
5.1 Introduction .................................................. 98
5.2 Mathematical preliminaries ...................................... 99
5.2.1 Derivation of ROM ........................................... 101
5.3 Design of model predictive controllers .......................... 102
5.4 Applications .................................................. 104
5.4.1 Diffusion-reaction processes ................................ 104
5.4.2 Wave suppression .......................................... 109
5.5 Conclusions .................................................. 115

Chapter 6
Output Feedback control with partial sensor information 117
6.1 Introduction .................................................. 117
6.2 Mathematical Formulation ...................................... 118
6.3 Methodology .................................................. 122
6.3.1 Gappy reconstruction ...................................... 123
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.1.1</td>
<td>Complete off-line snapshots</td>
<td>124</td>
</tr>
<tr>
<td>6.3.1.2</td>
<td>Incomplete off-line snapshots</td>
<td>126</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Adaptive Proper Orthogonal Decomposition</td>
<td>128</td>
</tr>
<tr>
<td>6.3.2.1</td>
<td>Off-line: Initial basis Construction</td>
<td>130</td>
</tr>
<tr>
<td>6.3.2.2</td>
<td>Derivation of Reduced order model</td>
<td>131</td>
</tr>
<tr>
<td>6.3.2.3</td>
<td>Online recursive update of ROM</td>
<td>134</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Complete Closed-loop methodology</td>
<td>136</td>
</tr>
<tr>
<td>6.4</td>
<td>Nonlinear static Output feedback control</td>
<td>137</td>
</tr>
<tr>
<td>6.5</td>
<td>Kuramoto-Sivashinsky equation</td>
<td>139</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Case 1: Complete off-line snapshots</td>
<td>141</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Case 2: Partial off-line snapshots</td>
<td>149</td>
</tr>
<tr>
<td>6.6</td>
<td>Conclusions</td>
<td>154</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Robust control using extended Kalman filter</td>
<td>159</td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>159</td>
</tr>
<tr>
<td>7.2</td>
<td>Mathematical Preliminaries</td>
<td>160</td>
</tr>
<tr>
<td>7.3</td>
<td>Problem formulation and design of robust controller</td>
<td>162</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Derivation of reduced order model using Galerkin’s method</td>
<td>164</td>
</tr>
<tr>
<td>7.4</td>
<td>Extended Kalman filter</td>
<td>166</td>
</tr>
<tr>
<td>7.5</td>
<td>Design of Robust controller using EKF state estimates</td>
<td>169</td>
</tr>
<tr>
<td>7.6</td>
<td>Application to diffusion reaction process</td>
<td>170</td>
</tr>
<tr>
<td>7.6.1</td>
<td>Estimator implementation</td>
<td>172</td>
</tr>
<tr>
<td>7.6.2</td>
<td>Controller implementation</td>
<td>172</td>
</tr>
<tr>
<td>7.7</td>
<td>Conclusions</td>
<td>176</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Conclusions and Future work</td>
<td>180</td>
</tr>
<tr>
<td>8.1</td>
<td>Conclusions</td>
<td>180</td>
</tr>
<tr>
<td>8.2</td>
<td>Future work</td>
<td>185</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Proofs of Chapter 2</td>
<td>187</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Proofs of Chapter 6</td>
<td>190</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>195</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Hierarchy of process control and optimization in manufacturing. . . 2

2.1 Flow Chart of Adaptive Proper Orthogonal Decomposition methodology. ................................................. 24

3.1 Process operation block diagram under proposed controller design. . 34

3.2 Catalytic rod. ................................................................................................................. 40

3.3 a) Open loop profile of the state of the diffusion-reaction process with a linear spatial operator (Eq.3.14). b) Spatial profile of the dominant basis function obtained from the initial ensemble of the system of Eq. 3.14. ................................................................. 42

3.4 a) Closed-loop temperature profile of the system of Eq.3.14 using distributed actuation, \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \). b) Temporal profile of control action using distributed actuation for the system of Eq.3.14. c) Temporal profile of the dominant basis function for the system of Eq.3.14 when using distributed actuation. d) Temporal profile of the number of dominant empirical basis functions used in the reduced-order ODE model Eq.3.8 to capture the desired 99% energy of the ensemble. .............................................................................. 44

3.5 a) Closed-loop temperature profile of the system of Eq.3.14 using point actuation, \( b(z) = \delta(z - 0.4\pi) \). b) Temporal profile of control action using point actuation. c) Temporal profile of the first dominant basis function for the system of Eq.3.14 using a point actuator. d) Temporal profile of the second dominant basis function for the system of Eq.3.14. ........................................................................................................ 45

3.6 a) Number of dominant empirical basis functions used in the reduced-order model (Eq.3.8) for the system of Eq.3.14 as a function of time for point actuation, \( b(z) = \delta(z - 0.4\pi) \). b) Temporal profiles of eigenvalues of \( H \) and of the largest eigenvalue of \( c_q \) for the system of Eq.3.14 using a point actuator. .................................................. 46
3.7 Open loop profile of the state of the diffusion-reaction process with a nonlinear spatial operator (Eq.3.16). b) Spatial profile of the dominant basis function obtained from the initial ensemble of the system of Eq.3.16. ................................................................. 48

3.8 a) Closed-loop temperature profile of the system of Eq.3.16 using distributed actuation, \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \). b) Temporal profile of control action for the system Eq.3.16. c) Temporal profile of the first dominant basis function for the system of Eq.3.16. d) Temporal profile of the second dominant basis function for the system of Eq.3.16. ......................................................... 50

3.9 a) Temporal profile of number of dominant empirical basis functions used in the reduced-order model (Eq.3.8) for the system of Eq.3.16 (distributed actuation, \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \)). b) Temporal profiles of eigenvalues of \( H \) and of the largest eigenvalue of \( c_q \) for the system of Eq.3.16 using a spatially distributed actuator. 51

3.10 a) Closed-loop temperature profile of the system of Eq.3.16 for uniform initial condition of \( x_0 = 0.5 \) (distributed actuation, \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \)). b) Closed-loop temperature profile of the system of Eq.3.16 for a uniform initial condition of \( x_0 = -0.5 \) (distributed actuation) ................................................................. 51

3.11 a) Closed-loop temperature profile of the system of Eq.3.16 using point actuator, \( b(z) = \delta(z - 0.4\pi) \). b) Temporal profile of control action. c) Temporal profile of the first dominant basis function for the system of Eq.3.16 when using a point actuator. d) Temporal profile of the second dominant basis functions for the system described by Eq.3.16. ......................................................... 52

3.12 a) Temporal profile of number of dominant empirical basis functions used in the reduced-order model (Eq.3.8), to capture the desired 99% of energy of the ensemble for the system of Eq.3.16 (point actuation, \( b(z) = \delta(z - 0.4\pi) \)). b) Temporal profiles of eigenvalues of \( H \) and of the largest eigenvalue of \( c_q \). ......................................................... 53

3.13 a) Closed-loop temperature profile of the system of Eq.3.16 using point actuator \( b(z) = \delta(z - 0.4\pi) \) for 10% uncertainty in \( \beta_T \). b) Temporal profile of control action. c) Temporal profile of the first dominant basis function. d) Temporal profile of the second dominant basis function. ......................................................... 55

3.14 a) Temporal profile of number of basis functions of the system of Eq.3.16 using point actuator \( b(z) = \delta(z - 0.4\pi) \) for 10% uncertainty in \( \beta_T \). b) Temporal profiles of eigenvalues of \( H \) and of the largest eigenvalue of \( c_q \). ......................................................... 55
4.22 Temporal profile of CPU time for updating the ROM using APOD.

The time for computing the ROM using the standard POD, online, was $18 \times 10^{-3}$ s. Average acceleration obtained using APOD was 3 times compared to the standard POD approach. The time for computing the ROM using the standard POD was $18 \times 10^{-3}$ s. Average acceleration obtained using APOD was 3 times compared to the standard POD approach.

5.1 Open-loop profile of the state of the diffusion-reaction process (Eq. 3.16).

5.2 a) Closed-loop profile of the state of the diffusion-reaction process (Eq. 3.16). b) Temporal profile of the manipulated control action with $b(z) = H(z - 0.3\pi) - H(z - 0.6\pi)$.

5.3 a) Temporal profile of the MPC objective function. b) Temporal profile of the dimensionality of the ROM.

5.4 Temporal profile of the manipulated control action for nominal parameters and for $-20\%$ variation of the initial condition and for a $-12.5\%$ variation of $\beta_T$.

5.5 Open-loop profile of the state of the diffusion-reaction process (Eq. 5.12).

5.6 a) Closed-loop profile of the state of the diffusion-reaction process (Eq. 5.12). b) Temporal profile of the manipulated control action with 3 control actuators.

5.7 a) Temporal profile of the MPC objective function. b) Temporal profile of the dimensionality of the ROM.

6.1 Block diagram of closed-loop structure when using APOD.

6.2 Open-loop profile of the state of Eq. 4.15 with $\nu = 0.25$.

6.3 a) Closed-loop profile of the state of Eq. 4.15. b) Temporal profile of the control action, $u(t)$. ($\epsilon = 0.99$).

6.4 a) $L_2$ norm of the closed-loop profile of Eq. 4.15 with $\epsilon = 0.99$. b) Temporal profile of the control action, $u(t)$.

6.5 a) Temporal profile of the Lyapunov function, $V(t)$.

6.6 a) Temporal profile of the Lyapunov function, $V(t)$. b) Temporal profile of the dimensionality of the ROM.

6.7 Temporal profile of the Lyapunov function when using APOD. (\(\epsilon = 0.99\)).

6.8 Temporal profile of the state of the diffusion-reaction process with gappy reconstruction.

6.9 a) Contour plot of the evolution of KSE from $t = 0$ to $t = 2.5$ with 20% data missing per snapshot. b) Contour plot after 1st iteration with 20% data missing per snapshot. c) Contour plot after 1st iteration with 30% data missing per snapshot.
6.10 a) Closed-loop profile of the state of Eq. 4.15. b) Temporal profile of the control actuation, $u(t)$. ($\epsilon = 0.99$) ............................................. 155
6.11 a) $L_2$ norm of the closed-loop profile of Eq. 4.15. b) Temporal profile of the Lyapunov function, $V(t)$. ($\epsilon = 0.99$) ................................. 156
6.12 a) Temporal profile of the state reconstruction error using gappy reconstruction. b) Temporal profile of dimensionality of the local ROM of Eq. 4.15. ($\epsilon = 0.99$) ............................................. 157
6.13 Closed-loop process evolution for small value of $\epsilon = 0.9$ a) $L_2$ norm of the closed-loop profile of Eq. 4.15. b) Temporal of the Lyapunov function, $V(t)$ c) Temporal profile of dimensionality of the local ROM of Eq. 4.15. ............................................. 158

7.1 Process operation block diagram under proposed controller design. . 163
7.2 Temporal profile of the 2-norm of the error between true and estimated states and the trace of $P$ obtained during the open-loop operation of Eq.7.12 .................................................. 173
7.3 Open-loop profile of Eq.7.12 with measurement noise .................. 175
7.4 Estimated surface profile of Eq.7.12 in closed loop with controller of Eq.7.11 with $r \equiv 0$. ................................................................. 176
7.5 Closed-loop estimated surface profile of Eq.7.12 using robust controller of Eq.7.11 ................................................................. 177
7.6 Temporal profile of the 2-norm of the error between true and estimated states and the trace of $P$ obtained during the closed-loop operation of Eq.7.12 with the robust controller ......................... 178
7.7 Control action needed to stabilize Eq.7.12 using robust controller; with measurement noise. .................................................. 179
7.8 Control action needed to stabilize Eq.7.12 calculated using robust controller; no measurement noise. ........................................ 179
ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor Dr Antonios Armaou for his motivation, support and guidance throughout my research work. I am really indebted to him for encouraging and guiding me to become an independent researcher. His patience, enthusiasm for research and his friendly nature has left a deep impression on me.

Next I would like to thank Dr Borhan, Dr Maranas and Dr Langelaan for agreeing to serve on my committee. I learnt a lot interacting with them both during their courses and during my research presentations. I also would like to thank my lab mates Amit varshney, Gama and Samira for their help especially during the initial years of my Ph.D. program. Specially, I would like to express my gratitude to Amit for encouraging and helping me during my first year here at Penn state.

I really appreciate the patience, love and support given by my family members. Finally, I would like to express my heart felt gratitude to Ananda Caitanya prabhu, Bhakti Saranga prabhu and all Penn state vedic society members for making my stay at Penn state a meaningful one.
1.1 Focus of the thesis

Different decision levels exist in implementing process control and optimization in process control industries. This hierarchy is presented in figure 1.1: it starts with planning and scheduling, wherein production goals that meet various demand and logistic constraints are set. Planning & scheduling usually operates over relatively long time scales and tend to be decoupled from rest of the activities. In the next level, real time optimization (RTO) is utilized to coordinate with the network of process units and to provide optimal set points for each unit by solving a steady-state optimization. The online computation of these optimal set points allows the profits from the process to be maximized while satisfying operating
constraints. The set points obtained are subsequently utilized in advanced process control, where advanced nonlinear controllers are utilized to compute the dynamic moves necessary to achieve those set points. These dynamics moves are eventually implemented in the plant using the regulatory control framework.

![Figure 1.1. Hierarchy of process control and optimization in manufacturing.](image)

In this thesis, we especially focus on the advanced process control level. More
specifically, we concentrate on developing efficient control & optimization schemes for processes that can be mathematically described by dissipative nonlinear PDE systems. Such PDE systems arise naturally in the modeling of wide variety of industrially important processes such as transport-reaction and fluid flow systems (e.g. plasma-enhanced chemical vapor deposition, czochralski crystal growth and many other fluid-dynamical systems and biological systems). In the rest of this chapter we will present a description of main challenges that exist during the design of control and optimization schemes for these PDE processes and recent research results for transport reaction processes and fluid flow processes. Towards the end of this chapter we will present the objectives and structure of the present work.

1.1.1 Previous work & Challenges

Over the last twenty years significant research within the area of process control has focused on analysis and control of lumped chemical processes described by nonlinear ordinary differential equations (ODEs). Excellent reviews of results in the area of nonlinear process control can be found in [1, 2, 3, 4, 5, 6, 7, 8]. In the last decade research focus has also encompassed analysis and control of nonlinear distributed parameter systems. The research activity in this area has been motivated by a wealth of industrially important processes (e.g., MOVPE, plasma-enhanced chemical vapor deposition (PECVD), Czochralski crystal growth and various fluid dynamic systems) which exhibit significant spatial variations due to the presence
of strong diffusive and convective mechanisms. Without hoping to present a comprehensive overview of the excellent results in the area of control for distributed parameter systems, we focus on a brief discussion of results in the area of control of dissipative distributed parameter systems. Distributed chemical processes with significant diffusive phenomena can be described by systems of linear/nonlinear parabolic partial differential equations (PDEs). Parabolic PDEs are characterized by a finite-number of modes that describe their dominant long-term dynamic behavior [9]. As a result, the standard approach to control of parabolic PDEs utilizes modal decomposition techniques to obtain ODEs, which are then used for controller design [10, 11, 12, 13, 14, 15]. A drawback of this approach is the possibly high dimensionality of the resulting ODEs in order to capture the PDE dynamics accurately, thus leading to high dimensionality of the resulting controllers. Motivated by this, in [16] singular functions were employed for linear PDEs to derive ODE models used for controller synthesis. In another approach, the concept of approximate inertial manifolds [17, 18, 19] was employed, leading to low-order ODE models that accurately describe parabolic PDEs, that were subsequently used for nonlinear controller design [20, 21, 22, 23] and dynamic optimization [24].

Feedback controllers were also developed based on a combination of geometric and control Lyapunov function techniques [25, 26, 27] to compensate for the effect of model uncertainty and handle constraints on the control action [28]. The developed nonlinear control algorithms were successfully applied to a variety of
advanced materials processes [29, 25] and fluid dynamic systems [30, 31, 32]. The reader may refer to [33, 34, 35, 25] for reviews on control of nonlinear distributed parameter systems.

An important issue however that was not explicitly addressed by the above controller designs is the presence of constraints in the process operation. State & input constraints are widely prevalent in current industrial practice (relevant examples can be found in [36, 37, 38, 39]) and this motivates a need for controller design methodologies which explicitly address these process constraints. Model predictive control (MPC), also known as receding horizon control, is one such powerful tool for handling these process constraints with an optimal control setting. The control action in MPC is calculated by solving online a finite-horizon open-loop dynamic optimization problem (DOP) at each sampling time. As the process evolves, the DOP is recursively solved in order to provide further future control action. This recursion continues till the end of the process cycle. As the control action is computed online during process evolution, MPC has the capabilities to suppress the external disturbances and tolerate model inaccuracies (using feedback control) during the course of forcing the system to follow certain optimal path that respects the process constraints. Extensive reviews on various MPC formulations along with their corresponding control-relevant issues such as closed-loop stability, performance and constraint satisfaction can be found in [40, 41, 42, 43, 36, 44, 45].

Although there are substantial advantages of MPC designs especially to indus-
trial community, the stability guarantees offered in MPC are linked to the feasibility of the optimization problem and moreover the set of initial conditions starting from where a given MPC formulation is guaranteed to be feasible is not explicitly characterized. Lyapunov based controllers can provide explicitly-defined regions of attraction for the closed-loop system [46], however these controllers are usually suboptimal with respect to prespecified performance criteria and can not directly incorporate performance in the design. To alleviate these concerns [47, 48] have introduced a Lyapunov-based MPC formulation that guarantees stability from an explicitly characterized set of initial conditions in the presence of state and input constraints. This MPC formulation was also recently extended for applications involving switched nonlinear systems and hybrid systems [49, 47, 50, 51, 52, 53].

Even though most of the research in MPC seems to be focused on systems modeled by ODEs, the question of MPC designs for PDEs is also gaining interest in the research community (due to the wide range of industrial applications [25]). In [54] Lyapunov functionals were used to analyze the MPC design problem on the basis of infinite-dimensional systems while others [55] have utilized approximate ODE systems obtained through spatial discretization of the PDE system in the MPC methodologies. These designs are in general computationally expensive and hence are difficult to implement on-line. Motivated by this, Galerkin’s method was used successfully to derive ROMs, leading to MPC designs that were computationally less expensive [37, 56, 39].
However, the above formulation of ROMs cannot be directly applied to systems which have nonlinear spatial differential operators as one cannot obtain the required eigenfunctions analytically. To overcome this limitation researchers have focused on data-driven methods such as proper orthogonal decomposition (POD) and method of snapshots [57, 58]. This method has been profusely used in model reduction [59, 60, 61, 62], optimization [24, 63, 64, 65, 66, 67, 68], sensor placement [69, 70] and geometric control [31, 71, 72, 73, 74] of distributed processes without considering process constraints.

To handle the situations where in the snapshots or data sets used are incomplete (gappy), [75] proposed a modification for the basic POD based methodology. In the modification, especially focusing on image reconstruction, the basic POD based methodology was combined with the least-squares approach to estimate the necessary modal coefficients from the available incomplete data. The application of this methodology was however extended to aerodynamic [76, 77] & fluid mechanics [78] problems. An extension of the gappy POD methodology was also proposed [78, 79] to eliminate its dependence on the initial guess, used in filling the missing data in a given data vector, while finding the best possible reconstruction.

The ROM computed using POD is accurate only in a restricted zone, where the POD model is created [80]. As a result, the above data-driven methods though assume an *apriori* availability of a large ensemble of snapshots to correctly capture the incidence of new trends during the process evolution using the basis functions
computed off-line from that snapshot ensemble. However, generating such an ensemble is not straightforward (and experimentally infeasible) as it necessitates using suitably designed inputs to excite all the modes [59, 81]. This is especially limiting when considering the control problem; there is an urgent need to refine the ROMs on command as new trends appear [82, 83].

Motivated by the lack of general methodology for the computationally efficient update of ROM and its consequent use in the synthesis of model-based controllers for nonlinear dissipative PDE systems the objectives of the present doctoral thesis are:

- Development of a computationally efficient ROM methodology, called the Adaptive proper orthogonal decomposition (APOD) methodology, for the derivation & efficient update of nonlinear low-order ROMs. This algorithm will relax the requirement of a representative ensemble of solutions.

- Construction of practically implementable feedback control systems (using the above updated low-order ROMs) that can deal with the issues of nonlinearity, model uncertainty and constraints; extensions to cases when the availability of distributed sensors is restricted.

- Application of the developed controller design methods to applications in chemical engineering (transport-reaction processes) and for development of active control for the suppression of instabilities exhibited by falling and shallow liquid films).
1.1.2 Structure of thesis

Throughout the thesis, we will assume that the PDE systems under consideration possess a unique solution which is sufficiently smooth. The mathematical questions of existence, uniqueness of the solutions of these PDE systems will not be addressed in this manuscript and can be found in [84]. Also we focus on parabolic PDE systems for which the manipulated inputs and controlled outputs are bounded. The rest of the thesis is structured as follows.

Chapter 2 presents the adaptive proper orthogonal decomposition (APOD) methodology. We initially compute the empirical basis functions, by employing APOD off-line on the available off-line process data obtained from the PDE system. These basis functions will eventually be utilized in a Galerkin method framework to derive the ROM. As the validity of these models is confined to a small region of the entire state space spanned by the available initial process data; there arises a need for updating the models (starting with the update of the basis functions) using the available closed-loop process data. During the closed loop process evolution APOD updates these models (using the closed loop process data) thus extending the region of validity of these models.

In Chapter 3 we design nonlinear state feed back controllers, under the assumption of continuous availability of full state measurements, using the ROMs obtained by applying APOD. The proposed approach is applied to a typical diffusion-reaction process with both linear and nonlinear spatial differential operators, to
demonstrate the effectiveness of the methodology.

Chapter 4 focuses on the challenging problem of synthesis of nonlinear output feed back controllers for nonlinear parabolic PDE systems, using limited periodic measurements and continuous point measurements. Initially, a ROM is constructed based on a relatively small data ensemble this ensemble is recursively updated as additional process data becomes available periodically. Using the updated ROM and continuous measurements available from restricted number of sensors a static output feedback controller is subsequently designed. This controller is successfully used to achieve the desired control objective of stabilization and tracking of fluid flow processes (described by Kuramoto-Sivashinksy equation (KSE) and in FitzHugh-Nagumo equations). APOD is 360% faster compared to a straightforward POD implementation.

In Chapter 5 we employ adaptive ROMs in the design of model predictive controllers, under constraints on the control action, for stabilization of nonlinear parabolic PDE processes. As periodic closed-loop process data becomes available (during closed-loop operation under the constructed MPC), we recursively update the ROM by employing our APOD methodology thus extending the validity of ROM & the controller over the current operating region. The effects of employing the adaptive methodology on performance of MPC is studied. The design of such MPC controllers is illustrated by employing the methodology on numerical simulations.
Chapter 6 focuses on designing output feedback controllers for nonlinear PDE systems using partial state measurements. We initially reconstruct the partial measurements using gappy reconstruction technique. These reconstructed measurements are then utilized for the derivation and update of reduced order models (ROM) using our adaptive proper orthogonal decomposition (APOD) methodology. The use of APOD methodology allows the development and on-demand update of a locally accurate low-dimensional ROM thus resulting in a computationally efficient alternative to using a large dimensional ROM with global validity. Based on the low-dimensional ROM and continuous measurements available from restricted number of sensors a static output feedback controller is subsequently designed. The effectiveness of the present approach is illustrated on an unstable fluid flow process modeled by KSE equation.

In Chapter 7 we present our initial results on dynamic output feedback based robust control of quasi-linear parabolic PDE systems with time-varying uncertain variables. The states of the process required for designing controllers are dynamically estimated from limited number of noisy process measurements employing an extended Kalman filter (EKF). The designed EKF based robust controller addresses both the issues of model uncertainty and sensor noise. This methodology is successfully illustrated on an representative example wherein the desired objective is to stabilize an unstable operating point in a catalytic rod. A finite dimensional robust controller, utilizing dynamically estimated states, is used to successfully
stabilize the catalytic process to an open-loop unstable steady-state.

Finally Chapter 8 presents the main conclusions of this work along with future directions that may be pursued.
In this section we formulate and update the basis functions of the PDE system using the adaptive proper orthogonal decomposition (A POD). We initially compute the empirical basis functions by employing A POD off-line on the available off-line process data. The validity of these basis functions is confined to a small region of the entire state space spanned by the available initial process data. During the closed loop process evolution A POD updates these basis functions (using the closed loop process data) thus extending their region of validity. These updated basis functions will eventually be utilized in the formulation and update of reduced order models (ROMs).
2.1 Computation of initial basis function

In this section, we use the available off-line solution data of the PDE system (process) to construct the initial basis functions necessary for the derivation of the ROM. It is based on proper orthogonal decomposition (POD) a procedure used to compute an optimal set of empirical basis functions from an appropriately constructed set of solutions of the PDE system. In general (but not in this method; see remark 1), the ensemble of solutions is constructed by computing the solutions of the PDE system for different values of $u(t)$, and different initial conditions [59, 67, 24].

We initially formulate a data ensemble employing the available off-line process data. Application of POD to this data ensemble provides an orthogonal set of basis functions for the representation of the ensemble, as well as a measure of the relative contribution of each basis function towards the total energy (mean square fluctuation) of the ensemble. A truncated series representation of the ensemble data in terms of these dominant basis functions has a smaller mean square error than a representation by any other basis of the same dimension [58]. Therefore, it yields the most efficient way for computing the basis functions (corresponding to the largest empirical eigenvalues) capturing the dominant patterns of the ensemble.

Let $v_\kappa$ denote the snapshot of the system available at time $t_\kappa$ and let the total number of snapshots available be $N$. Note that by snapshot we represent the spatial profile of the PDE system at particular time instant. We will briefly present the
key features of POD. The reader may refer to [85, 58, 86] for a detailed presentation and analysis of POD.

We define the ensemble average as $\langle \Psi_\kappa \rangle := \frac{1}{N} \sum_{\kappa=1}^{N} \Psi_\kappa(z)$, where $\Psi_\kappa$ denotes the snapshot of a vector quantity at time $t_\kappa$ (non-uniform sampling of the snapshots and weighted ensemble averages can be also considered [59]). The issue is how to obtain the most typical or characteristic spatial profile (in a sense that will become clear below) $\phi(z)$ from these snapshots $\{v_\kappa\}$. Mathematically, this problem can be posed as the one of obtaining a function $\phi(z)$ that maximizes the following objective function:

$$\text{Maximize } \frac{\langle (\phi, v_\kappa)^2 \rangle}{\langle \phi, \phi \rangle}$$

s.t. $(\phi, \phi) = 1, \ \phi \in L^2(\Omega)$

(2.1)

which, in other words, implies that the projection of $v_\kappa$ on the subspace spanned by $\phi(z)$ captures the maximum energy contained in these snapshots. The constraint $(\phi, \phi) = 1$ is imposed to ensure that the function, $\phi(z)$, computed as a solution of the above maximization problem, is unique. An alternative way to express the constrained optimization problem of Eq.2.1 is to solve the perturbation problem for $\phi$:

$$\frac{dL(\phi + \delta \psi)}{d\delta}(\delta = 0) = 0, \ (\phi, \phi) = 1$$

(2.2)

where $L = \langle (\phi, v_\kappa)^2 \rangle - \lambda((\phi, \phi) - 1)$ is the corresponding Lagrangian functional and $\delta$ is a real number. Using the definitions of inner product and ensemble
average, \( \frac{dL(\phi + \delta\psi)}{d\delta}(\delta = 0) \) can be computed from the following expression:

\[
\frac{dL(\phi + \delta\psi)}{d\delta}(\delta = 0) = \int_{\Omega} \left( \left\{ \int_{\Omega} \langle v_\kappa(z) v_\kappa(\bar{z}) \rangle \phi(z) dz \right\} - \lambda \phi(\bar{z}) \right) \Gamma(\bar{z}) d\bar{z} \tag{2.3}
\]

Since \( \Gamma(\bar{z}) \) is an arbitrary function, the necessary conditions for optimality take the form:

\[
\int_{\Omega} \langle v_\kappa(z) v_\kappa(\bar{z}) \rangle \phi(z) dz = \lambda \phi(\bar{z}), \quad (\phi, \phi) = 1 \tag{2.4}
\]

Introducing the two-point correlation function:

\[
K(z, \bar{z}) = \langle v_\kappa(z) v_\kappa(\bar{z}) \rangle = \frac{1}{N} \sum_{\kappa=1}^{N} v_\kappa(z) v_\kappa(\bar{z}) \tag{2.5}
\]

and the integral operator:

\[
R := \int_{\Omega} K(z, \bar{z}) d\bar{z} \tag{2.6}
\]

the optimality condition of Eq.2.4 reduces to the following eigenvalue-eigenfunction problem of the integral operator:

\[
R\phi = \lambda\phi \implies \int_{\Omega} K(z, \bar{z}) \phi(\bar{z}) d\bar{z} = \lambda \phi(z) \tag{2.7}
\]

The computation of the solution of the above integral eigenvalue problem is, in general, a very expensive computational task. To circumvent this problem, Sirovich, in 1987, introduced the method of snapshots [57]. The central idea of this tech-
unique is to assume that the requisite eigenfunction, $\phi(z)$, can be expressed as a linear combination of the snapshots i.e.,

$$
\phi(z) = \sum_k \psi^k v_k(z)
$$

(2.8)

where $\psi^k$ denotes the kth element of vector $\psi$.

Substituting the above expression for $\phi(z)$ on Eq.2.7, we obtain the following eigenvalue problem:

$$
\int_{\Omega} \frac{1}{N} \sum_{\kappa=1}^{N} v_{\kappa}(z)v_{\kappa}(\bar{z}) \sum_{k=1}^{N} \psi^k v_k(\bar{z}) d\bar{z} = \lambda \sum_{k=1}^{N} \psi^k v_k(z)
$$

(2.9)

Defining the $(\kappa,k)$th element $C_{\kappa k}^N$ of matrix $C_N$ as

$$
C_{\kappa k}^N := \frac{1}{N} \int_{\Omega} v_{\kappa}(\bar{z}) v_k(\bar{z}) d\bar{z}
$$

(2.10)

the eigenvalue problem of Eq.2.9 can be equivalently written as:

$$
C_N \psi = \lambda \psi
$$

(2.11)

The solution of the above eigenvalue problem (obtained by utilizing standard methods from linear algebra [87]) yields $N$ eigenvectors $\psi_1, \psi_2, \cdots \psi_N$ which can be used in Eq.2.8 to construct $N$ basis functions $\phi_\kappa(z), \kappa = 1, \cdots, N$. By construction, matrix $C_N$ is symmetric and positive semi-definite, and thus, its eigenvalues, $\lambda_\kappa$,
κ = 1, . . . , N, are real and non-negative. The relative magnitude of the eigenvalues represents a measure of the fraction of the “energy” embedded in the ensemble captured by the corresponding basis functions. We order the calculated empirical basis functions such that

\[ \lambda_1 > \lambda_2 > \ldots > \lambda_N > 0. \]  

(2.12)

Furthermore, the resulting basis functions form an orthonormal set, i.e.:

\[ \int_{\Omega} \phi_i(z)\phi_j(z)dz = 0, \quad i \neq j \quad \text{and} \quad \int_{\Omega} \phi_i(z)\phi_i(z)dz = 1. \]  

(2.13)

Iterative methods, such as Krylov subspace methods [87], can be used to reduce the computational cost associated with the computation of the eigenvalues and basis functions.

**Remark 1.** *We will recursively update the basis functions using the closed-loop process data in APOD consequently, we need not perform an exhaustive sampling of the state-space of the PDE by evolving the system from a number of different initial conditions and for different values of actuation during ensemble generation.*
2.2 Adaptive Proper Orthogonal Decomposition

POD requires *apriori* availability of sufficiently large ensemble of PDE solution data in which all the possible spatial modes (including those that might appear in closed-loop evolution of the system) are excited. This is necessary to ensure that the resulting basis functions (and hence the ROM derived using these basis functions) capture the global dynamics of the system during the closed-loop process evolution. This large ensemble of solution data should then be generated by computing the solutions of the PDE system for different values of \( u(t) \) and different initial conditions [59]. However, it is difficult to generate such an ensemble and no well defined methodology exists for generating it.

The resulting empirical basis functions, therefore, are representative of the corresponding ensemble only. As a result, the approximate ROM model obtained using these basis functions in Galerkin’s method may not be a valid approximation of the PDE model in a broad region of state space. Also complete open-loop excitation of the process would be not be sufficient, as the closed-loop process will not in general stay within the same region of state space during its evolution. As a result the optimal basis functions obtained through the open-loop system excitation may actually become superfluous and irrelevant, thus increasing the observer size unnecessarily or even negatively impacting accuracy.

One possible solution is to continue augmenting the ensemble of snapshots and subsequently recomputing the basis functions as more information regarding the
process becomes available. However, this would require repeated evaluations of the
eigenvalue-eigenvector problem of a continuously increasing in size covariance ma-
trix which may become computationally expensive with time and hence unsuitable
for online computations.

To circumvent the latter problems, a computationally inexpensive algorithm
[88, 89] that allows for construction and recursive update of eigenfunction, once
new measurements from the process become available, is needed. Our efforts in
this direction will now be presented. This methodology called adaptive proper
orthogonal decomposition (A POD) consists of two steps:

2.2.1 Initial basis Construction

We initially utilize the available collection of N off-line data snapshots, \( \{ v_k \}_{k=1}^N \), to
construct the initial basis for the PDE system. We first construct the covariance
matrix \( C_N \) then solve the following eigenvalue-eigenvector problem (Eq.2.11)

\[
C_N \psi = \lambda \psi
\]

to compute N eigenvalues. We partition the eigenspace of the covariance matrix,
\( C_N \), into two subspaces; the dominant one containing the modes which capture
at least \( \epsilon \) percent of energy in the ensemble (denoted as \( \mathbb{P} \)) and the orthogonal
complement to \( \mathbb{P} \) containing the rest of the modes (denoted as \( \mathbb{Q} \)). Such a partition
is possible due to the fact that the dominant dynamics of dissipative PDEs are
finite (typically small) dimensional [9]. Note that we define $\epsilon$ as the percentage energy of the ensemble captured by dominant basis functions. We assume that out of $N$ possible eigenvectors of $C_N$, $m$ have the corresponding eigenvalues such that \( \sum_{i=1}^{m} \frac{\lambda_i}{\sum_{i=1}^{N} \lambda_i} \leq \frac{\epsilon}{100} \); $m$ eigenmodes of $C_N$ capture $\epsilon$ percent of energy in the ensemble. These eigenvectors are then used in the following equation

$$\phi_i(z) = \sum_k \psi_k^i v_k(z), \ i = 1, \cdots, m.$$  

to compute $N$ basis functions; here $\phi_i$ represents the $i^{th}$ eigenfunction and $\psi_i$ is the $i^{th}$ eigenvector of $C_N$. An orthonormal basis for the subspace $\mathbb{P}$ can be obtained as:

$$Z = [\psi_1, \psi_2, \ldots, \psi_m], \ Z \in \mathbb{R}^{N \times m}$$  \hspace{1cm} (2.14)

where $\psi_1, \psi_2, \ldots, \psi_m$ denote the eigenvectors of $C_N$ that correspond to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$. Note that the eigenfunctions computed by these eigenvectors capture the dominant dynamics of the PDE system. The orthogonal projection operators $P$ and $Q$ onto subspaces $\mathbb{P}$ and $\mathbb{Q}$ can be computed as

$$P = ZZ^T, \ Q = I - ZZ^T$$  \hspace{1cm} (2.15)

where $I$ denotes the identity matrix of dimension $N$. 
2.2.2 Online basis refinement

During the course of closed loop process evolution we assume that periodically new snapshots become available. We recursively update the orthonormal basis for the subspace $\mathbb{P}$ upon the periodic arrival of these new snapshots, possibly by increasing or decreasing the size of the basis if required and by maintaining the accuracy of basis by performing orthogonal power iteration, while the orthonormal basis for $\mathbb{Q}$ can be computed from the fact that $\mathbb{Q}$ is the orthonormal complement of $\mathbb{P}$. We maintain that the extra work required for the above process is small as long as the dimension of $\mathbb{P}$ is small (this amounts to choosing an appropriate value for $\epsilon$). The algorithm outlined below computes an approximation to $Z$ without requiring the solution of the eigenvalue-eigenvector problem of the covariance matrix (Eq.2.11). We assume that during each step at most one eigenmode joins the subspace $\mathbb{P}$. We also assume that the process sampling is fast enough so that the appearance of new patterns is captured during process evolution. The algorithm requires the dimensionality of the covariance matrix, $C_N$, to remain constant, which we achieve by discarding the oldest snapshot from the ensemble as a new one is obtained. As a new snapshot from the process becomes available, the subspace $\mathbb{P}$ may change in the following three ways:

- The dimension of the dominant subspace $\mathbb{P}$ may increase i.e., one mode corresponding from $\mathbb{Q}$ becomes necessary to capture the desired percentage of energy in the ensemble.
Some of the eigenmodes of the subspace $\mathcal{P}$ may no longer be necessary to capture the required $\epsilon$ percent of the energy. In this case, the basis $Z$ should be updated and its dimension should be simultaneously decreased.

- The dimensionality of $\mathcal{P}$ remains unchanged. However, the basis $Z$ is updated, whenever the current basis is not accurate, to maintain the accuracy of the basis.

A flow chart illustrating the above steps is presented in Figure 2.1. In the following subsections, the above steps are explained in detail.

### 2.2.2.1 Increasing the size of the basis

In this section, we consider the case when one eigenmode of subspace $\mathcal{Q}$ becomes dominant as a new snapshot is added, in the sense that the associated eigenfunction becomes necessary to accurately describe the ensemble of solutions. The eigenmode thus becomes an element of subspace $\mathcal{P}$ and leaves $\mathcal{Q}$. To monitor this event we employ the following lemma.

**Lemma 1.** The eigenvalues of $c_q = QC_NQ$ is the subset of eigenvalues of $C_N$ that correspond to the eigenmodes which belong to subspace $\mathcal{Q}$.

See Appendix A for proof.

We thus monitor the percentage contribution of the dominant eigenvalue of $c_q = QC_NQ$, namely $\lambda_{m+1}$ towards the total energy of the ensemble. We define
Figure 2.1. Flow Chart of Adaptive Proper Orthogonal Decomposition methodology.
the percentage contribution of $\lambda_{m+1}$ as

$$\xi = \frac{\lambda_{m+1}}{\sum_{i=1}^{m+1} \lambda_i}$$  \hspace{1cm} (2.16)$$

If $\xi$ increases to more than $(100 - \epsilon)$ percent we append $Z$, the basis of subspace $\mathbb{P}$, with the corresponding eigenvector (We note that the eigenvalues $\lambda_i \forall i=1,\ldots,m$ can be computed by solving a small eigenvalue problem, see next subsection for details). We find the dominant eigenspace of $c_q$ using the following power iteration

$$q^{(v+1)} \approx (c_q)^v q^{(0)}$$  \hspace{1cm} (2.17)$$

The above iteration then produces iterates that asymptotically lie in the dominant eigenspace of $c_q$, provided the initial iterate $q^{(0)}$ has a nonzero component in that direction. The dominant eigenvalue of $c_q$ can be computed as

$$\lambda_{m+1} = (q^c)^T c_q q^c$$  \hspace{1cm} (2.18)$$

where $q^c$ is the converged solution of the power iteration Eq.2.17. The new eigenvector then becomes part of

$$Z = [\psi_1, \psi_2, \ldots, \psi_m, \psi_{m+1}], \ Z \in \mathbb{R}^{K \times m+1}$$

and its eigenvalue gets added to the dominant subspace $\mathbb{P}$. Consequently, the
2.2.2.2 Decreasing the size of the basis

As new snapshots are added and old snapshots are eliminated from the ensemble, the dominant eigenspace of $C_N$ continuously changes. Power iteration in the previous section identifies scenarios when one of the eigenmodes becomes dominant. However, it is likely that during the process some of the eigenmodes in subspace $\mathbb{P}$, may no longer be necessary to capture the desired percentage of energy. In such cases, it is required to decrease the size of basis $Z$ such that it spans the dominant eigenspace only. To test whether it is required to decrease the size of the basis we introduce the following $m \times m$ matrix

$$H = Z^T C_N Z, \quad H \in \mathbb{R}^{m \times m}.$$ (2.19)

The eigenvalues of $H$ are a subset of the eigenvalues of $C_N$ and can be computed with little computational effort as long as $m$ remains small. If only $\hat{m}$, with $\hat{m} < m$, eigenvalues of $H$ are dominant (in a sense that only $\hat{m}$ eigenmodes are required to capture $\epsilon$ percent of energy of the ensemble), then span$\{ZV\}$ provides a good approximation to the dominant eigenspace of $C_N$, where the basis $V \in \mathbb{R}^{m \times \hat{m}}$ is obtained from the eigenvectors of $H$ corresponding to its $\hat{m}$ dominant eigenvalues. Hence, the step

$$Z = \text{orth}(ZV)$$ (2.20)
where \( \text{orth}(\cdot) \) denotes Gram-Schmidt orthonormalization, automatically reduces the size of the basis whenever required.

### 2.2.2.3 Maintaining the accuracy of the basis

During the process evolution it may become apparent that even though the basis dimensionality remains the same, there is an increase in the error between the new snapshots and the projection identified using the “old” basis functions. The following one step orthogonal power iteration is performed in order to maintain the accuracy of the basis after each addition of a snapshot.

\[
Z = \text{orth}(C_NZ).
\]  
\[
(2.21)
\]

**Remark 2.** We note that the orthogonal projections \( P \) and \( Q \) should satisfy \( QC_NP = 0 \) (a result which follows from the fact that \( P \) is an orthogonal projector, see Appendix A). Hence, the accuracy of the basis can be also evaluated by computing the matrix \( \mathcal{E} = (I - ZZ^T)C_N(ZZ^T) \). If the norm of \( |\mathcal{E}| \geq |\mathcal{E}_{\text{thr}}| \), where \( \mathcal{E}_{\text{thr}} \) is a user defined threshold, then we maintain the accuracy of the basis.

A flow chart illustrating the above steps is presented in Figure 2.1. Based on the new values of \( Z \), we now compute the revised eigenfunctions \( \phi_1, \phi_2, \ldots, \phi_m \) as a linear combination of the snapshots given by the following equation

\[
\phi_i(z) = \sum_k \psi_i^k v_k(z), \quad i = 1, \ldots, m.
\]  
\[
(2.22)
\]
where $\psi_i^k$ denotes the $k^{th}$ element of vector $\psi_i$.

**Remark 3.** In the proposed approach for model reduction approach the snapshots used are obtained during closed-loop system evolution as opposed to all other proper orthogonal decomposition-based reduction approaches that are based on open-loop snapshots, and thus, these snapshots and the resulting ROM account for the impact of controller functional form on the process. It is important to note this intimate relation between APOD and the controller.

**Remark 4.** The foundation of the present APOD methodology lies on the statistical inference of trends in data based on observations, that in general contain noise. As a result the methodology retains the attributes of PCA [90] i.e., the robustness with respect to random noise in measurements. Similarly to standard PCA, measurement noise is rejected in APOD due to the exclusion of basis functions, that have very small positive eigenvalues from subspace $\mathbb{P}$. These basis functions are the ones that primarily contain the effects of measurement noise.
3.1 Introduction

The problem of state feedback control of spatially distributed processes described by dissipative partial differential equations (PDEs) is considered. Typically, this problem is addressed through model reduction where finite dimensional reduced order models (ROM’s) to the original infinite dimensional PDE system are derived. The key step in this approach is the computation of basis functions that are subsequently utilized to obtain the ROM’s using the method of weighted residuals. A common approach to this task is the proper orthogonal decomposition combined with the method of snapshots. To circumvent the issue of a priori availability of a sufficiently large ensemble of PDE solution data, we focus on the recursive
computation of basis functions as additional data from the process become available. Initially, an ensemble of basis functions is constructed based on a relatively small number of snapshots and the covariance matrix is computed. The dominant eigenspace of this matrix is then utilized to compute the empirical basis functions required for model reduction. This dominant eigenspace is recomputed with the addition of each snapshot with possible increase or decrease in its dimensionality; due to its small dimensionality the computational burden is relatively small. The proposed approach is applied to representative examples of dissipative PDEs, with both linear and nonlinear spatial differential operators, to demonstrate the effectiveness of the proposed methodology.

3.2 Mathematical Preliminaries

We focus on the problem of feedback control of spatially distributed processes described by highly dissipative PDEs with the following state-space description:

\[
\frac{\partial x}{\partial t} = A(x) + b(z)u + f(x),
\]

\[
y_e = \int_{\Omega} c(z) x \, dz,
\]

\[
y_m = \int_{\Omega} s(z) x \, dz.
\]
subject to the mixed-type boundary conditions:

\[ q(x, \frac{dx}{d\eta}, \ldots, \frac{d^{n_o-1}x}{d\eta^{n_o-1}}) = 0 \text{ on } \Gamma \] \hspace{1cm} (3.2)

and the following initial condition

\[ x(z, 0) = x_0(z). \] \hspace{1cm} (3.3)

In the above PDE system, \( x(z, t) \in \mathbb{R}^n \) denotes the vector of state variables, \( y_c \in \mathbb{R}^m \) denotes the vector of controlled outputs, \( t \) is the time, \( y_m \in \mathbb{R}^{n_{m_o}} \) denotes the vector of measured outputs, \( z \in \Omega \subseteq \mathbb{R} \) is the spatial coordinate, \( \Omega \) is the domain of definition of the process and \( \Gamma \) is its boundary. \( A(x) \) is a highly dissipative, possibly nonlinear, spatial differential operator of order \( n_o \), where \( n_o \) is an even number; \( f(x) \) is a nonlinear vector function which is assumed to be sufficiently smooth with respect to its arguments, \( c(z) \) is a known smooth vector function of \( z \) which is determined by the desired performance specifications in the domain \( \Omega \) and \( s(z) \) is a known smooth vector function of \( z \) which is determined by the location and type of measurement sensors (e.g., point or distributed sensing), \( u = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^k \) denotes the vector of manipulated inputs, \( b(z) \in \mathbb{R}^{n \times k} \) is a known smooth matrix function of \( z \) of the form \( [b_1(z), b_2(z), \ldots, b_k(z)] \), where \( b_i(z) \) describes how the \( i^{th} \) control action \( u_i(t) \) is distributed in the spatial domain \( \Omega \), \( q(x, \frac{dx}{d\eta}, \ldots, \frac{d^{n_o-1}x}{d\eta^{n_o-1}}) \) is a nonlinear vector function which is assumed to be sufficiently smooth, \( \frac{dx}{d\eta} |_{\Gamma} \)
denotes the derivative in the direction perpendicular to the boundary and $x_0(z)$ is a smooth vector function of $z$. We note that in the case of point actuation which influences the system at $z_0$ (i.e., $b_i(z)$ is equal to $\delta(z - z_0)$ where $\delta(\cdot)$ is the standard Dirac function), we approximate the function $\delta(z - z_0)$ by the finite value $1/2\epsilon$ in the interval $[z_0 - \epsilon, z_0 + \epsilon]$ (where $\epsilon$ is a small positive real number) and zero elsewhere in the domain. We define the inner product and norm in $L_2[\Omega]$, where $L_2[\Omega]$ is the space of square integrable functions defined in $\Omega$, as follows:

\[
(\phi_1, \phi_2) = \int_{\Omega} \phi_1^*(z)\phi_2(z)dz, \quad ||\phi_1||_2 = (\phi_1, \phi_1)^{1/2}
\]  

(3.4)

where $\phi_1, \phi_2 \in L_2[\Omega]$ and $^*$ denotes the complex conjugate transpose.

We also define the Lie derivative of the scalar function $h_i(x)$ with respect to the vector function $f(x)$ as $L_fh_i(x) = \frac{\partial h_i(x)}{\partial x} f(x)$. $L^k_i h_i(x)$ denotes the $k$-th order Lie derivative and $L_g L^k_j h_i(x)$ denotes the mixed Lie derivative.

### 3.3 Problem formulation and solution methodology

In this section, our objective is to present an outline of the steps of the adaptive model reduction and control methodology for processes that are described by the system of Eqs.3.1-3.3. The control problem is formulated as the one of deriving a feedback control law $u(t) = G(x(t))$, such that the closed-loop system is stabilized.
at a desired set point. Without loss of generality, we assume the setpoint is $x(z, t) = 0$. The steps of the proposed methodology to achieve the above task are:

0. Initially generate an ensemble of solution data either through experimental observations or from detailed numerical simulations.

1. Analyze the PDE solution data and obtain the basis functions using POD (section 2.1) and use the method of weighted residuals to construct a finite-dimensional reduced order model (ROM) of the PDE.

2. Design a state feedback controller based on the above ROM.

3. Utilize the adaptive POD methodology (section 2.2) to recursively modify the basis functions and the ROM as new process measurements become available from the PDE system. This step may require addition or deletion of basis functions thus may result in a dimensionality change of the ROM.

4. Redesign the feedback controller based on the modified ROM.

A closed-loop diagram illustrating the different steps of the developed methodology is presented in Figure 3.1. We note that the initial data required for evaluating the basis functions can be obtained either experimentally (by allowing the process to evolve with no control action for a short period of time) or from previously obtained historical data, or numerically by using offline simulation packages for PDE systems such as Fluent and Comsol and subsequently evaluating the simulation results. The following subsections are intended to describe the above steps in detail.
3.3.1 Formulation of reduced order model

We employ the identified empirical basis functions to derive finite-dimensional approximations of the infinite-dimensional PDE system of Eq.3.1 by using the method of weighted residuals. To simplify the notation, without loss of generality we consider the system of Eq.3.1 with \( n = 1 \). In principle, \( x(z, t) \) can be represented as an infinite weighted sum of a complete set of basis functions \( \phi_k(z) \). We can obtain an approximation \( x_N(z, t) \), by truncating the series expansion of \( x(z, t) \) up to order \( N \), as follows:

\[
x_N(z, t) = \sum_{k=1}^{N} a_k(t) \phi_k(z) \xrightarrow{N \to \infty} x(z, t) = \sum_{k=1}^{\infty} a_k(t) \phi_k(z) \tag{3.5}
\]
where $a_k(t)$ is a time-varying coefficient called the mode of the system.

Substituting the expansion of Eq.3.5 into Eq.3.1, multiplying the PDE with the weighting functions, $\omega(z)$, and integrating over the entire spatial domain (i.e., taking inner product in $L_2[\Omega]$ with the weighting functions), the following $N$-th order system of ODEs is obtained.

\[
-\sum_{k=1}^{N} \dot{a}_k \left( \int_{\Omega} \omega(z) \phi_k(z) dz \right) \\
+ \int_{\Omega} \omega(z) A \left( \sum_{k=1}^{N} a_k(t) \phi_k(z) \right) dz + \int_{\Omega} \omega(z) b(z) u dz \\
+ \int_{\Omega} \omega(z) f \left( \sum_{k=1}^{N} a_k(t) \phi_k(z) \right) dz = 0, \quad v = 1, \ldots, N
\]

(3.6)

The weighting functions in the above equation determine the type of weighted residual method being used. When the weighting functions are the basis functions, $\Omega_k(z) = \phi_k(z)$, the method of weighted residuals reduces to Galerkin’s method. In the proposed methodology, since we are using empirical basis functions for this expansion we slightly abuse the terminology and we continue to call this Galerkin’s method. The resulting ODE system can be written compactly in the form

\[
\dot{a} = \mathcal{F}(a) + \mathcal{G} u
\]

(3.7)

where $a \in \mathbb{R}^N$ are now called empirical eigenmodes and $\mathcal{F}$, $\mathcal{G}$ are vector and matrix functions of appropriate dimensions defined in Eq.3.9 below.
3.3.2 Controller design using feedback linearization

In this section, we employ feedback linearization to design state feedback controllers for the system of Eq.3.1, based on the representation of Eq.3.6. To simplify the development, we represent the system in the following compact form:

\[
\dot{a} = \mathcal{F}(a) + \mathcal{G}u = \mathcal{F}(a) + \sum_{i=1}^{k} \mathcal{G}_i u_i,
\]

\[
y_m = \int_{\Omega} s(z) x \, dz,
\]

\[
y_{c,i} = h_i(a), \quad i = 1, \ldots, k
\]

where

\[
\mathcal{F}(a) = \int_{\Omega} \phi(z) A \left( \sum_{k=1}^{N} a_k(t) \phi_k(z) \right) dz + \int_{\Omega} \phi(z) f \left( \sum_{k=1}^{N} a_k(t) \phi_k(z) \right) dz
\]

\[
\mathcal{G}_i = \int_{\Omega} \phi(z) b_i(z) \, dz
\]

\[
h_i(a) = \int_{\Omega} c^i(z) x \, dz
\]

Where \( k \) is the number of manipulated inputs, \( u_i \) is the \( i^{th} \) manipulated input, \( y_m \) is the measured output vector, \( S \) denotes the measurement sensor shape function, and \( y_{c,i} \) is the \( i^{th} \) controlled output. We assume that the relative degree \( r_i \) of the system of Eq.3.8 is well defined for all values of \( a \). [1].

We use feedback linearization to design state feedback controllers which have
the following general form:

\[ u = p(a) + Q(a)\hat{v} \quad (3.10) \]

where \( p(a) \) is a smooth vector function, \( Q(a) \) is a smooth matrix, and \( \hat{v} \in \mathbb{R}^k \) is the constant reference input vector. Based on the relative degrees of the system of Eq.3.7, we assign the following closed-loop behavior to the controlled outputs \( y_{c,i} \):

\[
\sum_{i=1}^{k} \sum_{j=0}^{r_i} \beta_{ij} \frac{d}{dt} y_{c,i} = \hat{v}. \quad (3.11)
\]

The characteristic matrix of the system Eq.3.8

\[
C_0(a) = \begin{bmatrix}
L_{G_1} L_{\mathcal{F}r_1}^{-1} h_1(a) & \cdots & L_{G_l} L_{\mathcal{F}r_1}^{-1} h_1(a) \\
L_{G_1} L_{\mathcal{F}r_2}^{-1} h_2(a) & \cdots & L_{G_l} L_{\mathcal{F}r_2}^{-1} h_2(a) \\
\vdots & \vdots & \vdots \\
L_{G_1} L_{\mathcal{F}r_k}^{-1} h_k(a) & \cdots & L_{G_l} L_{\mathcal{F}r_k}^{-1} h_k(a)
\end{bmatrix}
\]

is assumed to be invertible (i.e., \( \det(C_0(a)) \neq 0 \)). Eq.3.8 can be used to derive state feedback controller (of the form given by Eq.3.10) that guarantee output behavior as described by Eq.3.11:

\[
u = \{[\beta_{1r_1} \ldots \beta_{kr_k}] C_0(a)\}^{-1}\{\hat{v} - \sum_{i=1}^{k} \sum_{j=0}^{r_i} \beta_{ij} L_j^i h_i(a)\}. \quad (3.12)
\]

The proof of the closed-loop properties of the proposed controller can be found in
and are omitted for brevity.

### 3.3.3 Recursive update of empirical basis functions and controller reconfiguration

The basis functions computed by the POD in section 2.1 requires *a priori* availability of a sufficiently large ensemble of PDE solution data to compute empirical basis functions. However, in practice, it is difficult to generate such an ensemble so that all possible dominant spatial modes are appreciably contained within the corresponding snapshots. The resulting basis functions, therefore, are representative of the corresponding ensemble only. During closed-loop simulation, situations may arise when the existing basis functions fail to accurately represent the dynamics of the PDE system.

One possible solution is to continue augmenting the ensemble of snapshots and subsequently recomputing the basis functions as more information regarding the process becomes available. However, this would require the solution of the eigenvalue-eigenvector problem of Eq.2.11, which may become computationally expensive and hence unsuitable for online computations as the process evolves. Furthermore, while we are traversing different regions of the state-space during the process evolution, old snapshots may not contain pertinent information of the process behavior in the local region.

To circumvent the latter problems, we utilize the adaptive POD methodology
proposed in section 2.2 for recursively updating the empirical basis functions once the new measurements from the process become available. Initially, we update the orthonormal basis, $Z$ for the subspace $\mathbb{P}$. Note that this basis is obtained as

$$Z = [\psi_1, \psi_2, \ldots, \psi_m], \quad Z \in \mathbb{R}^{K \times m}$$ (3.13)

where $\psi_1, \psi_2, \ldots, \psi_m$ denote the eigenvectors of $C_N$ that correspond to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$. As a new snapshot from the process becomes available, the subspace $\mathbb{P}$ is updated in one of the three different ways using APOD methodology (section 2.2). Based on the obtained new values of $Z$, we now compute the revised basis functions $\phi_1, \phi_2, \ldots, \phi_m$ using (eq.2.8)

$$\phi(z) = \sum_k \psi^k v_k(z)$$

Using these basis functions, we refine our reduced-order model

$$\dot{a} = \mathcal{F}(a) + Gu$$

and reconfigure the controller (Eq.3.12) using the updated reduced-order model. This step assures that the reduced-order model captures new trends that appear when the process traverses through variable state space during closed-loop operation.
3.4 Application to diffusion-reaction process

3.4.1 Linear spatial operator

In this section, we apply the proposed adaptive model reduction and control methodology to a typical diffusion-reaction process that exhibits nonlinear dynamic behavior. Specifically, we consider an elementary exothermic reaction \( \text{A} \rightarrow \text{B} \) taking place on a thin catalytic rod (Figure 3.2). The temperature of the rod is adjusted by means of an actuator located along the length of the rod. Assuming that the reactant A is present in excess, the spatial profile of the dimensionless temperature of the rod is described by the following parabolic PDE:

\[
\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_T (e^{-\gamma/(1+x)} - e^{-\gamma}) + \beta_U (b(z)u(t) - x)
\] (3.14)

![Figure 3.2. Catalytic rod.](image)
subject to the following boundary conditions:

\[ x(0, t) = 0, \ x(\pi, t) = 0, \ x(z, 0) = x_0(z) \]  \hspace{1cm} (3.15)

where \( x \) denotes the dimensionless rod temperature, \( z \) is the spatial coordinate along the axis of the rod, \( \beta_T \) denotes the dimensionless heat of reaction, \( \gamma \) denotes the dimensionless activation energy, \( \beta_U \) denotes the dimensionless heat transfer coefficient, \( u(t) \) denotes the magnitude of actuation, and \( b(z) \) accounts for the spatial profile of the actuator. Two different spatial distributions for the actuator were investigated. Initially, spatially distributed actuation with \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \), where \( H(\cdot) \) denotes the standard Heaviside function, was considered. A point actuator was also considered, the actuator distribution function in this case being expressed by \( b(z) = \delta(z - 0.4\pi) \), where \( \delta(\cdot) \) denotes the modified Dirac function. The nominal values of the parameters are: \( \beta_T = 16, \ \gamma = 2, \) and \( \beta_U = 2. \)

Figure 3.3a presents the evolution of the PDE for \( u(t) = 0 \) from an initial condition of \( x(z, 0) = 0.5 \). It can be observed that the system evolves away from the above steady-state to another steady-state characterized by a non-uniform distribution of temperature across the rod with the maximum reached at \( z = \pi/2 \). Hence, we conclude that the steady-state \( x(z, t) = 0 \) is an unstable one.

The control problem can be formulated as designing a state feedback controller that stabilizes the rod temperature to the spatially open-loop unstable steady-
state. The controlled output, $y_{c,i}$, is chosen to be the first dominant eigenmode (function $c(z)$ in Eq.3.1 is chosen to be the first dominant basis function) with a desired setpoint of $\hat{v} = 0$. We assume that the measurements of all the states of Eq.3.14 are available and focus on state feedback control. The relative degree of the system $r_i$ is one and the number of control inputs, $k$, is one. The controller parameters used in Eq.3.11 are $\beta_{11} = 1$ and $\beta_{10} = 0.95$.

### 3.4.2 Numerical results

In order to obtain a finite dimensional approximation of the infinite dimensional system of Eq.3.14, initially an ensemble of 100 snapshots was generated for $u(t) = 0$. This has been presented in Figure 3.3a. Each snapshot is a spatial profile obtained at a fixed time instant from numerical simulations of Eq.3.14-Eq.3.15.
The values of the discretized snapshot used were computed at 120 equispaced locations. Note that an exhaustive sampling of the state-space of the PDE for a number of different initial conditions and magnitudes of actuation was not required during the ensemble generation phase. Application of KLE at this initial stage resulted in a single dominant basis function which captured more than 99% of the energy embedded in the ensemble. This basis function is shown in Figure 3.3b. Based on this basis function a one dimensional reduced-order model was derived by applying Galerkin’s method to the PDE system of Eq.3.14-3.15. Based on this system a feedback linearizing controller was designed. During closed-loop process operation it was assumed that snapshots of the process evolution are available every $t_s = 0.1$ seconds. The finite dimensional process model and the control law were reconfigured after each new process measurement was received. In this work, the value of the user parameter $\epsilon$ was set to 99 i.e., the modes of the dominant subspace $P$ captured at least 99% of the energy in the ensemble.

The process was initially simulated with spatially distributed actuation $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$. Figure 3.4a presents the spatiotemporal profile of the rod temperature under closed-loop operation. We observe that the controller stabilizes the process at the open-loop unstable steady-state. Figure 3.4b presents the corresponding profile of the control action $u(t)$. We note that the control action $u(t)$ is a smooth function of time (i.e., the temporal profile exhibits no discontinuities or chattering) and converges to zero as the control objective is
Figure 3.4.  a) Closed-loop temperature profile of the system of Eq.3.14 using distributed actuation, $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$. b) Temporal profile of control action using distributed actuation for the system of Eq.3.14. c) Temporal profile of the dominant basis function for the system of Eq.3.14 when using distributed actuation. d) Temporal profile of the number of dominant empirical basis functions used in the reduced-order ODE model Eq.3.8 to capture the desired 99% energy of the ensemble.

achieved. Even though more process measurements from the closed-loop operation were included in the ensemble while simultaneously old snapshots were removed, one basis function was found to capture 99% energy of the ensemble during the process evolution. This can be observed in Figure 3.4d; the percentage contribution of the dominant eigenvalue of $c_q$, $\xi$, in this case never increased to more than 1 and hence the subspace $\mathbb{P}$ was not augmented. The dominant basis functions, however,
was recomputed to account for the continuously changing ensemble of snapshots as presented in subsection 2.2. The temporal profile of this dominant basis function is presented in Figure 3.4c.

![Figure 3.5](image)

**Figure 3.5.** a) Closed-loop temperature profile of the system of Eq.3.14 using point actuation, \( b(z) = \delta(z - 0.4\pi) \). b) Temporal profile of control action using point actuation. c) Temporal profile of the first dominant basis function for the system of Eq.3.14 using a point actuator. d) Temporal profile of the second dominant basis function for the system of Eq.3.14.

Subsequently, a point actuator \( b(z) = \delta(z - 0.4\pi) \) was employed to control the same process with the same control objective. Figure 3.5a presents the spatiotemporal profile of the rod temperature under closed-loop operation. We observe that the controller successfully stabilizes the process at the open-loop unstable steady-
state. Figure 3.5b presents the corresponding profile of the control action $u(t)$. We again observe that the control action is a smooth function of time; no discontinuities in the control action appear and $u(t)$ approaches and remains at zero as the control objective is achieved. Figures 3.5c and 3.5d present the temporal profiles of the first and second basis function.

**Figure 3.6.** a) Number of dominant empirical basis functions used in the reduced-order model (Eq.3.8) for the system of Eq.3.14 as a function of time for point actuation, $b(z) = \delta(z - 0.4\pi)$. b) Temporal profiles of eigenvalues of $H$ and of the largest eigenvalue of $c_q$ for the system of Eq.3.14 using a point actuator.

Figure 3.6a presents the temporal profile of the number of empirical basis functions employed to obtain the reduced-order process model. As more process measurements from the closed-loop operation were included in the ensemble while simultaneously old snapshots were removed, a new basis function became dominant and joined the dominant eigenspace $\mathbb{P}$ at $t = 1.5$. Consequently, the dimensionality of the reduced-order ODE model was increased from $m = 1$ to $m = 2$. This can also be explicitly observed in Figure 3.6b, as $\xi$ increased to more than 1%, the
basis of subspace $\mathbb{P}$ was appended. The controller was reconfigured at this point based on the revised process model. In addition, the basis functions were recomputed to account for the new trends appearing as the ensemble of snapshots was revised. The temporal profiles of these two dominant basis functions are presented in Figures 3.5c and 3.5d.

3.4.3 Spatially distributed actuation with nonlinear spatial operator and a spatially varying coefficient

In this section, we apply the proposed finite dimensional adaptive control method to the diffusion-reaction process considered in section 3.4.2 when the spatial differential operator is nonlinear (e.g., nonlinear dependence of thermal conductivity on temperature) and the dimensionless reaction rate constant $\beta_T$ is spatially-varying. In this case, the process model is given by the following nonlinear parabolic PDE:

$$\frac{\partial x}{\partial t} = \frac{\partial}{\partial z} \left( k(x) \frac{\partial x}{\partial z} \right) + \beta_T(z)(e^{-\gamma/(1+x)} - e^{-\gamma}) + \beta_U(b(z)u(t) - x)$$

subject to the following boundary conditions:

$$x(0,t) = 0, \ x(\pi,t) = 0$$

and the initial condition $x(z,0) = x_0(z)$. $k(x)$ now is expressed by an explicit nonlinear function of the state, $\beta_T(z)$ denotes the dimensionless heat of reaction
that is now an explicit function of the spatial coordinate $z$. The nominal values and expressions of the process parameters used in the presented simulations are:

$$k = 0.5 + 0.7/(x + 1), \quad x_0(z) = 0.5, \quad \beta_T(z) = 16[\cos(z) + 1], \quad \gamma = 2, \quad \text{and} \quad \beta_U = 2.$$

Two different actuator cases were investigated. Initially, a spatially distributed actuation with $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$ was considered and subsequently a point actuator $b(z) = \delta(z - 0.4\pi)$ was considered.

**Figure 3.7.** Open loop profile of the state of the diffusion-reaction process with a nonlinear spatial operator (Eq.3.16). b) Spatial profile of the dominant basis function obtained from the initial ensemble of the system of Eq.3.16.

Figure 3.7a presents the evolution of the PDE for $u(t) = 0$ from an initial condition of $x(z, t) = 0.5$. It can be observed that the operating steady-state $x(z, t) = 0$ is an unstable one and the system converges to a stable spatially nonuniform steady state. The control problem is again formulated as designing a feedback controller that stabilizes the rod temperature at the above open-loop unstable steady-state.
3.4.4 Numerical results

We initially generated an ensemble of 100 snapshots of the infinite dimensional system of Eq.3.16 with $u(t) = 0$, without performing an exhaustive sampling of the state-space of the PDE. This ensemble of snapshots has been presented in Figure 3.7a. Application of KLE along with the method of snapshots to this ensemble of snapshots resulted in a single dominant basis function which captured more than 99% of the energy embedded in the ensemble. This dominant basis function is presented in Figure 3.7b.

We then simulated the process using spatially distributed actuation $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$. Figure 3.8a presents the spatiotemporal profile of the rod temperature under closed-loop operation. We observe that the controller successfully stabilized the process at the open-loop unstable steady-state. Figure 3.8b presents the corresponding profile of the control action $u(t)$. We again note that the control action is a smooth function of time (with no discontinuities or chattering) and converges to zero as the control objective is achieved.

Owing to non-symmetric solution profile of Eq.3.16, the number of empirical basis functions required to capture 99% of energy of the ensemble changes from $m = 1$ to $m = 2$ at $t = 2.4$ (see Figure 3.9a). This can also be observed from Figure 3.9b; when $\xi$ increased to more than 1%, the basis of subspace $\mathbb{P}$ was appended. In addition, the basis functions were recomputed to account for the continuously changing ensemble of snapshots. This is demonstrated in Figures 3.8c
Figure 3.8. a) Closed-loop temperature profile of the system of Eq.3.16 using distributed actuation, \( b(z) = H(z - 0.3\pi) - H(z - 0.7\pi) \). b) Temporal profile of control action for the system Eq.3.16. c) Temporal profile of the first dominant basis function for the system of Eq.3.16. d) Temporal profile of the second dominant basis function for the system of Eq.3.16.

and 3.8d which present the temporal profiles of the two dominant basis functions, respectively.

To investigate the applicability of the method when many modes are excited, and to illustrate the region of attraction of the closed-loop system for this case, we simulated the system (under the proposed controller) from uniform initial conditions of \( x(z, 0) = 0.5 \) and \( x(z, 0) = -0.5 \), respectively.
Figure 3.9. a) Temporal profile of number of dominant empirical basis functions used in the reduced-order model (Eq.3.8) for the system of Eq.3.16 (distributed actuation, $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$). b) Temporal profiles of eigenvalues of $H$ and of the largest eigenvalue of $C_q$ for the system of Eq.3.16 using a spatially distributed actuator.

Figure 3.10. a) Closed-loop temperature profile of the system of Eq.3.16 for uniform initial condition of $x_0 = 0.5$ (distributed actuation, $b(z) = H(z - 0.3\pi) - H(z - 0.7\pi)$). b) Closed-loop temperature profile of the system of Eq.3.16 for a uniform initial condition of $x_0 = -0.5$ (distributed actuation)

Figure 3.10a presents the spatiotemporal profile of the rod temperature under closed-loop operation when starting from an initial condition of $x(z, 0) = 0.5$ and Figure 3.10b presents the spatiotemporal profile of the rod temperature when starting from an initial condition of $x(z, 0) = -0.5$. In both these simulations,
the slow and the fast modes of the system are initially excited. Furthermore, we illustrate the fact that a previously captured ensemble of snapshots (we used an ensemble from a previous simulation run) can be used by our method. Using the “old ensemble” and the proposed method we observe that the closed-loop system successfully converges to the open-loop unstable steady state in both cases.

Figure 3.11. a) Closed-loop temperature profile of the system of Eq.3.16 using point actuator, \( b(z) = \delta(z - 0.4\pi) \). b) Temporal profile of control action. c) Temporal profile of the first dominant basis function for the system of Eq.3.16 when using a point actuator. d) Temporal profile of the second dominant basis functions for the system described by Eq.3.16.

We subsequently used a point actuator \( b(z) = \delta(z - 0.4\pi) \) to simulate the above
process. Figure 3.11a presents the spatiotemporal profile of the rod temperature under closed-loop operation. We again observe that the controller was successful in stabilizing the process at open-loop unstable steady-state. The profile of the computed control action is presented in Figure 3.11b. We observe that it is a smooth function of time and that it converges to zero.

**Figure 3.12.** a) Temporal profile of number of dominant empirical basis functions used in the reduced-order model (Eq.3.8), to capture the desired 99% of energy of the ensemble for the system of Eq.3.16 (point actuation, \( b(z) = \delta(z - 0.4\pi) \)). b) Temporal profiles of eigenvalues of \( H \) and of the largest eigenvalue of \( c_q \).

Figure 3.12a presents the variation in the number of empirical basis functions employed to obtain the reduced-order process model. As more process measurements from the closed-loop operation were included in the ensemble while simultaneously old snapshots were removed, the percentage contribution \( \xi \), of the dominant eigenvalue of \( c_q \), increased to more than 1 (see Figure 3.12b) and consequently a new basis function joined the dominant eigenspace at \( t = 7.1 \). The dimensionality of the reduced-order ODE model was at that time instant increased from
$m = 1$ to $m = 2$. In addition, the basis functions were recomputed to account for continuously changing ensemble of snapshots. The temporal profiles of these dominant basis functions are presented in Figures 3.11c and 3.11d.

### 3.4.5 Effect of parametric uncertainty

To illustrate the effectiveness of the methodology under significant parametric uncertainty for the process described by Eq.3.16, we assumed a 10% uncertainty in the process parameter $\beta_T(z)$ which leads to a more unstable open-loop process behavior. The nominal value of $\beta_T(z)$ was set at $17.6[\cos(z) + 1]$ and all the other process parameters were at the nominal values presented in section 3.4.3. The controller (Eq.3.12), however was designed using $\beta_T(z) = 16[\cos(z) + 1]$. We subsequently used a point actuator $b(z) = \delta(z - 0.4\pi)$ to simulate the above process. Figure 3.12a presents the spatiotemporal profile of the rod temperature under closed-loop operation. We observe that the controller stabilized the process at the open-loop unstable steady state even in the presence of 10% uncertainty in one of its process parameters (which is directly coupled to the nonlinear terms in Eq.3.16). The computed control action was again a smooth function of time and no discontinuities or chattering appeared in its temporal profile. The temporal profile of this control action is presented in Figure 3.12b. Figure 3.13a, presents a temporal profile of the number of basis functions employed to obtain the reduced-order process model (Eq.3.8). A new basis function joined the dominant eigenspace at $t = 4.6$ when
Figure 3.13. a) Closed-loop temperature profile of the system of Eq.3.16 using point actuator $b(z) = \delta(z - 0.4\pi)$ for 10% uncertainty in $\beta_T$. b) Temporal profile of control action. c) Temporal profile of the first dominant basis function. d) Temporal profile of the second dominant basis function.

Figure 3.14. a) Temporal profile of number of basis functions of the system of Eq.3.16 using point actuator $b(z) = \delta(z - 0.4\pi)$ for 10% uncertainty in $\beta_T$. b) Temporal profiles of eigenvalues of $H$ and of the largest eigenvalue of $C_q$. 
the percentage contribution $\xi$, of the dominant eigenvalue of $c_q$, increased to more than 1 (see Figure 3.13b). The second basis function was removed from the dominant eigenspace at $t = 12.2$ when the corresponding eigenvalue of the first basis function was found to capture the desired percentage (99%) of energy of ensemble. Figures 3.12c and 3.12d present the temporal profiles of the two basis functions. It can be clearly observed that the basis functions were recomputed to account for the continuously changing ensemble of snapshots.
CHAPTER FOUR

OUTPUT FEEDBACK CONTROL

4.1 Introduction

We address the problem of tracking and stabilization of dissipative distributed parameter systems, by designing static output feedback controllers using adaptive proper orthogonal decomposition methodology (A POD). Initially, an ensemble of basis functions is constructed based on a relatively small data ensemble which is then recursively updated as additional process data becomes available periodically. The proposed A POD methodology relaxes the need for a representative ensemble of snapshots (in the sense that it contains the global dynamics of the process). An accurate reduced-order model (ROM) is constructed and periodically refined based on these updated basis functions. Using the ROM and continuous measurements
available from restricted number of sensors a static output feedback controller is
subsequently designed. This controller is successfully used to achieve the desired
control objective of stabilization and tracking in Kuramoto-Sivashinsky equation
and in FitzHugh-Nagumo equation.

4.2 Mathematical Formulation

We focus on designing output feedback controllers for parabolic PDE’s described
by the following state-space system

\[ \frac{\partial \bar{x}}{\partial t} = \mathcal{L}(\bar{x}) + b(z)u + f(\bar{x}), \]

\[ y_c = \int_{\Omega} c(z)\bar{x}dz, \]

\[ y_m = \int_{\Omega} s(z)\bar{x}dz. \]  

subject to mixed-type boundary conditions

\[ q \begin{pmatrix} \bar{x} \\ \frac{d\bar{x}}{d\eta} \\ \vdots \\ \frac{d^{n_0-1}\bar{x}}{d\eta^{n_0-1}} \end{pmatrix} = 0 \text{ on } \Gamma \]

and the following initial condition:

\[ \bar{x}(z, 0) = \bar{x}_0(z) \]
where \( \tilde{x}(z,t) \in \mathbb{R}^n \) denotes the vector of state variables, \( z = [z_1, z_2, z_3] \in \Omega \subset \mathbb{R}^3 \) is the vector of spatial coordinates, \( u \in \mathbb{R}^l \) denotes the vector of manipulated inputs; \( \Omega \) is the domain of definition of the process and \( \Gamma \) is its boundary. \( y_c \in \mathbb{R}^l \) and \( y_m \in \mathbb{R}^p \) denote the vector of controlled output and measured output respectively. \( \mathcal{L}(\tilde{x}) \) is an \( n_0 \) order dissipative, possibly nonlinear, spatial differential operator which includes higher-order spatial derivatives, \( f(\tilde{x}) \) is a nonlinear vector function, \( q(\tilde{x}; \frac{d\tilde{x}}{d\theta}, \ldots, \frac{d^{n_0-1}\tilde{x}}{d\theta^{n_0-1}}) \) is a nonlinear vector function which is assumed to be sufficiently smooth, \( \left. \frac{d\tilde{x}}{d\theta} \right|_{\Gamma} \) denotes the derivative in the direction perpendicular to the boundary and \( x_0(z) \) is the initial condition. \( b(z) \in \mathbb{R}^l \) is a known smooth matrix function of \( z \) of the form \([b_1(z) \ b_2(z) \ \cdots \ b_l(z)]\), where \( b_i(z) \) describes how the \( i^{th} \) control action \( u_i(t) \) is distributed in the spatial domain \( \Omega \), \( c(z) \) is a known vector function of \( z \) which is determined by the desired performance specifications in the domain \( \Omega \) and \( s(z) \) is a known smooth vector function of \( z \) which is determined by the shape (e.g., point or distributed) of the of the measurement sensor. We note that in the case of point actuation which influences the system at \( z_0 \) (i.e., \( b_i(z) \) is equal to \( \delta(z - z_0) \) where \( \delta(\cdot) \) is the standard Dirac function), we approximate the function \( \delta(z - z_0) \) by the finite value \( 1/2\hat{\epsilon} \) in the interval \( [z_0 - \hat{\epsilon}, z_0 + \hat{\epsilon}] \) (where \( \hat{\epsilon} \) is a small positive real number) and zero elsewhere in the domain \( \Omega \). A similar description is followed for point sensors.

To simplify the presentation, we now formulate the parabolic PDE system of Eq.4.1 as an infinite dimensional system in the Hilbert space \( \mathcal{H}(\Omega, \mathbb{R}^n) \) with \( \mathcal{H} \)
being the space of n-dimensional vector functions defined on \( \Omega \) that satisfy the boundary conditions in Eq.4.2. We define the inner product and norm in \( \mathcal{H} \) as follows:

\[
(\phi_1, \phi_2) = \int_{\Omega} \phi_1(z)\phi_2(z)dz, \quad ||\phi_1||_2 = (\phi_1, \phi_1)^{1/2}
\]

(4.4)

where \( \phi_1, \phi_2 \in \mathcal{H}[\Omega, \mathbb{R}^n] \). Defining the state function \( x \) on \( \mathcal{H}[\Omega, \mathbb{R}^n] \) as \( x(t) = \bar{x}(z, t), t > 0, z \in \Omega \), the operator \( \mathcal{L} \) in \( \mathcal{H}[\Omega, \mathbb{R}^n] \) as

\[
\mathcal{A}(x) = \mathcal{L}(\bar{x}), x \in D(\mathcal{A}) = \left\{ x \in \mathcal{H}[\Omega, \mathbb{R}^n]; q\left(\bar{x}, \frac{d\bar{x}}{d\eta}, \ldots, \frac{d^{n_0-1}\bar{x}}{d\eta^{n_0-1}}\right) = 0 \text{ on } \Gamma \right\}
\]

(4.5)

and the input, controlled output and measured output operators as \( \mathcal{B}u = bu, \mathcal{C}x = (c, x), \mathcal{S}x = (s, x) \) the system of Eq.4.1-4.3 takes the form

\[
\dot{x} = \mathcal{A}(x) + \mathcal{B}u + f(x), x(0) = x_0.
\]

(4.6)

\[
y_c = \mathcal{C}x, y_m = \mathcal{S}x.
\]

where \( f(x) = f(\bar{x}(z, t)) \) and \( x_0 = \bar{x}_0(z) \). We assume that the long-term dynamics of the above PDE system is finite dimensional and consequently the modes of the spatial operator, \( \mathcal{A}(x) \), can be partitioned into finite number of slow (dominant) and infinite number of stable and fast modes.

In the next section, we design output feedback controllers for Eq.4.1-4.3 using the following approach: the available off-line process data of Eq.4.1-4.3 is first utilized in APOD to compute the empirical basis functions. These empirical basis
functions are then employed in a Galerkin method framework to derive finite-dimensional ODE systems. The validity of these models is confined to a small region of the entire state space \((\mathcal{H}(\Omega))\) spanned by the available initial process data. During the closed loop process evolution APOD updates these models (using the closed loop process data) extending the region of validity of these models. These updated ODE systems will then be used in the synthesis of nonlinear static output feedback controllers to stabilize the closed loop system and force the output to follow a desired trajectory.

### 4.3 Galerkin’s method

In this section, we derive a finite-dimensional approximation (ROM) of the system of Eq.4.6 using the APOD computed empirical basis functions. Let \(\mathcal{H}_s, \mathcal{H}_f\) be the two subspaces of \(\mathcal{H}(\Omega)\). We assume that the subspaces \(\mathcal{H}_s\) and \(\mathcal{H}_f\) are appropriately defined such that the \(\mathcal{H}_f\) subspace includes all the fast evolving and stable process modes, while \(\mathcal{H}_s\) includes the slow evolving and possibly unstable ones. By definition such a separation exists for dissipative processes and a finite number of modes belongs in \(\mathcal{H}_s\), owing to the elliptic nature of the spatial differential operator. For our model and controller construction, we assume that the basis functions that capture the desired percentage of energy of the snapshot ensemble (denoted as \(\epsilon\) in the subsection 3.2.2) also form a complete basis of \(\mathcal{H}_s\). We thus define \(\mathcal{H}_s = \text{span}\{\phi_1, \phi_2, \ldots, \phi_m\}\) and \(\mathcal{H}_f = \mathcal{H} \setminus \mathcal{H}_s\). Clearly \(\mathcal{H} = \mathcal{H}_s \oplus \mathcal{H}_f\) and \(\mathcal{H}_f\)
is an infinite-dimensional subspace, while $\mathcal{H}_s$ is a finite-dimensional one.

Defining orthogonal projection operators $P$ and $Q$ the state $x \in \mathcal{H}(\Omega)$ can be decomposed as $x_s = P \in \mathcal{H}_s$ and $x_f = Q \in \mathcal{H}_f$. The state $x$ of the system of Eq.4.1 now can be expressed as:

$$x = x_s + x_f = P x + Q x$$

(4.7)

Applying $P$ and $Q$ to the system of Eq.4.6 and using the above decomposition of $x$ the system of Eq.4.6 can be equivalently expressed as:

$$\frac{dx_s}{dt} = A_s(x_s, x_f) + B_s u + f_s(x_s, x_f)$$

$$\frac{\partial x_f}{\partial t} = A_f(x_s, x_f) + B_f u + f_f(x_s, x_f)$$

(4.8)

$$y_c = C x_s + C x_f, y_m = S x_s + S x_f$$

$$x_s(0) = P x(0) = P x_0, x_f(0) = Q x(0) = Q x_0$$

where $A_s = P A(x_s + x_f), B_s = P B, f_s = P f, A_f = Q A(x_s + x_f), B_f = Q B$ and $f_f = Q f$ and the notation $\partial x_f/\partial t$ is used to denote that the state $x_f$ belongs in an infinite dimensional subspace ($\mathcal{H}_f$).

Neglecting the infinite dimensional fast and stable $x_f$ subsystem in Eq.4.8 the
following \( m \)-dimensional \( x_s \) subsystem is obtained:

\[
\frac{dx_s}{dt} = A_s(x_s, 0) + B_s u + f_s(x_s, 0)
\]

\[
y_{cs} = C x_s, \quad y_{ms} = S x_s
\]

\[x_f \equiv 0\]  

where the subscript \( s \) in \( y_{cs} \) and \( y_{ms} \) indicates that these outputs are associated with the \( x_s \) subsystem. The above system can be directly used for controller design employing standard control methods for ODEs [12].

### 4.4 Nonlinear static Output feedback control

In this section, we utilize the \( m \) dimensional subsystem in Eq.4.9 in designing nonlinear static output feedback controllers for the parabolic PDE system of the form given in Eq. 4.1-4.3. In our previous work [89], we designed state feedback controllers for parabolic PDE systems based on continuous availability of full state measurements of the process. The availability of such full state measurements is usually restricted in practice due to limited availability of sensors. In this work, we relax this assumption by requiring the availability of snapshots (for updating the ROM of Eq.4.9 using APOD) to be available periodically. The controller, however, is designed based on continuous point measurements available from restricted
number of sensors.

We consider the synthesis of static output feedback controller of the following form:

\[ u = \alpha(\tilde{x}_s) \]  \hspace{1cm} (4.10)

where \( \alpha(\cdot) \) is a smooth vector function and \( \tilde{x}_s \) are the estimates of the states in Eq.4.9. We estimate the states, \( \tilde{x}_s \), with a static observer using the limited number of continuous point measurements. The following assumption is needed in the design of the static observer.

**Assumption 1.** The number of available point measurements, \( p \), is equal to or greater than the number of modes of the \( x_s \) subsystem, \( m \), i.e. \( p \geq m \), and the inverse of the operator \( S \), \( S^\perp \), exists, so that \( \tilde{x}_s = S^\perp y_m \). When the finite-dimensional space, \( H_s \), is expressed using the empirical basis functions as basis functions the operator \( S^\perp \) becomes a matrix and is given by

\[ S^\perp = (S^TS)^{-1}S^T \]  \hspace{1cm} (4.11)

Note that the existence of inverse of matrix \( S \) depends on the location and shape of the measurement sensors; this amounts to properly choosing \( s(z) \) in Eq.4.1. We utilize the ROM in Eq.4.9 to construct an appropriate control Lyapunov function
[91], \( V(\tilde{x}_s) \), given by the following form.

\[
V(\tilde{x}_s) = \frac{1}{2} \tilde{x}_s^T \tilde{x}_s \tag{4.12}
\]

We note that \( V(\tilde{x}_s) \) is a smooth, positive definite and radially unbounded function (i.e., \( V(\tilde{x}_s) \to \infty \) only if \( \tilde{x}_s \to \infty \)).

The specific form of the static output feedback controller [92] is given by

\[
u = -k(\tilde{x}_s, c_0) L_{B_s} V(\tilde{x}_s) \tag{4.13}
\]

\[
k(\tilde{x}_s, c_0) = \begin{pmatrix}
c_o + \frac{L_{F_s} V(\tilde{x}_s) + \sqrt{(L_{F_s} V(\tilde{x}_s))^2 + (L_{B_s} V(\tilde{x}_s))^4}}{(L_{B_s} V(\tilde{x}_s))^2}, L_{B_s} V(\tilde{x}_s) \neq 0 \\
c_o, L_{B_s} V(\tilde{x}_s) = 0
\end{pmatrix}
\]

where \( F_s = A \tilde{x}_s + f_s(\tilde{x}_s, 0) \), \( L_{F_s} V = \frac{\partial V}{\partial \tilde{x}_s} F_s \), \( L_{B_s} V = \frac{\partial V}{\partial \tilde{x}_s} B_s \). The time derivative of \( V \) along the trajectories of the closed-loop slow subsystem of Eq.4.9

\[
\dot{V} = -c_0(L_{B_s} V(\tilde{x}_s))^2 - \sqrt{(L_{F_s} V(\tilde{x}_s))^2 + (L_{B_s} V(\tilde{x}_s))^4} \tag{4.14}
\]

Since \( \dot{V} \) is negative definite, the controller in Eq.4.13 stabilizes the parabolic PDE system of Eq.4.1 provided that the ROM provides an accurate description of process dynamics. A closed-loop diagram illustrating the different steps of the developed methodology is presented in Figure 4.1.

**Remark 5.** The positive parameter \( c_0 \) allows a certain degree of flexibility in shap-
ing the dynamic behavior of the closed-loop system. For example, a large value of $c_0$ will make $\dot{V}$ more negative and therefore generate a faster transient response.

Note that a positive value of $c_0$ is not necessary for stabilization.

**Remark 6.** When implementing the above controller in closed-loop process simulations, the numerical integrations could result in chattering-like behavior for the control input near origin. This problem is circumvented by adding a sufficiently small positive number $\eta$ to the $(L_{B_s}V(\bar{x}_s))^2$ in the denominator of Eq.4.13. The addition of this parameter obviously leads to some offset in the closed-loop response. However, this offset can be made arbitrarily small by choosing $\eta$ sufficiently small. A tradeoff thus exists between the smoothness of the control action (corresponds to large $\eta$) and the smaller offset in the closed-loop response (corresponds to using
small value for η).

**Remark 7.** Input constraints that arise due to the physical limitations inherent in the capacity of control actuators is beyond the scope of this work. However, we do note that the input constraints can be incorporated in the proposed methodology by employing appropriate feedback controllers [93, 94]. In future APOD implementations, input constraints will be explicitly taken into account in formulation of the controllers.

**Remark 8.** Note that the Lyapunov function, $V$, inherently changes with a change in the dimensionality of subspace, $H_s$. The controller is subsequently reconfigured based on the new Lyapunov function and the updated ROM (switching it to a different configuration). Thus, the stability aspect of the controller should be addressed using the hybrid systems theory. For the controller to be Lyapunov stable, apart from requiring $V$ to decrease when there is no change in the dimensionality of $H_s$, we impose an additional constraint: the value of $V$ at the beginning of each new interval (i.e., when a new closed-loop snapshot becomes available) should be lower than the value of $V$ at the beginning of the interval preceding it [94]. Note that a finite number of dimensionality changes will occur during process operation and at most a dimensionality change will occur as frequently as a new snapshot inclusion. The above controller in Eq.4.13 guarantees local exponential stability to the projected system only. The stability aspect of the process can then be evaluated using arguments from singular perturbations [25].
4.4.1 Recursive update of empirical basis functions and controller reconfiguration

To avoid the requirement of \textit{a priori} availability of a large ensemble of snapshots (as is usually required while using the POD scheme in section 2.1) during the computation of “relevant” empirical basis functions, we update the initially computed basis functions and the ROM in Eq.4.9 using our APOD methodology. The updated ROM is then used to reconfigure the controller (Eq.4.13). This step reassures that the reduced-order model captures new trends that appear when the process traverses through variable state space during closed-loop operation.

4.5 Applications

4.5.1 Kuramoto-Sivashinsky equation

In this section, we evaluate through computer simulations the ability of APOD and the proposed output feedback controller in stabilizing the Kuramoto-Sivashinsky equation (KSE) at the steady state $x(z, t) = 0$. KSE can adequately describe incipient instabilities arising in a variety of physico-chemical systems including falling liquid films [95], unstable flame fronts [96], interfacial instabilities between two viscous fluids [97]. Numerical studies [98, 99, 100] on the dynamics of KSE with periodic boundary conditions have revealed the existence of steady and periodic wave solutions, as well as chaotic behavior for very small values of $\nu$. 
We consider the integrated form of the controlled Kuramoto-Sivashinsky equation:

\[
\frac{\partial x}{\partial t} = -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x \frac{\partial x}{\partial z} + \sum_{i=1}^{t} b_i u_i(t) \tag{4.15}
\]

subject to the periodic boundary conditions:

\[
\frac{\partial^j x}{\partial z^j}(-\pi, t) = \frac{\partial^j x}{\partial z^j}(\pi, t), j = 0, \ldots, 3 \tag{4.16}
\]

and the initial condition

\[
x(z, 0) = x_0(z) \tag{4.17}
\]

where \( x \in \mathcal{H}([-\pi, \pi], \mathbb{R}) \) is the state of the system, \( z \) is the spatial coordinate, \( t \) is the time and \( 2\pi \) is the length of the spatial domain, \( u_i(t) \) is the \( i^{th} \) manipulated input. The tunable parameter \( \eta \) in the design of the controller (see remark 6) was set at \( \eta = 0.001 \). The spatial differential operator of system of Eq.4.1, for this problem is of the form:

\[
\mathcal{A}(x) = -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x \frac{\partial x}{\partial z}; \tag{4.18}
\]

\[
x \in D(\mathcal{A}) = \left\{ x \in \mathcal{H}([-\pi, \pi]; \mathbb{R}); \frac{\partial^j x}{\partial z^j}(-\pi, t) = \frac{\partial^j x}{\partial z^j}(\pi, t), j = 0, \ldots, 3 \right\}
\]

Six point control actuators were assumed to be available at the following locations \( L = [\pi/4, \pi/5, \pi/2, -\pi/2, -\pi/6, -\pi/4] \); the corresponding spatial distribution functions at these locations are \( b_i(z) = \delta(z - L_i); i = 1, \ldots, 6 \). Full state
measurements (snapshots) of the system are assumed to be available periodically at a period of \( t = 0.5 \). On the other hand, continuous point measurements were assumed to be available from 12 point measurement sensors placed uniformly across the domain of the process \((-\pi, \pi)\). The sensor shape distribution function, \( s(z) \) at these respective positions would be \( s_i(z) = \delta(z - P_i) \); \( i = 1, \ldots, 12 \), here \( P_i \) is the location of \( i^{th} \) sensor. In all the simulation runs, the system is assumed to be at a spatially non-uniform initial condition

\[
x_0 = 3\sin(z) - \cos(2z) - \sin(5z) + 2\cos(5z) \in \mathcal{H}([-\pi, \pi], \mathbb{R})
\]

In the first simulation run, we set \( \nu = 0.4 \). Figure 4.2(a), presents the open-loop evolution of KSE. It is observed that the process stabilizes to a non-uniform steady state. The control objective for this case would thus be to stabilize the process at \( x(z, t) = 0 \). Initially, an ensemble of 100 snapshots was collected by simulating the process with \( u(t) \equiv 0 \) till \( t = 2 \). Application of POD to this ensemble resulted in 5 basis functions that captured 99.99\% of the energy of the ensemble. These basis functions were then used in the computation of the ROM (Eq.4.9) for the above process. An output feedback controller of Eq.4.13 was subsequently designed, based upon the ROM, to achieve the desired control objective.

To present the effectiveness of APOD in capturing the new dominant trends that appear during closed-loop process evolution, we let the process evolve in closed-loop with a controller designed based on the initial ROM (computed using
Figure 4.2. a) Open-loop profile of the state of 4.15 with $\nu = 0.4$. b) $L_2$ norm of the closed-loop profile of the state of 4.15, using POD.
the initial ensemble as collected above and standard POD). As the initial ensemble spans a smaller subspace of the entire state space \( \mathcal{H}[-\pi, \pi] \) of the process, the ROM computed using this ensemble does not remain valid/accurate when the process traverses through different regions of the state space under the direction of the controller. As a result the \( L^2 \) norm of the closed-loop evolution profile of KSE presented in figure 4.2(b) does not converge to zero and the controller clearly fails to achieve the necessary control objective.

We then employ the APOD algorithm and once again use it to derive the ROM and design the controller based on ROM. Figure 4.3(a) presents the closed-loop evolution of KSE. The controller successfully stabilizes the process at the desired operating point of \( x(z, t) = 0 \), this is also observed in figure 4.3(b) wherein the \( L^2 \) norm of the process converges to zero. The positive definite Lyapunov function, \( V \), in figure 4.5 also converges uniformly to zero. Since the evolution of Lyapunov function \( V \) satisfies the two requirements required for closed-loop stability (discussed in remark 8) it is implied that the process is closed-loop stable. Figure 4.4(a) presents the temporal profile of this change in the dimensionality of the dominant eigenspace, \( \mathcal{H}_s \). Even when there was no change in the dimensionality of the dominant eigenspace, the basis functions were constantly updated to capture the new trends of the process that appear during closed-loop process evolution (section 3.2). Figure 4.4(b) presents the corresponding control action, \( u(t) \) that was utilized to achieve the above control objective. We note that the control action
Figure 4.3. a) Closed-loop profile of the state of 4.15, with $\nu = 0.4$, and b) $L_2$ norm of the closed-loop profile of the state of 4.15, using APOD.
is a smooth function of time, without any discontinuities and chattering. However, the control action may exhibit chattering if we opt for smaller values of the tunable parameter (remark 6).

The control objective, in this case, was successfully achieved as the dominant eigenspace of the process operator, $A(x)$, was constantly updated (by APOD) to account for continuous changes in the process due to closed-loop evolution. The dimensionality of the dominant eigenspace however, was increased when the basis functions do not capture 99.99% energy of the current ensemble of snapshots. On the other hand the dimensionality was reduced, when all the basis functions in the dominant eigenspace was not required to capture the 99.99% of energy.

Figure 4.6 presents a temporal plot of the inner product $(\phi_i, \phi_i(t_f))$, where $i = 1, \cdots, m$ and $\phi_i(t_f)$ is the profile of the $i^{th}$ empirical eigenfunction ($\phi_i$) at the end of process operation ($t = t_f$). Note that by construction $(\phi_i(t_f), \phi_i(t_f)) = 1$. As APOD recursively revises the empirical basis functions, during the closed-loop process operation, we observe that the value of $(\phi_i, \phi_i(t_f))$ converges to 1. The plot in figure 4.6 thus quantifies the temporal changes that are observed in the empirical basis functions, $\phi_i$, as compared to their final profile, $\phi_i(t_f)$. Furthermore, even widely different empirical basis functions converge smoothly with no oscillatory behavior. The temporal profile of the third eigenfunction, $\phi_3$, which exhibits a large variation in figure 4.6, is presented in figure 4.7. We observe a slow change in the spatial profile of the eigenfunction ($\phi_3$), as APOD updates the eigenfunction
Figure 4.4. a) Number of basis functions as a function of time. b) Temporal profile of control action, when using APOD.
Figure 4.5. Lyapunov function as a function of time, when using APOD.

Figure 4.6. Temporal profile of the inner product of empirical basis functions with respect to their respective final time spatial profile. \((\phi_i, \phi_i(t_f))\); \(t_f = 15\).
Figure 4.7. Temporal profile of the third eigenfunction, \( \phi_3 \).

to incorporate the effects of actuator on the closed-loop process dynamics.

In the second simulation run, we set \( \nu = 0.25 \). Figure 4.8(a), presents the traveling wave pattern observed in the open-loop evolution of KSE. The control objective for this case would be to shift the open-loop behavior and to stabilize the process at \( x(z,t) = 0 \).

As in the previous case, we collect an ensemble of 100 open-loop snapshots by simulating the process, Eq.4.15-4.17 till \( t = 2 \). Application of POD to these snapshots resulted in 7 basis functions which capture 99.999\% of the energy in the ensemble. An initial ROM was then computed using these ensemble of snapshots in Galerkin’s method (section 4). The computed ROM was later utilized in the design of an output feedback controller in Eq.4.13.
Figure 4.8. a) Open-loop profile of the state of 4.15 with $\nu = 0.25$. b) $L_2$ norm of the closed-loop profile of the state of 4.15, using POD.
Initially, the process was operated in closed-loop using the controller designed based on the initial ROM. Figure 4.8(b) presents the $L_2$ norm of the closed-loop profile of KSE; the $L_2$ norm of the process does not converge to zero and oscillates as the initial ROM does not capture the traveling wave behavior observed in the open-loop process. We can thus conclude that the controller designed in this case completely fails to achieve the desired closed-loop objective.

We then employed the proposed APOD algorithm and operate the process in closed-loop. Figure 4.9(a) presents the closed-loop evolution profile of KSE. It is observed that the controller successfully stabilizes the process at the desired operating point of $x(z, t) = 0$ and as a result the $L_2$ norm of the system, presented in figure 4.9(b), converges to zero.

Figure 4.10a presents the temporal profile of the dimensionality of the dominant eigenspace, $\mathcal{H}_s$. We also note that even when there was no change in the dimensionality of the dominant eigenspace the basis functions (and hence the ROM) were constantly updated to capture the new trends that become available during closed-loop process operation (section 3.2). Figure 4.10(b) presents the temporal profile of the control action. The small discontinuity observed at $t = 0.5$ is due to a change in the dimensionality of the dominant subspace $\mathcal{H}_s$, which is now accounted for by the controller. As the controller successfully achieves the desired control objective, we note that the positive definite Lyapunov function $V$ in figure 4.11 converges uniformly to zero. Note also that $V$ is always decreasing, even
Figure 4.9. a) Closed-loop profile of the state of Eq. 4.15, and b) $L_2$ norm of the closed-loop profile of the state of Eq. 4.15, using APOD.
Figure 4.10. a) Number of basis functions as a function of time. b) Temporal profile of control action, when using APOD.
Figure 4.11. Lyapunov function as a function of time, when using APOD.

Though there is a dimensionality change in $H_s$; thus, its evolution satisfies the two requirements for closed-loop stability as discussed in remark 8.

To present the temporal changes observed in the basis functions during the closed-loop process operation, we present a temporal plot of the inner product $(\phi_i, \phi_i(t_f))$ in figure 4.12. As observed in the case for $\nu = 0.4$, the value of $(\phi_i, \phi_i(t_f))$ converges to 1 as APOD revises the process basis functions during the closed-loop process operation. Furthermore, even widely different empirical basis functions converge smoothly with no oscillatory behavior. Figure 4.13 presents the temporal profile of the fifth process eigenfunction, $\phi_5$, that exhibited a large variation in figure 4.12. A slow change in the spatial profile of the eigenfunction $(\phi_5)$ is observed, as APOD updates the eigenfunction to incorporate the effects of
Figure 4.12. Temporal profile of the inner product of empirical basis functions with respect to their respective final time spatial profile. \((\phi_i, \phi_i(t_f)); t_f = 15\).

Figure 4.13. Temporal profile of the fifth eigenfunction, \(\phi_5\).
Figure 4.14. Temporal profile of CPU time for updating the ROM using APOD. The time for computing the ROM at each time step using the standard POD, online, was $9.5 \times 10^{-3}$ secs. Average acceleration obtained using APOD was 3.7 times compared to the standard POD approach.

To better demonstrate the computational acceleration of APOD, figure 4.14 presents the temporal profile of CPU time-spent per evaluation of APOD and subsequent reformulation of ROM. We note that the average CPU time was $2.7 \times 10^{-3}$ secs, which is appreciably lower than the corresponding values of CPU times using standard POD online (on the updated covariance matrix, $C_N$), namely, $9.5 \times 10^{-3}$ secs. Note that APOD is 370% faster compared to a standard POD implementation. CPU times are for a Pentium IV 3.02 GHz processor.
4.5.2 The FitzHugh-Nagumo Equation

In this section, we utilize APOD and the proposed output feedback controller to track a desired reference trajectory of FitzHugh-Nagumo (FHN) system. The FHN model, was originally proposed by FitzHugh [101] and Nagumo et al.[102] to describe nerve-impulse propagation. Variants of the FHN equation have been used to describe pattern formations of excitable media in reaction engineering [103] and physiology [104]. The FHN equation is described by the following pair of coupled PDE's

\[
\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial z^2} + v - w - v^3 + b(z)u(t)
\]

\[
\frac{\partial w}{\partial t} = \delta \frac{\partial^2 w}{\partial z^2} + \epsilon_p(v - p_1w - p_0)
\]

(4.19)

with boundary conditions:

\[
\frac{\partial v}{\partial z}\bigg|_0 = \frac{\partial v}{\partial z}\bigg|_L = 0
\]

\[
\frac{\partial w}{\partial z}\bigg|_0 = \frac{\partial w}{\partial z}\bigg|_L = 0
\]

(4.20)

and initial condition

\[
v(0, z) = v_0(z), w(0, z) = w_0(z)
\]

(4.21)

where \(v(t, z), w(t, z)\in \mathbb{R}\) are the system variables that are usually termed as ‘activator’ and ‘inhibitor’ respectively, \(u(t)\in \mathbb{R}^k\) is the array of control variables, \(t\) is the time, \(z\) is the spatial coordinate, \(b(z)\) is a row vector describing the control actuators, \(\epsilon_p\) is the relative time scale, \(\delta\) is the relative diffusivity of the two
species, $p_1$, $p_0$ are process parameters determining the local dynamics and $L$ is the length of the spatial domain. This set of equations, depending on parameter values, can exhibit multiple steady state solutions, as well as spatially non-uniform periodic solutions [105, 106]. The process parameter values used in this work are $L = 20$, $\delta = 4$, $\epsilon_p = 0.014$, $p_1 = 2$, $p_0 = -0.03$. The initial conditions used are $v_0 = 0.05\cos(\pi z/L)$ and $w_0 = 0.05\cos(\pi z/L)$. The linear spatial differential operator of system of Eq.4.1, for this problem is:

$$A(x) = \begin{pmatrix} \frac{\partial^2 v}{\partial z^2} & 0 \\ 0 & \delta \frac{\partial^2 w}{\partial z^2} \end{pmatrix}$$

We assume that three control actuators are available:

$$b(z) = \begin{bmatrix} g(z, 0.25L) & g(z, 0.5L) & g(z, 0.75L) \end{bmatrix}$$

where $g(z, z_0) = \exp(-0.3(z - z_0)^2)$; note that the actuator distribution functions extend over the entire spatial domain of the process. Full state measurements (snapshots) of the system are assumed to be available periodically at a period of $t = 0.5$. Continuous point measurements were assumed to be available from 12 point measurement sensors placed uniformly across the domain of the process $[0, L]$. Simulating Eq.4.19-4.21 with $u(t) = 0$ and initial conditions far from stable stationary states results in transients converging to a locally stable, spatiotemporally varying limit cycle orbit shown in Figures 4.15(a) and 4.15(b) for $v(t)$ and
Figure 4.15. Open-loop stable periodic orbit of: (a) v and (b) w (3D plots); (c) v and (d) w (greyscale plots).

The limit cycle observed in the open-loop plots arise through a hopf bifurcation from a branch of steady states on which our open-loop target unstable stationary state lies. This unstable state is denoted as \( v_{ss} \) and \( w_{ss} \) in figure 4.16(a) and figure 4.16(b) respectively. The control objective in this case is to drive and stabilize the process Eq.4.19-4.21 at the reference trajectories \( v_t \) and \( w_t \) starting from the initial stationary states \( v_{ss} \) and \( w_{ss} \) respectively.

For convenience the system to control will be represented in deviation form
Figure 4.16. Spatial profile of initial stationary state and the desired reference trajectory for (a) $v$ and (b) $w$. 
with respect to the reference trajectory.

\[
\begin{align*}
\frac{\partial \bar{v}}{\partial t} & = \frac{\partial^2 \bar{v}}{\partial z^2} + \bar{v} - \bar{w} - (\bar{v} + v_t)^3 + v_t^3 + b(z)u(t) \\
\frac{\partial \bar{w}}{\partial t} & = \delta \frac{\partial^2 \bar{w}}{\partial z^2} + \epsilon_p (\bar{v} - p_1 \bar{w} - p_0)
\end{align*}
\]

(4.22)

with boundary conditions:

\[
\begin{align*}
\frac{\partial \bar{v}}{\partial z} \bigg|_0 & = \frac{\partial \bar{v}}{\partial z} \bigg|_L = 0 \\
\frac{\partial \bar{w}}{\partial z} \bigg|_0 & = \frac{\partial \bar{w}}{\partial z} \bigg|_L = 0
\end{align*}
\]

(4.23)

and initial condition

\[
\bar{v}(0, z) = v_0(z) - v_t, \bar{w}(0, z) = w_0(z) - w_t
\]

(4.24)

where \( \bar{v} = v - v_t, \bar{w} = w - w_t \) and \( v_t, w_t \) are the target reference trajectories. The tracking of \( v(z, t) \) and \( w(z, t) \) in Eq.4.19-4.21 is equivalent to stabilizing the system of Eq.4.22 at zero.

Initially a data ensemble consisting of 100 open-loop snapshots was generated by simulating the PDE Eq.4.22-4.24 until time \( t = 25 \) using \( u(t) = 0 \). Simulations of the above PDE yielded two ensembles (for \( v(z) \) and \( w(z) \) fields). Application of POD to these ensembles resulted in 3 basis functions for \( v \) and \( w \), these basis functions capture 99.99% of the energy embedded in their respective ensembles. Figure 4.17(a) & 4.17(b) presents the closed-loop process evolution of \( v(z, t) \) and \( w(z, t) \).
Figure 4.17. Closed-loop profile of the state of 4.19 (a) \( v \) and (b) \( w \), using APOD.
respectively. We observe that the controller successfully stabilizes the process (ex-
pressed in deviation variables) Eq.4.22-4.24 at the $\bar{v}(z,t) = 0$ and $\bar{w}(z,t) = 0$. In
other words the controller successfully drives the process Eq.4.19-4.21 to track the
desired reference trajectories $v_t$ and $w_t$ respectively. Note that in these simulations
the controller was switched on at $t = 25$ (i.e, at the end of data collection for the
formulation of the initial ensemble).

The temporal profile of the dimensionality of the dominant eigenspace is pre-
sented in figure 4.18(a), the dimensionality of the dominant eigenspace was in-
creased or decreased as demanded by APOD. Figure 4.18(b) presents the temporal
profile of the control action. We note that the control action is observed to be a
smooth function of time with no discontinuities or chattering.

Figure 4.19(a) and 4.19(b), presents the temporal profile of the $L_2$ norm of
$\bar{v}(z,t)$ and $\bar{w}(z,t)$. The $L_2$ norm converges very close to zero. The small offset
observed especially in $L_2$ norm of the $\bar{w}(z,t)$ can be eliminated by choosing a
sufficiently small value for $\eta$ (remark 6). Figure 4.20 presents the temporal profile
of the Lyapunov function $V$, it is observed that $V$ uniformly converges to zero
as the desired control objective is achieved. Since the evolution of $V$ satisfies
the two requirements required for closed-loop stability (remark 8); it is implied
that the process is closed-loop stable. Figures 4.21(a) and 4.21(b) compare the
closed-loop spatial profiles of the system variables $v$ and $w$ at the end of the
process operation against the desired reference trajectories. It is observed that
Figure 4.18. a) Number of basis functions as a function of time, and b) temporal profile of control action.
Figure 4.19. a) $L_2$ norm of the closed-loop profile of (a) $v(z,t)$ and (b) $w(z,t)$. 
the output feedback controller successfully drives the process Eq. 4.19-4.21 to the desired reference trajectories.

To better demonstrate the computational acceleration of APOD, figure 4.22 presents the temporal profile of CPU time-spent per two evaluations of APOD (for updating $\phi_v$ and $\phi_w$ respectively) and subsequent reformulation of ROM. We note that the average CPU time was $5 \times 10^{-3}$ secs, which is again appreciably lower than the corresponding values of CPU times using standard POD online (on the updated covariance matrix, $C_N$), namely, $18 \times 10^{-3}$ secs. We again observe that APOD is 360% faster compared to a straightforward POD implementation.
Figure 4.21. Spatial profiles of desired trajectory and obtained closed loop profile a) $v$ and b) $w$. 
Figure 4.22. Temporal profile of CPU time for updating the ROM using APOD. The time for computing the ROM using the standard POD, online, was $18 \times 10^{-3}$ secs. Average acceleration obtained using APOD was 3.6 times compared to the standard POD approach.

4.6 Conclusions

This chapter presents the output feedback controller design of dissipative PDE systems using adaptive proper orthogonal decomposition (APOD) methodology. Initially, the available off-line process data was employed in APOD to compute the empirical basis functions, then the empirical functions were used as basis functions in a Galerkin’s method framework to construct the reduced order model (ROM) for the system that accurately describes the dominant dynamics of the system. This ROM was then utilized in the synthesis of the output feedback controller to achieve the desired control objective. The empirical basis functions (and hence
the ROM) were updated using closed loop process data in APOD, thus ensuring the validity of the basis functions and ROM in capturing the new trends that appear during closed process evolution. We used two representative examples of dissipative PDEs, a system of coupled nonlinear one-dimensional PDEs (the FitzHugh-Nagumo equations), widely used to describe the formation of patterns in reacting and biological systems, and the Kuramoto-Sivashinsky equation, a model that describes incipient instabilities in a variety of physical and chemical systems, to demonstrate the implementation and evaluate the effectiveness of APOD along with the proposed output feedback controller.
5.1 Introduction

In this chapter, we employ adaptive reduced order models (ROMs) in the design of model predictive controllers for stabilization of processes that are mathematically expressed as parabolic partial differential equation (PDE) systems. Initially, we construct a locally valid ROM of the PDE system employing the basis functions computed by applying an adaptive model reduction methodology called APOD on a small data ensemble. This ROM is then utilized in the design of model predictive controllers (MPC) under constraints on the control action. As periodic closed-loop process data becomes available (during closed-loop operation under the constructed MPC), we recursively update the ROM by employing our com-
putationally efficient adaptive model reduction methodology thus extending the validity of ROM over the current operating region. The effects of employing the adaptive methodology on performance of MPC is studied. The design of such MPC controllers is illustrated by employing the methodology on numerical simulations.

5.2 Mathematical preliminaries

We focus on designing model predictive controllers for nonlinear parabolic PDEs of the following form:

$$\frac{\partial \bar{x}}{\partial t} = \mathcal{L}(\bar{x}) + b(z)u + f(\bar{x}).$$  \hfill (5.1)

subject to the following boundary and initial conditions:

$$q \left( \bar{x}, \frac{d\bar{x}}{d\eta}, \ldots, \frac{d^{n-1}\bar{x}}{d\eta^{n-1}} \right) = 0 \text{ on } \Gamma$$  \hfill (5.2)

$$\bar{x}(z, 0) = \bar{x}_0(z)$$

and the following input constraints

$$u_{\text{min}} \leq u \leq u_{\text{max}}$$  \hfill (5.3)

where $\bar{x}(z, t) \in \mathbb{R}^n$ denotes the vector of state variables, $z = [z_1, z_2, z_3] \in \Omega \subset \mathbb{R}^3$ is the vector of spatial coordinates, $u \in \mathbb{R}^l$ denotes the vector of manipulated inputs; $u_{\text{min}}$ and $u_{\text{max}}$ denotes the lower and upper bounds on the manipulated
input, $\Omega$ is the domain of definition of the process and $\Gamma$ is its boundary. $\mathcal{L}(\bar{x})$ is a dissipative, possibly nonlinear, spatial differential operator, $f(\bar{x})$ is a nonlinear vector function, $q(\bar{x}, \frac{dx}{d\eta}, \ldots, \frac{d^{n_\alpha-1}x}{d\eta^{n_\alpha-1}})$ is a nonlinear vector function which is assumed to be sufficiently smooth, $\frac{dx}{d\eta}|_\Gamma$ denotes the derivative in the direction perpendicular to the boundary and $\bar{x}_0(z)$ is the initial condition. $b(z) \in \mathbb{R}^l$ is a known matrix function of $z$ of the form $[b_1(z) \ b_2(z) \ \cdots \ b_l(z)]$, where $b_i(z)$ describes how the $i^{th}$ control action $u_i(t)$ is distributed in the spatial domain $\Omega$.

The parabolic PDE system of Eq. 5.1 can be recast as an infinite dimensional system in an appropriate Hilbert space $\mathcal{H}(\Omega, \mathbb{R}^n)$, $\mathcal{H}$ being the space of $n$-dimensional vector functions defined on $\Omega$ that satisfy the boundary conditions in Eq. 5.2.

$$\mathcal{H} = \left\{ x \in L_2[\Omega, \mathbb{R}^n]; \ q\left(x, \frac{dx}{d\eta}, \ldots, \frac{d^{n_\alpha-1}x}{d\eta^{n_\alpha-1}}\right) = 0 \text{ on } \Gamma \right\}$$

(5.4)

We define the inner product and norm in $\mathcal{H}$ as follows:

$$\langle \phi_1, \phi_2 \rangle = \int_\Omega \phi_1^*(z)\phi_2(z)dz, \ ||\phi_1||_2 = (\phi_1, \phi_1)^{1/2}$$

(5.5)

where $\phi_1, \phi_2 \in \mathcal{H}[\Omega, \mathbb{R}^n]$. Defining the state function $x$ on $\mathcal{H}$ as $x(t) = \bar{x}(z,t), t > 0, z \in \Omega$, the operator $\mathcal{A}$ in $\mathcal{H}[\Omega, \mathbb{R}^n]$ as $\mathcal{A}(x) = \mathcal{L}(\bar{x})$, the input, controlled output and measured output operators as $Bu = bu$, the system of Eqs. 5.1-5.2 acquires
the following form in the Hilbert space, $\mathcal{H}$:

$$\dot{x} = A(x) + Bu + f(x), \ x(0) = x_0$$

$$u_{min} \leq u \leq u_{max}. \quad (5.6)$$

where $f(x) = f(\bar{x}(z,t))$ and $x_0 = \bar{x}_0(z)$.

**Assumption 2.** The long-term dynamics of the above PDE system is finite dimensional in a sense that the state $x$ of the above system can be accurately described by a finite number of degrees of freedom. Thus, in principle, a finite number of basis functions of $\mathcal{H}$ can accurately approximate the long term behavior of $x$. Furthermore, the state $x$ can be partitioned into finite number of slow and possibly unstable modes and an infinite number of stable and fast modes.

### 5.2.1 Derivation of ROM

We derive a locally valid ROM of the system of Eq. 5.6 using the obtained local basis functions (section 2.2.1). Applying orthogonal projection operators $\mathcal{P}$ and $\mathcal{Q}$ (section 4.3) to the system of Eq. 5.6 and using the singular perturbation arguments for infinite dimensional systems [25] the following $m$-dimensional $x_s$
subsystem is obtained:

\[
\frac{dx_s}{dt} = A_s(x_s, 0) + B_s u + f_s(x_s, 0)
\]

\[x_f \equiv 0\] (5.7)

Under the assumptions already stated, the above finite dimensional system is an accurate approximation of the dominant dynamics of the infinite dimensional system of Eq. 5.6.

As the above ROM has a small range of validity (Eq. 5.7) it may may not remain valid during the course of closed-loop process evolution. To avoid this situation, we periodically update the ROM by updating the basis functions, \(\phi\), using APOD (section 2.2) thus extending its validity over the current operational space.

### 5.3 Design of model predictive controllers

In this section, we utilize the updated ROMs in the synthesis of model predictive controllers to stabilize the system given by Eqs. 5.1-5.2. We assume that the full state measurements from the process becomes available periodically. The control law is obtained by formulating and solving the following open-loop optimal control problem with a receding control horizon.
\[ u^o = \arg \min_u J(x_s, u) \]

\[ \text{s.t.} \]

\[ \frac{dx_s}{dt} = A_s(x_s, 0) + B_s u + f_s(x_s, 0), \quad x_s(t_0) = P x_0, \]

\[ u \in \mathcal{U}, \]

We employed a standard performance index \( J \)

\[ J(x_s, u) = \int_t^{t+T_p} (q_s ||x_s||^2 + u^T R u) d\tau + g_s(x_s(t+T_p))^2 \tag{5.9} \]

where \( T_p \) denotes the prediction horizon, \( q_s \) & \( g_s \) are strict positive numbers. \( \mathcal{U} \) denotes the set of admissible input values which is assumed to be compact. Thus \( \mathcal{U} = \{ u(t) \in \mathbb{R}^l : u_{i_{\text{min}}} \leq u(t) \leq u_{i_{\text{max}}}, i = 1, \ldots, l \} \), where \( u_{i_{\text{min}}} \) & \( u_{i_{\text{max}}} \) are the maximum & minimum bounds on \( u_i \).

We discretize the temporal domain into \( m_t \) intervals with a step length of \( \delta t_i = t_i - t_{i-1}, \forall i = 1, \ldots, m_t \). The control action \( u(t) \) is then expressed as a series of the form

\[ u(t) = \sum_{i=0}^{m_t-1} u_{i+1} [H(t - t_i) - H(t_{i+1} - t)] \tag{5.10} \]

where \( H(\cdot) \) is the standard Heaviside function. Using control vector parametrization (CVP) methodology and the above discretized form of control vector, we reformulate the dynamic optimization problem in Eq.5.8 as an algebraic nonlinear one. CVP involves the temporal discretization of the control vector only, and the
solution of the dynamic equality constraints through direct integration, keeping track of constraint violations during the process evolution [107]. Solution of the optimization problem yields an optimal input sequence $u^o$ at each sampling instance and only the first input vector in the sequence is actually implemented. Subsequently, the prediction horizon is moved forward by one time-step, and the above problem is re-solved using new process measurements.

5.4 Applications

5.4.1 Diffusion-reaction processes

In this section, we apply the proposed adaptive model reduction and control methodology to a typical diffusion-reaction process that exhibits nonlinear dynamic behavior. Specifically, we consider an elementary exothermic reaction $A \rightarrow B$ taking place on a thin catalytic rod as described in section 3.4.3. A spatially distributed actuator with $b(z) = H(z - 0.3\pi) - H(z - 0.6\pi)$, where $H(\cdot)$ again denotes the standard Heaviside function, was considered. The nominal values and expressions of the process parameters used in the presented simulations are: $k = 0.5 + 0.7/(x + 1)$, $x_0(z) = 0.5$, $\beta_T(z) = 13[\cos(z) + 1]$, $\gamma = 2$, and $\beta_U = 2$. Figure 5.1 presents the evolution of the PDE for $u(t) = 0$ from an initial condition of $x(z,0) = 0.5$. It is observed that the system evolves away from the above steady-state to another steady-state characterized by a non-uniform distri-
distribution of temperature across the rod. Hence, we conclude that the steady-state $x(z,t) = 0$ is an unstable one. As a result we formulate the control problem as the one that stabilizes the rod temperature around the spatially open-loop unstable steady-state. We initially collected an ensemble of 100 open-loop snapshots of the system of Eq.3.16 with $u(t) = 0$, without performing an exhaustive sampling of the state-space of the PDE. This ensemble of snapshots is presented in Figure 5.1. Applying the APOD step resulted in a single dominant basis function which captured more then 99% of the energy embedded in the ensemble.

We utilized the MPC formulation presented in section IV to stabilize the unsteady steady-state $x(z,t) = 0$. Mathematically, the optimization problem, for

\[ \text{Figure 5.1.} \] Open-loop profile of the state of the diffusion-reaction process (Eq.3.16).
computing the optimal control law, is formulated as

\[
u^o = \arg \min_u \int_t^{t+T_p} (q_s |x_s|^2 + u^T Ru) d\tau + g_s(x_s(t+T_p))^2\]

s.t.

\[
\frac{dx_s}{dt} = A_s(x_s, 0) + B_s u + f_s(x_s, 0), \quad x_s(t_0) = P x_0, \quad u \in \mathcal{U},
\]

where

\[
A_s = \int_0^\pi \left( \frac{\partial}{\partial x} k(x) \frac{\partial}{\partial z} - \beta_U x \right) \phi(z) dz
\]

\[
f_s(x_s) = \int_0^\pi \beta_T (e^{-\gamma/(1+x)} - e^{-\gamma}) \phi(z) dz
\]

\[
B_s = \beta_U \int_0^\pi b(z) \phi(z) dz
\]

A prediction horizon of \(T_p = 6\), with time step \(\delta t = 0.25\) and penalty parameters \(q_s = 100, R = 20I^{1 \times 1}, g_s = 300\) were used. \(\mathcal{U} = \{u(t) \in \mathbb{R}^1 : -0.6 \leq u(t) \leq 0.6\}\).

The control vector was discretized using \(m_c = 4\) intervals over a control horizon of \(T_c = 4\) and the resulting discretized control vector was utilized for the solution of the above optimization problem using CVP methodology. The optimization problem was solved using the MATLAB subroutine fmincon. The first input vector from the obtained optimal control sequence \(u_o\) was actually implemented in the plant Eq.3.16. Subsequently, the prediction horizon was moved forward by one time-step, the ROM was updated by using the obtained closed-loop measurement
& APOD methodology and the above optimization problem was re-solved using the new ROM & process measurements. Figure 5.2(a) presents the closed-loop profile of the state $x(z,t)$; it is clear that the controller successfully drives the process to the spatially uniform steady state of $x(z,t) = 0$. Figure 5.2(b) presents the corresponding control action that drives the process to the steady state. We note that the computed control action stays within the constraint set $U$. Figure 5.3(a) presents the temporal profile of the objective function $J$ (computed at discrete time-steps of $t = 0.25$). We note that the value of the objective function converges to zero upon the achievement of the desired control objective.

As more process measurements from the closed-loop operation were included in the ensemble while simultaneously old snapshots were removed, a new basis function became dominant and joined the dominant eigenspace $\mathbb{P}$ at $t = 0.75$. Consequently, the dimensionality of the ROM of Eq.3.16 increased from $m = 1$ to $m = 2$; this increase in dimensionality is presented in figure 5.3(b). Following the update of ROM the constraints of the MPC were revised and the optimization problem Eq.5.16 was re-solved using the new ROM & new process measurements.

To test the robustness of the methodology, we varied the process parameters, initial conditions and actuator distribution functions. In all the cases the process successfully converged to desired steady state. Figure 5.4 presents the computed control action for a $-12.5\%$ variation in $\beta_T$ and $-20\%$ variation in initial condition. We note that in the both cases the system is driven to the spatially uniform steady
Figure 5.2. a) Closed-loop profile of the state of the diffusion-reaction process (Eq. 3.16). b) Temporal profile of the manipulated control action with $b(z) = H(z - 0.3\pi) - H(z - 0.6\pi)$

state $x(z, t) = 0$ faster and with less control action, since in these two cases the effect of destabilizing nonlinearity is smaller in the first case of variation in $\beta_T$ and in the second case the system starts closer to the spatially uniform steady state.
5.4.2 Wave suppression

In this section, we illustrate the proposed methodology on Kuramoto-Sivashinsky equation (KSE) with distributed actuation.

\[
\frac{\partial x}{\partial t} = -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x \frac{\partial x}{\partial z} + \sum_{i=1}^{l} b_i u_i(t)
\]  (5.12)
Figure 5.4. Temporal profile of the manipulated control action for nominal parameters and for $-20\%$ variation of the initial condition and for a $-12.5\%$ variation of $\beta_T$.

subject to the periodic boundary conditions:

$$\frac{\partial^j x}{\partial z^j}(-\pi, t) = \frac{\partial^j x}{\partial z^j}(\pi, t), j = 0, \ldots, 3$$

(5.13)

and the initial condition

$$x(z, 0) = x_0(z)$$

(5.14)

where the $x$ is the state of the system $x \in \mathcal{H}([-\pi, \pi], \mathbb{R})$ and is considered to be sufficiently smooth (i.e., differentiable 4 times), $z$ is the spatial coordinate, $t$ is the time and $u_i(t)$ is the $i^{th}$ manipulated input. The spatial differential operator of
system of Eq. 5.1, for this problem is of the form:

\[
\mathbf{A}(x) = -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x \frac{\partial x}{\partial z}
\]
\[
\left\{ x \in \mathcal{H}([\pi, \pi]; \mathbb{R}); \frac{\partial^j x}{\partial z^j}(-\pi) = \frac{\partial^j x}{\partial z^j}(\pi), j = 0, \ldots, 3 \right\}
\] (5.15)

where the length of the spatial domain is \(2\pi\) and the diffusion parameter in Eq. 5.12 was set as \(\nu = 0.25\). Three control actuators were assumed to be available at the following locations \(L = [0.4\pi, 0.6\pi, -0.3\pi]\); the corresponding spatial distribution functions at these locations are \(b_i(z) = \delta(z - L_i); i = 1, \ldots, 3\). In these simulation runs, the following spatially non-uniform initial condition was considered:

\[
x_0 = \sum_{i=1}^{4} \sin(iz)
\]

Figure 5.4.2, presents the wave pattern observed in the open-loop evolution of KSE wherein we observe the formation of persistent waves. Thus the control objective was set to stabilize the process in a optimal way in the neighborhood of the spatially uniform steady state \(x(z, t) = 0\).

For the design of controller, an initial ensemble of 100 open-loop snapshots \((N = 100)\) was collected by simulating the process with \(u(t) \equiv 0\) till \(t = 2\). Note that no exhaustive sampling of state space as performed in [24] was not required. Application of POD to this ensemble resulted in \(m = 3\) basis functions that captured 99.99% of the energy of the ensemble. These basis functions were
then employed in the computation of the local ROM (Eq. 5.7) for the above process.

We then utilized the MPC formulation presented in section IV to stabilize the unsteady steady-state $x(z, t) = 0$. The optimization problem for computing the optimal control law was formulated as:

$$u^o = \arg \min_u \int_{t}^{t+T_p} [q_s |x_s|^2 + u^T R u] d\tau + g_s (x_s(t + T_p))^2$$

s.t.

$$\frac{dx_s}{dt} = A_s(x_s, 0) + B_s u, \quad x_s(t_0) = P x_0,$$

$$u \in \mathcal{U},$$

\textbf{Figure 5.5.} Open-loop profile of the state of the diffusion-reaction process (Eq.5.12).
where

\[ A_s = \int_{-\pi}^{\pi} \left( -\nu \frac{\partial^4 x}{\partial z^4} - \frac{\partial^2 x}{\partial z^2} - x \frac{\partial x}{\partial z} \right) \phi(z) dz \]
\[ B_s = \int_{-\pi}^{\pi} b(z) \phi(z) dz \]

A prediction horizon of \( T_p = 7 \), with the time step \( \delta t = 0.2 \) and \( q_s = 100 \), \( R = 20I^{3\times3} \), \( g_s = 300 \) were used. \( \mathcal{U} = \{ u(t) \in \mathbb{R}^3 : -3 \leq u_i(t) \leq 3, i = 1 \cdots 3 \} \).

Figure 5.6(a) presents the closed-loop profile of the state \( x(z,t) \); the controller successfully drives the process to a neighborhood around the spatially uniform steady state of \( x(z,t) = 0 \) in a finite time. Figure 5.6(b) presents the corresponding control action that drives the process to the steady state. We note that the computed control action stays within the constraint set \( \mathcal{U} \). Also the value of the objective function \( J \) converges to zero upon achieving the control objective (Figure 5.7(a)).

The change in the dimensionality of the ROM due to APOD is presented in figure 5.7(b). We note that APOD increases the dimensionality of ROM from 3 to 4 at \( t = 0.6 \) as more basis functions were needed to capture the emerging new trends of the closed-loop process. To maintain the accuracy of the basis functions and the ROM we updated the ROM using the step 3 of the APOD methodology (section 2.2).

We also stabilized the above KSE process using a lower energy criterion of 99%. For this case APOD was found to have a superior performance compared
Figure 5.6. a) Closed-loop profile of the state of the diffusion-reaction process (Eq. 5.12). b) Temporal profile of the manipulated control action with 3 control actuators.

to just utilizing POD methodology. For the reasons of brevity we do not present these results in this manuscript. Further reduction of \( \epsilon \) lead to deterioration of performance. However the the MPC controller designed based on APOD still stabilized the process while a POD designed one failed.
5.5 Conclusions

We designed model predictive controllers for distributed processes using adaptive reduced order models. Initially a ROM was formulated using the available process data in the POD methodology. An optimization problem was then solved
(with a receding control horizon) to compute the optimal control action. We updated the ROM, upon the availability of closed-loop measurements from the process (from implementing the computed control actuation on the plant) thus eliminating the requirement for formulation of a completely excited data ensemble as required for controllers designed based on POD methodology. The effectiveness of the methodology was successfully demonstrated using two process examples at open-loop unsteady operating points. The designed MPC controller using adaptive ROM successfully stabilized the process.
CHAPTER SIX

OUTPUT FEEDBACK CONTROL
WITH PARTIAL SENSOR INFORMATION

6.1 Introduction

We address the problem of stabilization of processes that are mathematically expressed as parabolic partial differential equation systems when using partial state measurements. We initially reconstruct the partial measurements using gappy reconstruction technique. These reconstructed measurements are then utilized for the derivation and update of reduced order models (ROM) using our adapt-
tive proper orthogonal decomposition (APOD) methodology. The use of APOD methodology allows the development and on-demand update of a locally accurate low-dimensional ROM thus resulting in a computationally efficient alternative to using a large dimensional ROM with global validity. Based on the low-dimensional ROM and continuous measurements available from restricted number of sensors a static output feedback controller is subsequently designed. The design of such controllers is illustrated on an unstable process modeled by the Kuramoto-Sivashinsky equation; the designed controller successfully stabilizes the process in an efficient manner.

6.2 Mathematical Formulation

We focus on designing output feedback controllers for parabolic PDEs described by the following state-space system

\[
\frac{\partial \bar{x}}{\partial t} = \mathcal{L}(\bar{x}) + b(z)u + f(\bar{x}),
\]

\[
y_c = \int_{\Omega} c(z)\bar{x}dz,
\]

\[
y_m = \int_{\Omega} s_m(z)\bar{x}dz,
\]

\[
y_r = s_r(z,t)\bar{x}.
\]

subject to mixed-type boundary conditions

\[
q \left( \bar{x}, \frac{d\bar{x}}{d\eta}, \ldots, \frac{d^{n_o-1}\bar{x}}{d\eta^{n_o-1}} \right) = 0 \text{ on } \Gamma
\]
and the following initial condition:

\[ \bar{x}(z, 0) = \bar{x}_0(z) \]  

where \( \bar{x}(z, t) \in \mathbb{R}^n \) denotes the vector of state variables, \( z = [z_1, z_2, z_3] \in \Omega \subset \mathbb{R}^3 \) is the vector of spatial coordinates, \( u \in \mathbb{R}^l \) denotes the vector of manipulated inputs; \( \Omega \) is the domain of definition of the process and \( \Gamma \) is its boundary. \( y_c \in \mathbb{R}^l \) denotes the vector of controlled outputs and \( y_m \in \mathbb{R}^p \) and \( y_r \) denote the vectors of measured outputs for control and state reconstruction respectively. \( L(\bar{x}) \) is an \( n_0 \) order dissipative, possibly nonlinear, self adjoint spatial differential operator which includes higher-order spatial derivatives, \( f(\bar{x}) \) is a nonlinear vector function, \( q(\bar{x}, \frac{d\bar{x}}{d\eta}, \ldots, \frac{d^{n_0-1}\bar{x}}{d\eta^{n_0-1}}) \) is a nonlinear vector function which is assumed to be sufficiently smooth, \( \frac{d\bar{x}}{d\eta} \bigg|_{\Gamma} \) denotes the derivative in the direction perpendicular to the boundary and \( \bar{x}_0(z) \) is the initial condition. \( b(z) \in \mathbb{R}^l \) is a known matrix function of \( z \) of the form \( [b_1(z) \ b_2(z) \ \cdots \ b_l(z)] \), where \( b_i(z) \) describes how the \( i^{th} \) control action \( u_i(t) \) is distributed in the spatial domain \( \Omega \), \( c(z) \) is a known vector function of \( z \) which is determined by the desired performance specifications in the domain \( \Omega \). We note that in the case of point actuation which influences the system at \( z_0 \) (i.e., \( b_i(z) \) is equal to \( \delta(z - z_0) \) where \( \delta(\cdot) \) is the standard Dirac function), we approximate function \( \delta(z - z_0) \) by a boxcar function with a finite value of \( 1/2\hat{\epsilon} \) in the interval \([z_0 - \hat{\epsilon}, z_0 + \hat{\epsilon}]\) (where \( \hat{\epsilon} \) is a small positive real number) and zero elsewhere in the domain \( \Omega \).
An important component of this work is the sensors used. We assume the availability of two sets of measurements: (i) periodic (partial) snapshot measurements, $y_r$, and (ii) continuous measurements, $y_m$. The periodic measurements, $y_r$, will be utilized during the formulation & update of the ROM for the above equation Eq. 6.1 and may provide a complete or partial information of the state profile. The continuous measurements, $y_m$, along with the updated ROM are utilized in the design and operation of the designed output feedback controller. Note that $y_r$ is a profile while $y_m$ is a vector variable. $s_r(z,t)$ and $s_m(z)$ are the sensor shape functions corresponding to $y_r$ and $y_m$ respectively. Without loss of generality, we consider the one dimensional case; the sensor shape distribution for the periodic measurements, $y_r$, is expressed as a finite sum of boxcar functions

$$s_r(z,t_i) = \sum_{j=1}^{N_f} B(z; b_j(t_i), c_j(t_i))$$

(6.4)

$$B(z; b_j(t_i), c_j(t_i)) = H(z - b_j(t_i)) - H(z - c_j(t_i))$$

where $b_j(t_i) \& c_j(t_i)$ represents the boundaries of the boxcar function at time instant $t_i$, $N_f$ is a finite number and $H(\cdot)$ is the standard Heaviside function. We assume that over time the whole spatial domain is “covered” by the sensor measurements, $y_r$. On the other hand, for the continuous point measurements, $y_m$, we consider the most restrictive case wherein the corresponding sensor shape function is given by the dirac function $s_m(z) = \delta(z - z_i)$.

The parabolic PDE system of Eq. 6.1 can be recast as an infinite dimensional
system in the Hilbert space $\mathcal{H}(\Omega, \mathbb{R}^n)$ with $\mathcal{H}$ being the space of n-dimensional vector functions defined on $\Omega$ that satisfy the boundary conditions in Eq. 6.2.

$$\mathcal{H} = \left\{ x \in L_2(\Omega, \mathbb{R}^n); q \left( x, \frac{dx}{d\eta}, \ldots, \frac{d^{n-1}x}{d\eta^{n-1}} \right) = 0 \text{ on } \Gamma \right\}$$

(6.5)

We define the inner product and norm in $\mathcal{H}$ as follows:

$$(\phi_1, \phi_2) = \int_\Omega \phi_1^*(z)\phi_2(z)dz, \ ||\phi_1||_2 = (\phi_1, \phi_1)^{1/2}$$

(6.6)

where $\phi_1, \phi_2 \in \mathcal{H}[\Omega, \mathbb{R}^n]$. Defining the state function $x$ on $\mathcal{H}$ as $x(t) = \bar{x}(z, t), t > 0, z \in \Omega$, the operator $\mathcal{A}$ in $\mathcal{H}[\Omega, \mathbb{R}^n]$ as $\mathcal{A}(x) = \mathcal{L}(\bar{x})$, the input, controlled output and measured output operators as $Bu = bu, Cx = (c, x), S_m x = (s_m, x), S_r x = s_r x$, the system of Eqs. 6.1-6.3 acquires the following form in the Hilbert space, $\mathcal{H}$:

$$\dot{x} = A(x) + Bu + f(x), \ x(0) = x_0.$$  

(6.7)

$$y_c = Cx, \ y_m = S_m x, \ y_r = S_r x.$$  

where $f(x) = f(\bar{x}(z, t)), y_m \in \mathbb{R}^p, y_r \in \mathcal{H}[\Omega, \mathbb{R}^n]$ and $x_0 = \bar{x}_0(z)$.

Assumption 3. The long-term dynamics of the above PDE system are finite dimensional in a sense that the state $x$ of the above system can be accurately described by a finite number of degrees of freedom. Thus, in principle, a finite number of appropriately chosen basis functions of $\mathcal{H}$ are sufficient to describe the long term
behavior of $x$. Furthermore, the state $x$ can be partitioned into a finite number of slow and possibly unstable modes and an infinite number of stable and fast modes and there is a time-scale separation between their dynamic behavior.

6.3 Methodology

In this section, we will design output feedback controllers for Eqs. 6.1-6.3 using the APOD methodology. First, the available off-line (partial) process data in the representative ensemble is reconstructed using the gappy reconstruction procedure. These reconstructed off-line snapshots are then utilized to compute empirical basis functions by employing APOD off-line on the data ensemble. We utilize these identified basis functions in a “Galerkin’s” method to construct a reduced order model (ROM) for the above system. As the accuracy of a ROM constructed using the representative ensemble cannot be guaranteed globally, the design of controllers using such “global” ROM would be suspect. Consequently, we design the controller using a low-dimensional locally-valid ROM that has a small region validity (in the entire state space). We then extend the validity of this local ROM (in a computationally efficient way) by updating it periodically using the periodic snapshots, reconstructed online using gappy reconstruction procedure, in APOD. The updated ROM along with continuous point measurements are then employed to update the nonlinear static output feedback controllers and to stabilize the closed-loop system. Figure 6.1 presents the block diagram of our approach. In the
next subsections, we will first focus on the various individual steps involved in the procedure; towards the end of this section we will combine the different steps and present the overall methodology.

### 6.3.1 Gappy reconstruction

In this work, we utilize the gappy reconstruction procedure [75] to reconstruct both the off-line snapshots \( \{u_k\}_{k=1}^N \) and the online snapshots that become available during closed-loop system evolution. We first review this reconstruction procedure for the case where the off-line snapshots are completely available followed by the situation wherein the off-line snapshots are incomplete, in the sense that certain
part of the spatial domain solution is unavailable.

### 6.3.1.1 Complete off-line snapshots:

Let \( \{ \phi \}_{i=1}^{N} \) be the global basis functions computed by standard POD methodology (section 2.1) using the snapshot set \( \{ v_k \}_{k=1}^{N} \). We assume that out of the \( N \) basis functions of \( C_N \), \( w \) have the corresponding eigenvalues that capture \( \epsilon \) percentage energy of the ensemble, \( C_N \). The incompleteness of the online data is usually characterized by mask functions. In a particular snapshot these mask functions identifies the location of both the missing and the available data. In our application, the sensor shape function acts plays the role of a mask function i.e., \( s(z,t_i) = 0 \) if data is missing else \( s(z,t_i) = 1 \). Without loss of generality, let \( v_{N+1} \) be the online snapshot that needs to be reconstructed. Assuming, that this snapshot can be characterized by the existing snapshot set, the intermediate snapshot can be represented in terms of \( w \) basis functions (\( w \leq N \)) as follows:

\[
\tilde{v}_{N+1} \approx \sum_{i=1}^{w} \tilde{b}_i \phi_i \tag{6.8}
\]

In order to compute the POD coefficients \( \tilde{b}_i \), we minimize the following least-square criterion. The error that needs to be minimized is defined as \( ||v_{N+1} - \tilde{v}_{N+1}||^2_g \).

The ‘gappy’ norm used in the above expression is defined based on the following
inner product definition:

\[ (\phi, v_{N+1})_g = (\phi, s(z, t_{N+1})v_{N+1}), \quad ||v_{N+1}||_g = (v_{N+1}, v_{N+1})_g^{1/2} \] (6.9)

where \( s(z, t_{N+1}) \) is the mask function for the \( N + 1^{th} \) snapshot, \( v_{N+1} \). Minimization of the above least-square criterion leads to the following set of linear equations

\[ M\tilde{b} = f \] (6.10)

where the \( (i, j)^{th} \) element of the matrix \( M \) is given by

\[ M_{ij} = (\phi_i, \phi_j)_g \] (6.11)

and

\[ f_i = (\phi_i, v_{N+1})_g \] (6.12)

The intermediate snapshot \( \tilde{v}_{N+1} \) is obtained by solving the linear system Eq. 6.10 for \( \tilde{b} \) and substituting \( \tilde{b} \) in Eq. 6.8. The snapshot, \( v_{N+1} \) is then reconstructed by replacing the missing data by the corresponding intermediate snapshot i.e.,

\[ v_{N+1} = \tilde{v}_{N+1} \text{ if } s_r(z, t_{N+1}) = 0. \]
6.3.1.2 Incomplete off-line snapshots:

When the off-line snapshots are also not completely known, an iterative reconstruction procedure can be employed to obtain the empirical basis functions. Let \( \{ v_k \}_{k=1}^{N} \), be the ensemble of incomplete snapshots along with an associated ensemble of masks \( \{ s(z, t_k) \}_{k=1}^{N} \). The masks can in general be generated randomly as considered in [75, 77]. In this work, we employ a time-varying structure to the masks such that over time the whole spatial domain is “covered” by the sensors.

The iterative procedure consists of the following steps:

1. The first guess for the snapshot set is obtained by replacing the missing data in the off-line snapshots using smooth spline functions. Note that we avoid any bias due to the initial guess by carefully choosing the spline functions such that it’s average (over the regions where data is missing) matches the ensemble time average.

2. Use the standard POD technique to obtain the basis functions \( \{ \phi_{k}^{l} \}_{k=1}^{w_{l}} \) for the current iteration \( \{ l \} \).

3. Reconstruct all the snapshots in the snapshot set, \( \{ v_k \}_{k=1}^{N} \), by employing \( w_{l} \) basis functions using the methodology for complete off-line snapshots described in the above section (\( w_{l} \leq N \); choice of \( w_{l} \) depends on the desired accuracy). The intermediate snapshot set thus generated is represented by the snapshot set \( \{ \tilde{v}_k \}_{k=1}^{N} \).
4. The missing data in the snapshot set are now replaced with the reconstructed data i.e.,

$$v^l_k(z) = \begin{cases} v^l_k(z) & \text{if } s_r(z, t_k) \neq 0 \\ \tilde{v}^l_k(z) & \text{if } s_r(z, t_k) = 0 \end{cases}$$

(6.13)

or

$$v^l_k(z) = \tilde{v}^l_k(z)(1 - s_r(z, t_k)) + v^l_k(z)s_r(z, t_k)$$

where $k = 1 \cdots N$.

5. The iterative procedure terminates once the eigenvalues computed by POD converge, i.e.,

$$\max_{i=1, \cdots, w} |\lambda^{l+1}_i - \lambda^l_i| < \epsilon_g$$

(6.14)

where $\lambda^l_i$ is the $i^{th}$ eigenvalue of the covariance matrix $C_N$ at the $l^{th}$ iteration and $\epsilon_g$ is a user defined convergence threshold. If the above criteria is not satisfied then we set $l = l + 1$ and go to step 2.

**Remark 9.** Note that the convergence of the above reconstruction procedure depends on the quality of the initial guess. Consequently, during the initial iterations, the use of a large number of basis functions that capture a significant amount of energy of the snapshot set, will result in significant errors in the reconstructed ensemble as the initial basis functions are inaccurate. To avoid this, during the initial iterations ($l \leq 5$), we used basis functions which capture 99% of energy; subsequently for $l > 5$ a higher energy of 99.99% was utilized for better accuracy.
The above strategy has significantly improved the convergence and accuracy of the iteration procedure.

**Remark 10.** In step 1 of the above methodology, we use splines as an initial guess to replace the missing data as opposed to using a discontinuous time average [75, 77]. The application of splines considerably improved the convergence of the above iterations as the initial basis functions are not discontinuous.

**Remark 11.** Note that in step 4 of the above procedure, we use the intermediate repaired snapshot to fill in the gaps of the missing data (see Eq. 6.13). As a result, the reconstructed snapshots will contain discontinuities at the boundaries between missing data and available data. These discontinuities arise due to the inherent errors associated with the gappy reconstruction procedure. Thus, when the gappy reconstruction procedure is utilized for design of online feedback controllers these discontinuities need to be monitored and even smoothened (appropriately to retain orthogonality) as they may potentially introduce large numerical errors when we compute spatial derivatives.

### 6.3.2 Adaptive Proper Orthogonal Decomposition:

The gappy reconstruction procedure as presented in section 6.3.1 assumes that the snapshot to be reconstructed can be characterized by the existing snapshot set. This assumption in the realm of process control of distributed systems necessitates the *apriori* availability of a sufficiently large representative ensemble of PDE solu-
tion data in which all the possible spatial modes (especially including those that might appear in closed-loop evolution of the system in Eqs. 6.1-6.3) are excited and their long term behavior is also approximately captured. This is necessary to ensure that the resulting basis functions capture the global dynamics of the system and consequently gappy POD could be used, reliably, for the reconstruction of snapshots that may appear during the closed-loop process evolution.

The ROM computed using such representative ensemble of snapshots would be valid over the entire operational space, as the system passes through different regions of the state space during its closed-loop evolution. However, such ROMs tend to be of larger dimensionality and consequently implementing real-time feedback controllers using such ROMs is computationally demanding. On the other hand ROMs which are valid over a small operational region tend to be computationally efficient, however the validity of such models is limited to the region of the state space around which they are computed. We then utilize the APOD methodology (section 2.2) to address the issues of validity and computational efficiency of ROMs. APOD allows us to use a locally valid ROM since it provides a computationally efficient way for updating the ROM on-demand and assures its validity over the current operational space. We now briefly present the prominent features of APOD. This methodology consists of the following three steps:
6.3.2.1 Off-line: Initial basis Construction:

We initially use a collection of $M$ ($M \leq N$) off-line data snapshots, $\{v_k\}_{k=1}^M$, to construct the initial local basis for Eqs. 6.1-6.3. Note that these $M$ data snapshots cover a small operational space, consequently the basis computed using these snapshots may have a small region of validity. We first construct the covariance matrix $C_M$ then solve the following eigenvalue-eigenvector problem

$$C_M \psi = \lambda \psi \quad (6.15)$$

to compute $M$ eigenvalues. We partition the eigenspace of the covariance matrix, $C_M$, into two subspaces; the dominant one containing the modes which capture at least $\epsilon$ percent of energy in the ensemble (denoted as $P$) and the orthogonal complement to $P$ containing the rest of the modes (denoted as $Q$). By definition such a separation exists for dissipative processes and a finite number of modes belongs in $P$, owing to the elliptic nature of the spatial differential operator [9] (assumption 3). Note that we define $\epsilon$ as the percentage energy of the ensemble captured by dominant basis functions.

We assume that out of $M$ possible eigenvectors of $C_M$, $\tilde{w} (\tilde{w} < w)$ have the corresponding eigenvalues such that $\sum_{i=1}^{\tilde{w}} \lambda_i / \sum_{i=1}^{M} \lambda_i \leq \frac{\epsilon}{100}$: $\tilde{w}$ eigenmodes of $C_M$ capture $\epsilon$ percent of energy in the ensemble. These eigenvectors are then used in
the following equation

\[
\phi_i(z) = \sum_{k=1}^{M} \psi_i^k \psi_k(z), \quad i = 1, \ldots, \tilde{w}.
\]

to compute \(\tilde{w}\) basis functions; here \(\phi_i\) represents the \(i^{th}\) eigenfunction and \(\psi_i^k\) is the \(k^{th}\) element of the \(i^{th}\) eigenvector of \(C_M\). An orthonormal basis for the subspace \(\mathbb{P}\) can be obtained as:

\[
Z = [\psi_1, \psi_2, \ldots, \psi_{\tilde{w}}], \quad Z \in \mathbb{R}^{M \times \tilde{w}} \quad (6.16)
\]

where \(\psi_1, \psi_2, \ldots, \psi_{\tilde{w}}\) denote the eigenvectors of \(C_M\) that correspond to the eigenvalues \(\lambda_1, \lambda_2, \ldots, \lambda_{\tilde{w}}\). Note that the basis functions computed by these eigenvectors capture the dominant dynamics of the PDE system of Eqs. 6.1-6.3. The orthogonal projection operators \(P\) and \(Q\) onto subspaces \(\mathbb{P}\) and \(\mathbb{Q}\) can be computed as

\[
P = ZZ^T, \quad Q = I - ZZ^T \quad (6.17)
\]

where \(I\) denotes the identity matrix of dimension \(M\).

6.3.2.2 Derivation of Reduced order model:

In this section, we derive a locally valid ROM of the system of Eq. 6.7 using the above local basis functions. Using assumption 3, the Hilbert space \(\mathcal{H}\) is partitioned into two subspaces \(\mathcal{H}_s\) & \(\mathcal{H}_f\) respectively. \(\mathcal{H}_s\) includes the slow evolving modes whereas \(\mathcal{H}_f\) includes fast evolving stable process modes. From the assumption \(\mathcal{H}_f\)
is an infinite-dimensional subspace, while $\mathcal{H}_s$ is a finite-dimensional one. Clearly, $\mathcal{H} = \mathcal{H}_s \oplus \mathcal{H}_f$. Defining orthogonal projection operator $Q$, the state $x \in \mathcal{H}(\Omega)$ can be decomposed as $x_s = Px \in \mathcal{H}_s$ and $x_f = Qx \in \mathcal{H}_f$. The state $x$ of the system of Eq. 6.7 now can be expressed as:

$$x = x_s + x_f = Px + Qx$$ (6.18)

Applying projection operators $P$ and $Q$ to the system of Eq. 6.7 and using the above decomposition of $x$, the system of Eq. 6.7 can be equivalently expressed as:

$$\frac{dx_s}{dt} = A_s(x_s, x_f) + B_s u + f_s(x_s, x_f)$$
$$\frac{\partial x_f}{\partial t} = A_f(x_s, x_f) + B_f u + f_f(x_s, x_f)$$
$$y_c = C x_s + C x_f, y_m = S_m x_s + S_m x_f$$

(6.19)

where $A_s = PA(x_s + x_f), B_s = PB, f_s = Pf, A_f = QA(x_s + x_f), B_f = QB$ and $f_f = Qf$ and the notation $\partial x_f / \partial t$ is used to denote that the state $x_f$ belongs in an infinite dimensional subspace ($\mathcal{H}_f$).

Using singular perturbation arguments for infinite dimensional systems [25], we neglect the infinite dimensional fast and stable $x_f$ subsystem in Eq. 6.19. The
following $\tilde{w}$-dimensional $x_s$ subsystem is obtained:

$$\frac{dx_s}{dt} = A_s(x_s, 0) + B_s u + f_s(x_s, 0)$$

$$y_{cs} = C x_s, \ y_{ms} = S_m x_s$$

$$x_f \equiv 0 \quad (6.20)$$

where the subscript $s$ in $y_{cs}$ and $y_{ms}$ indicates that these outputs are associated with the $x_s$ subsystem. Under the assumptions already stated, the finite dimensional system is an accurate approximation of the dominant dynamics of the infinite dimensional system of Eq. 6.7.

Note that we use the basis functions, $\phi$, computed in the previous section to define the subspaces $H_s$ & $H_f$, i.e., $H_s = \text{span}\{\phi_1, \phi_2, \ldots, \phi_{\tilde{w}}\}$ and $H_f = H \setminus H_s$. In order to ensure the validity of the decomposition the following assumption is made:

**Assumption 4.** Subspace $P \in H$ defined as $P = \text{span}\{\phi_1, \phi_2, \ldots, \phi_{\tilde{w}}\}$, where $\phi$ are basis functions identified via the APOD methodology using the covariance matrix $C_M$, appropriately captures the $H_s$ subspace, i.e., $H_s \subseteq P$.

As the basis functions, $\phi$, have a small range of validity, the ROM (Eq. 6.20) computed using them may not remain valid during the course of closed-loop process evolution. To avoid this situation, we periodically update the basis functions, $\phi$, when necessary using APOD thus extending the Roms validity over the current operational space.
6.3.2.3 **Online recursive update of ROM:**

To update the locally valid ROM in Eq. 6.20, we assume that new snapshots with partial data becomes available “periodically” during the course of closed loop process evolution. We reconstruct these snapshots using the gappy reconstruction procedure in section 6.3.1. These reconstructed snapshots are used to recursively update the orthonormal basis for the subspace $\mathbb{P}$, possibly by increasing or decreasing the size of the basis if required and by maintaining the accuracy of basis by performing orthogonal power iteration. The orthonormal basis for $\mathbb{Q}$ can also be directly computed since $\mathbb{Q}$ is the orthonormal complement of $\mathbb{P}$. We maintain that the extra work required for the above process is small as long as the dimension of $\mathbb{P}$ is small. We then update the ROM in Eq. 6.20 using these updated basis functions. This step assures that during the closed-loop operation the ROM captures new trends that appear when the process traverses through previously unsampled regions in state space.

The algorithm outlined below computes an approximation to $Z$ without requiring the solution of the eigenvalue-eigenvector problem of the covariance matrix (Eq. 6.15). To simplify the algorithm we also assume that the dimensionality of the covariance matrix $C_M$ remains constant. This is achieved by discarding the oldest snapshot from the ensemble as a new one is obtained. As a new snapshot from the process becomes available, the subspace $\mathbb{P}$ may change in the following three ways:
• The dimension of the dominant subspace $P$ may increase i.e., one mode corresponding from $Q$ becomes necessary to capture the desired percentage of energy in the ensemble. This is ascertained by monitoring the contribution of the dominant eigenvalue of $c_q = QC_MQ$, namely $\lambda_{\tilde{w}+1}$ towards the total energy of the ensemble, i.e.,

$$\xi = \frac{\lambda_{\tilde{w}+1}}{\sum_{i=1}^{\tilde{w}+1} \lambda_i}$$

If $\xi$ increases to more than $(100 - \epsilon)$ percent we append $Z$, the basis of subspace $P$, with the corresponding eigenvector associated with $\lambda_{\tilde{w}+1}$.

• Some of the eigenmodes of the subspace $P$ may no longer be necessary to capture the required $\epsilon$ percent of the energy. In this case, the basis $Z$ should be updated and its dimension should be simultaneously decreased. To test this the following $\tilde{w} \times \tilde{w}$ matrix $H = Z^T C_M Z$ is introduced. If only $\hat{w}$, with $\hat{w} \leq \tilde{w}$, eigenvalues of $H$ are dominant then the basis $Z$ is updated and its dimension is simultaneously decreased.

• The dimensionality of $P$ remains unchanged. However the basis $Z$ is updated, whenever the current basis is not accurate, to maintain the accuracy of the basis. If after analyzing the accuracy of the current basis it is necessary to update the basis, the following one step power iteration $Z = \text{orth}(C_MZ)$ is executed.
A flow chart illustrating the above steps is presented in Figure 2.1. Based on the new values of \( Z \), we now compute the revised basis functions \( \phi_1, \phi_2, \ldots, \phi_{\tilde{w}} \) as a linear combination of the snapshots given by the following equation

\[
\phi_i(z) = \sum_{k=1}^{M} \psi_k^i v_k(z), \quad i = 1, \ldots, \tilde{w}.
\] (6.21)

where \( \psi_k^i \) denotes the \( k^{th} \) element of vector \( \psi_i \). These updated basis functions are then utilized to update the ROM of Eq.6.20. In between the periodic updates of ROM, we assume that the updated ROM computed in Eq. 6.20 remains a valid representation of the original system in Eq. 6.7. The periodic updates guarantee that the assumption 4 remains valid for the duration of the closed-loop operation.

### 6.3.3 Complete Closed-loop methodology

The complete closed-loop methodology, illustrating its different steps, is presented in figure 6.1. We initially reconstruct the available \( (M) \) off-line data snapshots using the gappy reconstruction procedure (section 6.3.1) off-line. These reconstructed snapshots are then employed in the off-line step of APOD to construct an initial ROM (subsections 6.3.2.1 & 6.3.2.2). We then utilize the on-line step of APOD methodology (whose individual steps are presented in subsection 6.3.2.3 and illustrated in figure 2.1) to periodically update the ROM using the closed-loop periodic (partial) snapshots, reconstructed online using gappy reconstruction procedure (section 6.3.1). The updated ROM along with continuous point measure-
ments is then employed to update the nonlinear static output feedback controllers and stabilize the closed loop system. The specific form and the stability aspects of the designed controller are discussed in the following section.

6.4 Nonlinear static Output feedback control

In this section, we utilize the $\tilde{w}$ dimensional ROM of Eq. 6.20 to design nonlinear static output feedback controllers for the PDE system of the form given in Eqs. 6.1-6.3. In our previous results [88, 89], we designed state feedback & output feedback controllers for PDE systems based on continuous & periodic availability of full state measurements of the process. The availability of such full state measurements, even periodically, is usually restricted in practice due to high costs and reliability issues associated with these sensors. In this work, we avoid this limitation by assuming the availability of partial snapshots, $y_r$, periodically. These periodic measurements are used for the state reconstruction and recursive update of ROM. We consider the synthesis of static output feedback controllers of the following form:

$$u = \alpha(\tilde{x}_s)$$  \hspace{1cm} (6.22)

where $\alpha(\cdot)$ is a smooth vector function and $\tilde{x}_s$ are the estimates of the states of the ROM Eq. 6.20. The above controller designed using the continuous point
measurements, $y_m$, consists of two parts (1) a Lyapunov based state feedback controller [92] and (2) a state observer. Using the static observer we estimate the states, $\tilde{x}_s$, using the limited number of continuous point measurements. The following assumption is needed in the design of the static observer.

**Assumption 5.** The number of available point measurements, $p$, is equal to or greater than the dimension of the subspace $H_s$, $\tilde{w}$, i.e. $p \geq \tilde{w}$, and the inverse of the operator $S_m$, $S_m^\perp$, exists, so that $\tilde{x}_s = S_m^\perp y_m$.

When the finite-dimensional space, $\mathbb{P}$, is expressed using the empirical basis functions as basis functions the operator $S_m^\perp$ becomes a matrix and is given by

$$S_m^\perp = (S_m^T S_m)^{-1} S_m^T$$  \hspace{1cm} (6.23)

Note that the existence of inverse of matrix $S_m$ depends on the location and shape of the measurement sensors; this amounts to properly choosing $s_m(z)$ in Eq. 6.1. During the operation of the above controller we only employ data from the point measurement sensors. Consequently, the sensor shape distribution function, $s_m(z)$, is independent of time and is given by $s_{mi}(z) = \delta(z - z_i)$, where $z_i$ is the location of $i^{th}$ sensor. The specific form and the stability aspects of the designed controller are discussed in the following lemma.

**Lemma 2.** Consider the parabolic PDE system in Eq. 6.7, for which assumptions
1, 2 & 3 hold. Also consider the nonlinear output feedback controller that is designed based on Eq. 6.20:

$$u = -k(\tilde{x}, c_0)L\beta V_{\tilde{w}}(\tilde{x})$$  \hspace{0.5cm} (6.24)

where

$$k(\tilde{x}, c_0) = \begin{pmatrix} c_o + \frac{L_{F_s}V_{\tilde{w}}(\tilde{x}) + \sqrt{(L_{F_s}V_{\tilde{w}}(\tilde{x}))^2 + (L_{B_s}V_{\tilde{w}}(\tilde{x}))^2}}{(L_{B_s}V_{\tilde{w}}(\tilde{x}))^2}, L_{B_s}V_{\tilde{w}}(\tilde{x}) \neq 0 \\ c_o, L_{B_s}V_{\tilde{w}}(\tilde{x}) = 0 \end{pmatrix}$$

$$\tilde{x} = S_m y_m, F_s = A\tilde{x} + f_s(\tilde{x}, 0), L_{F_s}V_{\tilde{w}} = \frac{\partial V_{\tilde{w}}}{\partial \tilde{x}}, L_{B_s}V_{\tilde{w}} = \frac{\partial V_{\tilde{w}}}{\partial \tilde{x}}, V_{\tilde{w}}(\tilde{x}(t_i)) = \frac{\zeta}{2} \tilde{x}(t_i)^T \tilde{x}(t_i),$$  where \(\zeta\) is a periodically updated used-defined parameter. Then the controller in Eq. 6.24 asymptotically stabilizes the system in Eq. 6.7.

See Appendix A for the proof of the lemma.

### 6.5 Kuramoto-Sivashinsky equation

In this section, we illustrate through numerical simulations the ability of the proposed output feedback controller in stabilizing the Kuramoto-Sivashinsky equation (KSE) using periodic partial measurements. The integrated form of KSE and the associated periodic boundary conditions are described in section 4.5.1. The value of the diffusion parameter \(\nu\) is set as \(\nu = 0.25\). Six control actuators were assumed to be available at the following locations \(L = [\pi/4, \pi/5, \pi/2, -\pi/2, -\pi/6, -\pi/4]\); the corresponding spatial distribution functions at these locations are \(b_i(z) =\)
Figure 6.2. Open-loop profile of the state of Eq. 4.15 with $\nu = 0.25$.

$\delta(z - L_i); i = 1, \ldots, 6$. Continuous point measurements were assumed to be available from 10 point measurement sensors placed uniformly across the domain of the process $(-\pi, \pi)$. The sensor shape distribution function, $s_m(z)$, for all time $t$, at these respective positions is $s_{mi}(z) = \delta(z - P_i); i = 1, \ldots, 10$, where $P_i$ is the location of $i^{th}$ sensor. In all the simulation runs, the following spatially non-uniform initial condition is considered: $x_0 = 3\sin(z) - \cos(2z) - \sin(5z) + 2\cos(5z)$. Figure 6.2, presents the traveling wave pattern observed in the open-loop evolution of KSE wherein we observe the formation of persistent waves. Thus the control objective is set to stabilize the process at the spatially uniform steady state $x(z,t) = 0$. 
6.5.1 Case 1: Complete off-line snapshots

We assume that the snapshots that were collected off-line (for construction of POD basis functions) are complete without any missing data. The online process measurements were however assumed to be available only from 10 point measurement sensors. The corresponding sensor shape distribution functions at these 10 locations are given by $s_m = s_r = \delta(z - P_i); \ i = 1, \ldots, 10$. We then utilize these point measurements in gappy reconstruction procedure for the online reconstruction of process states. As discussed in section 6.3.1, gappy reconstruction procedure assumes that the new snapshot to be reconstructed can be characterized by using the existing basis functions computed through off-line snapshots. To this end, we construct an off-line representative ensemble of snapshots by exciting the system Eq. 4.15 using different variations in the input $u(t)$ profile [24, 59]. Application of POD to this ensemble resulted in $w = 8$ global basis functions that captured 99% of the energy of the ensemble. These basis functions were further employed to compute the global ROM (section 6.3.2.2). The computed global ROM captures the entire closed-loop process dynamics, thus the online reconstruction using gappy procedure remains accurate.

However, the computed ROM tends to be of higher dimensionality consequently the design of controller using such global ROM is computationally expensive. To demonstrate an alternative procedure, we employ the computed global ROM only for online state reconstruction. The controller design, on the other hand is executed
using a low dimensional ROM (local ROM) that is valid over a local operational region. We then periodically update the local ROM using APOD, thus extending its validity over the current operational space.

For the design of controller, an initial ensemble of 100 open-loop snapshots ($M = 100$) was collected by simulating the process with $u(t) \equiv 0$ till $t = 2$. Application of POD to this ensemble resulted in $\tilde{w} = 3$ basis functions that captured 99% of the energy of the ensemble. These basis functions were then employed in the computation of the local ROM (Eq. 6.20) for the above process. An output feedback controller of Eq. 6.24 was subsequently designed, based upon this ROM, to achieve the desired control objective. We assume that the partial closed-loop state information becomes available at a time step of every $\delta t = 0.25$. The recursive check of the local ROM validity and consequently the task of online reconstruction of the snapshots was undertook at a period of every $\delta t = 0.25$. The parameter $\zeta$ in lemma 2 is set to be $\zeta = 1$ throughout the simulation time. The percentage error for snapshot reconstruction is computed as:

$$e_{\text{percent}} = \frac{||v_{N+1} - \tilde{v}_{N+1}||_2^2}{||v_{N+1}||_2^2} \times 100$$  \hspace{1cm} (6.25)$$

while the absolute error is computed as follows:

$$e_{\text{absolute}} = ||v_{N+1} - \tilde{v}_{N+1}||_2$$  \hspace{1cm} (6.26)$$
where \( v_{N+1} \) is the online snapshot that needs to be reconstructed and \( \tilde{v}_{N+1} \) is the gappy reconstructed snapshot (see Eq.6.8).

The process was operated in closed-loop using the designed controller. Figure 6.3(a) presents the closed-loop evolution profile of KSE. The controller successfully stabilized the process at the desired operating point of \( x(z, t) = 0 \). The control action used for achieving this result is presented in figure 6.3(b). We observe that the control action is a smooth function of time which converges to zero at the end of the successful closed-loop process operation. At time \( t = 9.8 \), we observe a small discontinuity in the control action due to the change of dimensionality of the ROM as presented in figure 6.5(b). Figures 6.4a & 6.4b present the temporal profile of the \( L_2 \) norm of the system and the Lyapunov function \( V_w(t) \). We note that both the \( L_2 \) norm and Lyapunov function converge uniformly to zero even though there is a dimensionality increase at time 9.8. Note that at that point we also observe an increase in the rate of state convergence to the desired flat profile. Figure 6.5a presents both the percentage error and the absolute error in on-line state reconstruction from 10 point sensors using gappy reconstruction procedure. We note that both the percentage error and the absolute error stay within 3\% and reach zero as the control objective is achieved. Note that the spike in the percentage error observed at \( t = 9.8 \) corresponds to a small change in the absolute error and is due to a system response to an increase in the control action (figure 6.3(b)) caused because the dimensionality of the ROM changes as determined by
Figure 6.3. a) Closed-loop profile of the state of Eq. 4.15. b) Temporal profile of the control action, u(t). (ε = 0.99)
Figure 6.4. a) $L_2$ norm of the closed-loop profile of Eq. 4.15. b) Temporal profile of the Lyapunov function, $V(t)$. ($\epsilon = 0.99$)
Figure 6.5. a) Temporal profile of the state reconstruction error using gappy reconstruction. b) Temporal profile of dimensionality of the local ROM of Eq. 4.15. ($\epsilon = 0.99$)
The change in the dimensionality of the local ROM is presented in figure 6.5(b). Note that the dimensionality of a global ROM computed using POD remains at 8; on the other hand, the dimensionality of a local ROM starts at 3 and increases to 5 for a brief time and then becomes 4 again. Using a local ROM considerably reduced the computational effort, while designing ROM based controllers that stabilized the process. This observed improvement will be more prominent during the implementation of APOD on a large scale process involving multiple system states.

To present the importance of APOD in updating the local ROM, we now switch
Figure 6.7. a) $L_2$ norm of the closed-loop profile of Eq. 4.15 without APOD. b) Closed-loop profile of state of Eq. 4.15 without APOD. ($\epsilon = 0.99$)
off the APOD and implement the controller that was designed based on the local
ROM with a corresponding dimensionality $\tilde{w} = 3$. As the validity of the local ROM
was restricted to a small region of the operational space and the closed-loop process
traversed through different regions of the state space, the local ROM failed to
describe the process. Accordingly, the Lyapunov function, $V_{\tilde{w}}(t)$ of the closed-loop
evolution profile of KSE (in figure 6.6) did not converge to zero and the controller
clearly failed to achieve the necessary control objective. The corresponding profile
of $L_2$ norm and the closed-loop profile is presented in figures 6.7(a) and 6.7(b)
respectively.

6.5.2 Case 2: Partial off-line snapshots

In case 2, we assume that the snapshots that were collected off-line (for con-
struction of POD basis functions) were incomplete i.e., there are regions where
information was unavailable. A typical pattern used for the incomplete data is
presented in figure 6.8. Note that there was no particular spatial subregion of
the domain that remains unobservable for all the times. In the presented simu-
lations, at any time instant 20% of the data in each snapshot was missing. We
first utilized the iterative gappy reconstruction procedure (section 3.2) to “fill” in
the incomplete data. The different stages involved in employing this reconstruction
procedure are presented in figure 6.9(a) by using open-loop contour plots. Initially,
we “replaced” the gaps in all the snapshots using a spline interpolation. We then
utilized this new ensemble as a first guess to construct an approximate POD basis. As discussed in remark 2, we note that a POD basis was constructed to capture 99% energy of the off-line ensemble during the initial iterations ($l < 5$) and for the rest of iterations an higher energy of 99.99% was used. Figure 6.9(b) presents the approximation of the ensemble at the end of the first iteration and figure 6.9(c) presents the approximation of the ensemble at the end of the 30th iteration. We note that as we increase the number of iterations the ensemble matches close to the original ensemble with no missing data. We note that a similar reconstruction procedure was performed on the (global) ensemble.

After the reconstruction of the off-line ensemble, we followed the proposed
Figure 6.9. a) Contour plot of the evolution of KSE from $t = 0$ to $t = 2.5$ with 20% data missing per snapshot. b) Contour plot after 1st iteration using gappy reconstruction procedure c) Contour plot after 30th iteration.
procedure elucidated in subsection 6.5.1. We initially derived the global ROM for
the online reconstruction of snapshots and subsequently a local ROM was derived
for the efficient design of online feedback controllers. As in the previous case, we
collected an initial ensemble of 100 open-loop snapshots by simulating the process
with $u(t) \equiv 0$ till $t = 2.5$ and utilized POD to compute a ROM using a 99% energy
criterion. A local ROM was then derived using these basis functions and
was utilized to design an output feedback controller of Eq. 6.24, setting $\zeta = 1$
throughout the process. The periodic update of local ROM and consequently the
online reconstruction of the snapshot was pursued at a period of every $\delta t = 0.25$.

Figure 6.10(a) presents the closed-loop evolution profile of KSE. We note that
the controller successfully stabilized the process at the desired operating point
of $x(z, t) = 0$. The control action used for achieving this result is presented in
figure 6.10(b). Similar to the previous case, we observe a small discontinuity in
control action at $t \approx 8$ due to a change in the dimensionality of the dominant
subspace $P$ (which is now accounted for by the controller). In spite of the small
discontinuity observed in the control action the system converged to zero at the
end of closed-loop process operation. Figures 6.11a & 6.11(b) present the temporal
profile of the $L_2$ norm of the system and the Lyapunov function $V_{\tilde{w}}(t)$, respectively.
Both the $L_2$ norm and Lyapunov function converge uniformly to zero, even though
there is a increase in the dimensionality of the ROM at $t \approx 8$, thus indicating
the successful closed-loop operation. Figure 6.12a presents both the percentage
error and the absolute error in on-line state reconstruction from snapshots with 20% incomplete data. The oscillation observed in the percentage error is due to the discontinuities that arise at the boundary points between the missing data and available data. We note that both the percentage error and the absolute error stay within 1.5%. The change in the dimensionality of the local ROM is presented in figure 6.12(b). The global ROM that would capture all the closed-loop process characteristics would require the dimensionality of 8 to capture 99% of the energy of the global ensemble. On the other hand the dimensionality of the local ROM started at $\tilde{w} = 3$ and increased to 5 at $t \approx 8$ and then reduced to a dimensionality of 4 at $t \approx 27$. Thus the use of local ROM significantly reduced the computational effort in the implementation of the online controller. We expect that the amount of computational savings to be more pronounced on the application of this methodology to a large-scale process involving a large dimensionality.

In order to present the robustness of the approach, we computed the local ROM using the value of 90% for the energy criterion ($\epsilon = 0.9$). Figures 6.13(a) & 6.13(b) present the temporal profile of $L_2$ norm and the Lyapunov function of the closed-loop system under the action of the designed controller in Eq. 6.24 respectively. We note that after the observed initial convergence the $L_2$ norm starts to increase at $t \approx 2$ as the local ROM (due to low dimensionality) failed to capture the new closed-loop trends. This is due to the small chosen value of the energy criterion $\epsilon$. However, APOD detected these non-captured new trends and automatically
increased the dimensionality of the local ROM at $t \approx 3$. Figure 6.13(c) presents this change in the dimensionality of the local ROM. Note that since the evolution of $V$ in figure 6.13(b) satisfied the hybrid stability requirements of Eq.B.8 in lemma 2; it is thus implied that the process was closed-loop stable.

Subsequently, the controller designed using this local ROM (with increased dimensionality) stabilized the process at $x(z,t) = 0$. Note that $\zeta$ was set to be $\zeta = 1$ in the simulation and it was not necessary to update its value during the process evolution. Thus, even though the error increased due to poorly chosen tunable parameter values, the proposed methodology was robust in stabilizing the process at the desired steady-state.

6.6 Conclusions

This work presented the efficient design of output feedback controllers (using partial sensor information) for processes mathematically modeled using parabolic PDEs. Initially, the partial states were reconstructed using gappy iteration procedure. These reconstructed measurements were then utilized for derivation and update of ROMs using APOD methodology. The efficient recursive ROM updates by APOD allowed us to use low-dimensional model while designing controllers, thus resulting in computational savings. The proposed methodology was successfully used to achieve the closed-loop stabilization of process mathematically expressed by Kuramoto-Sivashinsky equation.
Figure 6.10. a) Closed-loop profile of the state of Eq. 4.15. b) Temporal profile of the control actuation, u(t). ($\epsilon = 0.99$)
Figure 6.11. a) $L_2$ norm of the closed-loop profile of Eq. 4.15. b) Temporal profile of the Lyapunov function, $V(t)$. ($\epsilon = 0.99$)
Figure 6.12. a) Temporal profile of the state reconstruction error using gappy reconstruction. b) Temporal profile of dimensionality of the local ROM of Eq. 4.15. $(\epsilon = 0.99)$
Figure 6.13. Closed-loop process evolution for small value of \( \epsilon = 0.9 \) a) \( L_2 \) norm of the closed-loop profile of Eq. 4.15. b) Temporal of the Lyapunov function, \( V(t) \) c) Temporal profile of dimensionality of the local ROM of Eq. 4.15.
CHAPTER SEVEN

ROBUST CONTROL USING EXTENDED KALMAN FILTER

7.1 Introduction

This chapter focuses on dynamic output feedback based robust control of quasi linear parabolic partial differential equations (PDE) systems with time-varying uncertain variables. Especially processes that are described by dissipative PDE’s are considered. The states of the process required for designing controllers are dynamically estimated from limited number of noisy process measurements employing an extended Kalman filter. The issue of utilizing these estimated states in a robust controller to achieve the desired process objective is investigated. The controller
design needs to address both model uncertainty and sensor noise. The methodology is employed on an representative example wherein the desired objective is to stabilize an unstable operating point in a catalytic rod, where an exothermic reaction occurs. A finite dimensional robust controller, utilizing dynamically estimated states, is used to successfully stabilize the process to an open-loop unstable steady-state.

### 7.2 Mathematical Preliminaries

We focus on the problem of feedback control of spatially distributed processes described by highly dissipative PDEs with the following state-space description:

\[
\frac{\partial x}{\partial t} = Ax + b(z)u + f(x) + W(x, r(z)\theta(t)),
\]

\[
y_c = \int_\Omega c(z) x \, dz, \tag{7.1}
\]

\[
y_m = \int_\Omega s(z) x \, dz + v(t).
\]

subject to the mixed-type boundary conditions:

\[
C_1x(\alpha, t) + D_1 \frac{\partial x}{\partial z}(\alpha, t) - R_1 = 0
\]

\[
C_2x(\beta, t) + D_1 \frac{\partial x}{\partial z}(\beta, t) - R_2 = 0 \tag{7.2}
\]
and the following initial condition

\[ x(z, 0) = x_0(z). \]  \hspace{1cm} (7.3)

In the above PDE system, \( x(z, t) \in \mathbb{R}^n \) denotes the vector of state variables, \( y_c \in \mathbb{R}^k \) denotes the vector of controlled outputs, \( t \) is the time, \( y_m \in \mathbb{R}^{n_m} \) denotes the vector of measured outputs, \( z \in \Omega \subset \mathbb{R} \) is the spatial coordinate and \( \Omega = [\alpha, \beta] \) is the domain of definition of the process. \( \mathcal{A} \) is a highly dissipative, linear spatial differential operator, \( f(x) \) is a nonlinear vector function which is assumed to be sufficiently smooth with respect to its arguments, \( c(z) \) is a known smooth vector function of \( z \) which is determined by the desired performance specifications in the domain \( \Omega \) and \( s(z) \) is a known smooth vector function of \( z \) which is determined by the location and type of measurement sensors (e.g., point or distributed sensing). \( W(x, r(z)\theta(t)) \) is a nonlinear vector function, \( \theta(t) \) denotes uncertain process parameters or exogenous disturbances, \( r(z) \) is a known smooth function of \( z \) that specifies the position of action of the uncertain variables on \( \Omega \), \( v \sim \mathcal{N}(0, R) \) is a gaussian white noise sequence of intensity \( Q \). \( u = [u_1, u_2, \cdots, u_k] \in \mathbb{R}^k \) denotes the vector of manipulated inputs, \( b(z) \in \mathbb{R}^{n \times k} \) is a known smooth matrix function of \( z \) of the form \( [b_1(z), b_2(z), \cdots, b_k(z)] \), where \( b_i(z) \) describes how the \( i^{th} \) control action \( u_i(t) \) is distributed in the spatial domain \( \Omega \). \( C_1, D_1, C_2, D_2, R_1,R_2 \) are nonlinear vector functions which are assumed to be sufficiently smooth and \( x_0(z) \) is a smooth vector function of \( z \). We assume that for a given set of initial and
boundary conditions the system of Eqs. 7.1-7.3 has a unique solution. We formulate the problem in the space of square integrable functions $L_2[\Omega]$ and employ the following definition for the norm:

$$
(\phi_1, \phi_2) = \int_{\Omega} \phi_1^*(z)\phi_2(z)dz, \quad ||\phi_1||_2 = (\phi_1, \phi_1)^{1/2} \quad (7.4)
$$

where $\phi_1, \phi_2 \in L_2[\Omega]$ and $^*$ denotes the complex conjugate transpose.

### 7.3 Problem formulation and design of robust controller

In this section, our objective is to present an outline of the steps of the proposed dynamic output feedback control methodology for processes that are described by the system of Eqs.7.1-7.3. The control problem is formulated as the one of deriving a feedback control law $u(t) = G(\hat{x}(t))$, where $G(\hat{x}(t))$ is an estimate of $x(t)$, such that the closed-loop system is stabilized within a neighborhood of the desired set point. Without loss of generality, we assume the setpoint is $x(z,t) = 0$. The steps of the proposed methodology to achieve the above task are:

1. The eigenfunctions of the spatial differential operator $\mathcal{A}$ are used to derive reduced order model of the infinite-dimensional PDE system of Eqs.7.1-7.3 by using Galerkin’s method.
2. Employ Extended Kalman Filter to provide estimates of the states of the resulting finite-dimensional approximation, using information obtained from limited noisy measurement sensors.

3. Design a robust controller using the state estimates to drive the PDE system Eqs.7.1-7.3 within a neighborhood of the desired setpoint.

![Figure 7.1](image-url)  
**Figure 7.1.** Process operation block diagram under proposed controller design.

A block diagram elucidating the above steps is presented in Fig. 7.1. The following subsections briefly describe each of the above steps.
7.3.1 Derivation of reduced order model using Galerkin’s method

We employ the spectral eigenfunctions of the operator $A$ to derive a reduced order model (ROM) of the original infinite-dimensional PDE system of Eq. 7.1 by using Galerkin’s method. To simplify the notation, without loss of generality we consider the system of Eq. 7.1 with $n = 1$. In principle, $x(z, t)$ can be represented as an infinite weighted sum of a complete set of eigenfunctions $\phi_k(z)$. We can obtain an approximation $x_N(z, t)$, by truncating the series expansion of $x(z, t)$ up to order $N$, as follows:

$$x_N(z, t) = \sum_{k=1}^{N} a_k(t) \phi_k(z) \xrightarrow{N \to \infty} x(z, t) = \sum_{k=1}^{\infty} a_k(t) \phi_k(z) \quad (7.5)$$

where $a_k(t)$ is a time-varying coefficient called the mode of the system. The eigenfunctions are obtained from the solution of the eigenfunction eigenvalue problem of the operator $A$

$$A \phi = \lambda \phi. \quad (7.6)$$

We assume that the eigenfunction problem can be solved analytically. Furthermore, we order the eigenfunction-eigenvalue pairs such that

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq \cdots$$
A known property of highly dissipative PDEs is that the eigenspectrum of the operator $\mathcal{A}$ can be partitioned into a finite size of ones that are close to the imaginary axis and an infinite size set of eigenvalues that lies in the left half plane. Furthermore there is a separation between the “slow” and “fast” eigenvalues

$$\frac{\lambda_N}{\lambda_{N+1}} = O(\epsilon)$$

where $\epsilon$ is small number. This implies that the long term dynamics of the process can be accurately described by a finite dimensional approximation and there is a time-scale separation between the slow dynamics and the fast highly-stable ones.

Thus substituting the expansion of Eq.7.5 into Eq.7.1, multiplying the PDE with the eigenfunctions, $\phi(z)$, and integrating over the entire spatial domain (i.e., taking inner product in $L_2[\Omega]$ with the eigenfunctions), the following $N$-th order system of ODEs is obtained.

$$-\sum_{k=1}^{N} a_k (\int_{\Omega} \phi^*(z)\phi_k(z)dz) + \int_{\Omega} \phi^*(z)A\sum_{k=1}^{N} a_k(t)\phi_k(z)dz$$

$$+ \int_{\Omega} \phi^*(z)b(z)udz + \int_{\Omega} \phi^*(z)f(\sum_{k=1}^{N} a_k(t)\phi_k(z))dz$$

$$+ \int_{\Omega} \phi^*(z)W(\sum_{k=1}^{N} a_k \phi_k(z), r(z)\theta(t))dz = 0$$
The resulting ODE system along with the measurement equation is written in the following compact form

\[ \dot{a} = \mathcal{F}(a) + Gu + W_s(a, \theta) \]

\[ y_m = \Phi a + v \]  

(7.7)

where \( \mathcal{F}, W_s \) are vector functions of appropriate dimensions and \( \mathcal{G}, \Phi \) are matrix functions of appropriate dimensions. We note that the only information assumed to be available about the model uncertainty term, \( W_s \), is a time varying bounding function \( c_0(t) \) that captures the size of uncertain terms.

The availability of measurement sensors are often restricted and the measurement available from them tend to be noisy; estimation techniques to predict the system states in Eq.7.7 are required. Since the state equation (Eq.7.7) is nonlinear we use a nonlinear filtering tool called Extended Kalman filter to estimate the system states.

### 7.4 Extended Kalman filter

Given a linear model of the system along with noisy output measurements, the Kalman filter (KF) \([108]\) provides an optimal estimate of the system states. KF operates by propagating the mean & covariance of the system states through time. For nonlinear systems, various formulations of nonlinear Kalman filters are available \([109]\). Extended Kalman filter (EKF) is one such formulation which has been
heavily used as a nonlinear filtering tool in the literature. In EKF, the state equations of the system (Eq.7.7) are linearized at each time step and the states are estimated by using these linearized state equations in the KF. The relevant state equation and measurement equation from the above section is

\[
\dot{a} = \mathcal{F}(a) + Gu + w
\]

\[
y_m = \Phi a + v
\]

\[
w \sim \mathcal{N}(0, Q); \quad v \sim \mathcal{N}(0, R)
\]

Where \(w, v\) are Gaussian zero-mean white noise sequences with intensities \(Q\) and \(R\); represent process noise and measurement noise, respectively. We note that the process noise is introduced in the above equation to act as a cover for model uncertainty, as EKF doesn’t accommodate model uncertainty directly; rather the issue is implicitly addressed through the definitions of process noise and measurement noise. Because, in this work we employed a continuous time version of EKF we present a brief outline of the EKF; a detailed overview can be found elsewhere [110].

1. Initialize the state estimates \(\hat{a}\) and the error covariance \(P\).

2. Linearize the state equation at the current state estimate to obtain the fol-
lowing partial derivative matrix.

\[ A = \frac{\partial F(a)}{\partial a} \mid _{\dot{a}} \]

And solve the following Riccati equation for the error covariance matrix,

\[ \dot{P} = AP + PA^T + Q - P\Phi^TR^{-1}\Phi P \]

and compute the Kalman gain matrix.

\[ K = P\Phi^TR^{-1} \]

Use the computed Kalman gain matrix to get an revised estimate of the states \( a \), by solving the following equation.

\[ \dot{\hat{a}} = F(\hat{a}) + Gu + K[y - \Phi\hat{a}] \quad (7.10) \]

**Remark 12.** We note that EKF is not known to produce accurate estimates for systems with model uncertainties. Since we are using EKF in combination with a robust controller, which explicitly accounts for model uncertainty, we can expect the effect of model uncertainty on the state estimates obtained from EKF to be minimal.
Remark 13. To further refine our assumption concerning \( w \) term so that it more accurately reflects \( W_s(a,\theta) \), in the future we will consider sigma-point filter [109] for the proposed controller design.

7.5 Design of Robust controller using EKF state estimates

We employ a robust feedback controller to design a dynamic output feedback controller for the system of Eqs.7.1-7.3 using the state estimates of Eq.7.7 obtained using EKF. In [111] robust state & output feedback controllers were synthesized via Lyapunov’s direct method for quasi linear parabolic PDEs with slowly varying uncertain variables. Under certain assumptions, the designed controllers enforced closed loop stability and attenuated (asymptotically) the effect of process uncertainties on the output. In their work the authors have not utilized the available process model to provide better state estimates; the issue of measurement noise was not considered. We are currently investigating the impact of a combination of dynamic output feedback (using the state estimates obtained from EKF) and the robust controller on the closed loop performance of the system. We will briefly present the structure of the employed robust controller; a detailed overview can be found in [111]. The dynamic robust feedback control law used is of the following
\[ u = d(q) + Q(q)\dot{v} + r(q, t), \]

where \( d(q) \), \( r(q, t) \) are vector functions, \( Q(q) \) is a matrix, \( q \) is the vector of measured outputs and \( \dot{v} \) is a vector function of the external reference inputs and their time derivatives. The component \( d(q) + Q(q)\dot{v} \) in the controller is responsible for output tracking and is based on differential geometry; the component \( r(q, t) \) is responsible for the asymptotic attenuation of the effect of the uncertain variables on the outputs of the closed-loop slow system and is designed using Lyapunov arguments. Note that for \( r \equiv 0 \), the controller of Eq.7.11 attains the form of feedback linearizing controllers of [112, 1].

### 7.6 Application to diffusion reaction process

In this section, we use the above methodology to stabilize an unstable steady state of a typical diffusion-reaction process with a time varying uncertainty. Specifically, we consider a zero-th order exothermic reaction \( A \rightarrow B \) taking place on a thin catalytic rod. The temperature of the rod is adjusted by means of an actuator (by cooling the rod) located along the length of the rod. Assuming that the reactant A is present in excess, the spatial profile of the dimensionless temperature of the
rod is described by the following parabolic PDE.

\[
\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \beta_{T,n}(e^{-\gamma/(1+x)} - e^{-\gamma}) + \ldots
\]

\[+ \beta_U (b(z)u(t) - x) + e^{-\gamma/(1+x)}\theta(t) \tag{7.12}\]

Subject to the following boundary condition and initial conditions:

\[x(0, t) = 0, \ x(\pi, t) = 0, \ x(z, 0) = 0.05, \ x(z, 0) = 0.05 \tag{7.13}\]

The dimensionless rod temperature is given as \(x = \frac{T - T_0}{T_0}\), where \(T\) is the temperature of the reactor in \(^0\text{K}\) and \(T_0\) is the reference temperature used. The domain of this process is \(\Omega = [0, \pi]\); \(z\) is the spatial coordinate along the axis of the rod, \(\beta_{T,n}\) denotes the nominal dimensionless heat of reaction, \(\gamma\) denotes the dimensionless activation energy, \(\beta_U\) denotes the dimensionless heat transfer coefficient, \(u(t)\) denotes the magnitude of actuation, \(\theta(t)\) denotes a time varying uncertainty in the dimensionless heat of reaction and \(b(z)\) accounts for the spatial profile of the actuator. A spatially distributed actuation with \(b(z) = \sqrt{(2/\pi)}\sin(z)\) was considered. The nominal values of the parameters were \(\beta_{T,n} = 50, \ \gamma = 4, \ \text{and} \ \beta_U = 2\). In this numerical study, the slowly varying uncertainty in the process model (Eqs. 7.12-7.13) is assigned to be \(\theta(t) = \beta_{T,n}\sin(0.524t)\).
7.6.1 Estimator implementation

We now present the effectiveness of EKF by estimating the open loop profile of Eq.7.12 with no process uncertainty under a Gaussian white measurement noise $v \sim N(0,0.03)$ and the process is simulated until time $t_{final} = 15$. The value of the parameters used are: initial error covariance $P_0 = 0.15I$, where $I$ is an identity matrix of appropriate dimensions. We assume the availability of two noisy measurement sensors at positions $L/4$ and $3L/4$ on the catalytic rod of length $L$. The performance of the estimator was evaluated by calculating the 2-norm of the estimation error (between the states predicted using EKF and actual states evaluated from numerical simulation of Eqs.7.12-7.13) and trace of the error covariance matrix $P$, using Monte-Carlo simulations. Fig. 7.2 presents the average of 50 Monte-Carlo simulations, it can be observed that the trace matrix $P$ and the estimation error very rapidly converges very close to zero. In other words the performance of the estimator improves as it gets more information from the sensors.

7.6.2 Controller implementation

Fig. 7.3 presents the open-loop evolution of the PDE (with process uncertainty) for $u(t) = 0$. It can be observed that the open-loop process behavior is unstable even though the process noise and its derivatives, $\theta(t)$ and $\dot{\theta}(t)$, are kept small. The initial operating point $x(z,0) = 0$ is therefore an unstable one. The control objective
in this case is to design a dynamic output feedback controller that stabilizes the rod temperature to the spatially open-loop unstable steady state.

We utilize the dynamic output robust feedback controller based on the reduced order system Eq.7.7 to achieve the above objective. In the present case study the reduced order system was obtained using a truncated series expansion of $x$, using $N = 11$ eigenfunctions. Subsequently, the reduced order model was utilized to design the robust controller of the generic form of Eq.7.11. The expression for one of the parameters responsible for accounting model uncertainty ($r(\hat{a}, t)$) is presented below

$$r(\hat{a}, t) = -\chi \frac{\hat{a}}{|\hat{a}|} + \Lambda \int_{0}^{\pi} \phi_1(z) e^{\left[1+\sum_{i=1}^{N} a_i(t) \phi_i(z)\right]} d z$$  \hspace{1cm} (7.14)
where $\chi$, $\Lambda$ are adjustable control parameters and $\phi_1(z)$ is the first eigenfunction of the spatial differential operator in Eq.7.12. It should be noted that the controlled output chosen to stabilize the process (as the first eigenvalue gives us the desired separation required in Eq.7.3.1) was the first mode i.e $y_c = \hat{a}_1$. Also note that the expression for nonlinear time varying bounding function which captures the size of the uncertain terms in the system

$$c_0(\hat{a},t) = \int_0^\pi \phi_1(z) e^{(1+\sum_{i=1}^N \hat{a}_i(t) \phi_i(z))} \, dz$$

is explicitly used in the formulation of $r(\hat{a},t)$. The control parameters for this case study were set at $\chi = 1.2$ and $\Lambda = 0.01$.

In Fig. 7.4, we present the closed loop performance of Eq.7.12 using EKF and controllers designed with $r \equiv 0$. The closed loop performance clearly is unacceptable as the controller does not account for model uncertainty; moreover state estimates by EKF are not reliable in this case as EKF is sensitive to errors due to the unaccounted model dynamics. The closed loop performance of the PDE system (Eq. 7.12) using the robust controllers is presented in Fig. 7.5. The performance of robust controller far exceeds the performance of controllers designed using feedback linearization as it explicitly accounts for the process uncertainty.

The performance of the closed loop estimator is evaluated using Monte-Carlo simulations (Fig. 7.6). It is observed that as the process evolves, the trace of the error covariance matrix relaxes to zero and the estimation error remains bounded.
Figure 7.3. Open-loop profile of Eq. 7.12 with measurement noise very close to zero. We also observe that as the robust controller accounts for model uncertainty the estimates from EKF become reliable. Fig. 7.7 presents the temporal profile of control action used. The chattering observed in the control action is due to measurement noise present in the sensors. This is confirmed by simulating the process under no measurement noise and again stabilizing the open-loop unstable operating point of the process using the designed robust controller. The smooth control action obtained (used) in this simulation is presented in Fig. 7.8.
7.7 Conclusions

The issue of utilizing dynamically estimated states (obtained from EKF using limited noisy process measurements) in a robust controller, that addresses model uncertainty, was investigated. We initially found a finite dimensional approximation of the PDE system employing Galerkin’s method, then an EKF was designed to estimate the system states from the available noisy measurement data. Em-
ploying these estimated states along with a robust controller resulted in a reliable estimation of the system states, in presence of model uncertainty and simultaneously achieved the necessary control objective. The methodology was applied to a representative example wherein the control objective was to control temperature in a catalytic rod where an exothermic reaction occurs. It was observed from numerical simulations that the 2-norm of the estimation error asymptotically goes to zero as more measurements from the process was made available to the estimator. The robust dynamic output feedback controller, using the states estimated through EKF, was found to successfully stabilize the process around an open-loop unstable steady-state.

Figure 7.5. Closed-loop estimated surface profile of Eq.7.12 using robust controller of Eq.7.11.
Figure 7.6. Temporal profile of the 2-norm of the error between true and estimated states and the trace of $P$ obtained during the closed-loop operation of Eq.7.12 with the robust controller.
Figure 7.7. Control action needed to stabilize Eq.7.12 using robust controller; with measurement noise.

Figure 7.8. Control action needed to stabilize Eq.7.12 calculated using robust controller; no measurement noise.
8.1 Conclusions

The present thesis has focused on development of the novel methodology (called the adaptive proper orthogonal (APOD) methodology) for the systematic and practical solution of process control problems that arise in systems that are mathematically modeled by dissipative partial differential equations (PDEs). These equations arise naturally during the modeling of wide variety of industrially important processes such as transport-reaction and fluid flow systems (e.g. plasma-enhanced chemical vapor deposition, Czochralski crystal growth and many other fluid-dynamical sys-
tems and biological systems). The present research objective is strongly motivated by the need for new process control techniques that accommodates for high product quality specifications and stringent safety and environmental regulations that arise in modern chemical and materials processes. By bringing together tools from chemical engineering, control theory, applied mathematics, we have developed an effective methodology called the APOD. This methodology updates the reduced order model (ROM) of the PDE system in a recursive manner thus avoiding the hurdle of generating a fully excited data set for the formulation of the ROM. Furthermore, the conceptual foundation of APOD allows for the robust and fast construction of a ROM, contrary to simple techniques such as repeatedly applying POD. Effectiveness of the APOD methodology has been evaluated for different scenarios like restricted availability of sensor measurements, model uncertainty, actuator constraints and gappy measurements. The developed methodology is subsequently applied, via numerical simulations, on transport reaction processes (exothermic catalytic reaction in a reactor), fluid flow processes modeled by FHN and KS equations. Specifically the main contributions of this thesis can be summarized as follows:

**Development of APOD methodology.** In chapter 2, we developed the adaptive model reduction methodology called the “adaptive proper orthogonal decomposition (APOD)”. This methodology updates the basis functions (constructed using an adhoc dataset), using the closed-loop data that becomes available during
the process evolution as opposed to all other proper orthogonal decomposition-based reduction approaches that are based on open-loop snapshots, and thus, these snapshots and the resulting basis functions account for the impact of controller functional form on the process.

State feedback control. In chapter 3, we utilized the APOD methodology and designed state feedback controller using geometric control techniques. Initially, a ROM was formulated using the existing data-set and this ROM was updated during the online evolution of the process. The effectiveness of the proposed methodology was successfully demonstrated through the test case of a catalytic rod. ROMs designed and updated using APOD successfully stabilized the process even in the presence of parametric uncertainty.

Output feedback control. The controller design methodology presented in chapter 3 requires continuous availability of full state measurements of the process. The availability of such full state measurements is usually restricted due to limited availability of sensors. In chapter 4, we alleviated these concerns by extending the applicability of the APOD methodology to cases where the availability of distributed sensors is restricted. On the assumption of periodic availability of complete snapshots and continuous availability of point measurements from a restricted number of sensors Lyapunov based output feedback controllers were designed. These controllers were then applied on two test cases of dissipative PDEs, a system of coupled nonlinear one-dimensional PDEs (the FitzHugh-Nagumo equa-
tions) widely used to describe the formation of patterns in reacting and biological systems and the Kuramoto-Sivashinsky equation, a model that describes incipient instabilities in a variety of physical and chemical systems. The controllers designed using the APOD formulated ROM’s successfully stabilized the process at the desired set point on the other hand controllers designed using the ROM (with out using APOD) resulted in an unstable process. The computation acceleration achieved using the APOD methodology was around 350% as compared to repeatedly employing a standard POD methodology.

**Model predictive control.** In chapter 5, we further extended the applicability of the APOD methodology by designing model predictive controllers that address the issue of actuator constraints. Initially, a ROM was formulated using the available process data in the POD methodology. An optimization problem was then solved (with a receding control horizon) to compute the optimal control action. We updated the ROM, upon the availability of closed-loop measurements from the process (from implementing the computed control actuation on the plant) thus eliminating the requirement for formulation of a completely excited data ensemble as required for controllers designed based on POD methodology. The effectiveness of the methodology was successfully demonstrated using representative examples of heat conduction taking place in a catalytic rod and fluid flow modeled by Kuramoto-sivashinsky equation. The designed MPC controller successfully stabilized the process.
**Output feedback control using gappy snapshots.** The approach presented in chapter 4 still requires either continuous availability or periodic availability of full state measurement sensors. The availability of such full state measurements is usually restricted due to high cost and limited availability of sensors. We addressed these concerns in chapter 6 by expanding on ideas from chapter 4, designing output feedback controllers using only periodic partial distributed sensors and restricted continuous point measurement sensors. Initially, the partial measurements available from the sensors are reconstructed using a gappy iterative procedure. These reconstructed measurements were then utilized for derivation and update of ROMs using APOD methodology. The efficient recursive ROM updates by APOD allowed us to use low-dimensional model while designing controllers, thus resulting in computational savings. The proposed methodology was successfully used to achieve the closed-loop stabilization of process mathematically modeled by Kuramoto-Sivashinksy equation.

**Robust output feedback control design in presence of sensor noise.** In chapter 7, we presented the results of our initial efforts in addressing the issue of sensor noise and model uncertainty by designing robust controllers based on states estimated using extended Kalman filters (EKF). We assume that the availability of spectral basis functions by assuming the spatial operator of the PDE model to be linear. We then formulated a ROM of the PDE system employing Galerkin’s method, an EKF was then designed to estimate the system states from the noisy
measurement data. Employing these estimated states along with a robust controller resulted in a reliable estimation of the system states, in presence of model uncertainty and simultaneously achieved the necessary control objective. This promising approach was successfully tested on a representative example wherein the control objective was to regulate the temperature in a catalytic rod, where an exothermic reaction occurs.

8.2 Future work

Even though the current methodology has numerous advantages there are still many open issues to pursue. These will address some limitations such as accounting for model uncertainty, sensor noise and explicit 'apriori' ROM evaluation.

Adaptive model predictive control. In chapter 5, we presented the initial results for design of model predictive controllers using ROM’s and the APOD methodology. Using ROM’s instead of a full scale PDE model will significantly improve the computational speed, however the designed controller is as good as the ROM used. Consequently, there is a need for 'apriori' evaluation of the accuracy of these models. Ideas like the trust region approach [113] could act as starting point for research in this direction. These ideas could be tested on bigger test case e.g., nonlinear CFD model of industrial glass furnace [60].

Data reconstruction and control using moving distributed sensors. The use of mobile sensors has been receiving attention as it brings forth an added di-
mension to the efficient use of sensing and actuating devices as regards to reduction in power consumption, improved performance and efficient monitoring [114]. An important problem for the future development of this area is control and optimization of nonlinear distributed process using the information obtained from moving sensors. This would require the development of a methodology for the construction of spectral basis functions using the available moving sensor data and consequently utilizing the basis functions in the design of controllers for the nonlinear PDE processes. Some of the ideas on estimation of distributed process using mobile sensors [115] can potentially be utilized as a starting point for exploring this topic.

**Sensor noise and model uncertainty.** In chapter 7, we used spectral basis functions to address the issue of sensor noise and parametric uncertainty for PDE systems with linear spatial operators. An interesting extension would be to expand this work for the case of PDE systems with nonlinear spatial operators by employing the APOD methodology. This would be touching an important topic of adaptive Kalman filters and would bring forth interesting issues of effect of sensor noise on the basis functions computed using the APOD methodology.
Proofs of Lemma 1:

The eigenvalues of \( c_q = Q C_N Q \) is the subset of eigenvalues of \( C_N \) that correspond to the eigenmodes which belong to subspace \( Q \).

*Proof.* Since \( C_N \) is symmetric and positive definite, there exists a real block diagonal decomposition of \( C_N \),

\[
C_N = W J W^{-1},
\tag{A.1}
\]

where

\[
W = (W_1, W_2), W_1 \in \mathbb{R}^{N \times m}, W_2 \in \mathbb{R}^{N \times (N-m)},
\tag{A.2}
\]
and

\[ J = \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix}, \quad J_1 \in \mathbb{R}^{m \times m}, \quad J_2 \in \mathbb{R}^{(N-m) \times (N-m)}. \]  \hspace{1cm} (A.3)

The columns of \( W_1 \) and \( W_2 \) form the bases for the subspaces associated with \( \lambda_1, \ldots, \lambda_m \) and \( \lambda_{m+1}, \ldots, \lambda_N \) respectively. The block \( J_1 \) is associated with the eigenvalues \( \lambda_1, \ldots, \lambda_m \) and \( J_2 \) contains the block for \( \lambda_{m+1}, \ldots, \lambda_N \). By definition the range space of \( W_1 \) is \( \mathbb{P} \), which implies \( QW_1 = 0 \).

\[ C_N W_2 = W_2 J_2 \]  \hspace{1cm} (A.4)

and

\[ QW_2 J_2 = QC_N W_2 = QC_N (P W_2 + Q W_2) = QC_N Q W_2. \]  \hspace{1cm} (A.5)

Here we used the fact that \( QC_N P = 0 \). (Since \( P \) is an orthogonal projector we know that \( P^2 = P \). As a result we can write \( C_N P = P C_N P \), and the result follows from the fact that \( QP = 0 \)). Let \( V = [W_1, QW_2] \), and use \( Q^2 = Q \), \( QW_1 = 0 \) to get

\[ QC_N Q V = QC_N Q ([W_1, QW_2]) = (W_1, QW_2) \begin{pmatrix} 0 & 0 \\ 0 & J_2 \end{pmatrix}. \]  \hspace{1cm} (A.6)

Since \( W \) is nonsingular, it can be shown that \( V \) is also nonsingular.

\[ QC_N Q = V \begin{pmatrix} 0 & 0 \\ 0 & J_2 \end{pmatrix} V^{-1}. \]  \hspace{1cm} (A.7)
Since the eigenvalues of \( J \) are \( \lambda_{m+1}, \ldots, \lambda_N \) is a subset of eigenvalues of \( C_N \), the result follows.
Proof of Lemma 2: Consider the parabolic PDE system in Eq. 6.7, for which assumptions 1, 2 & 3 hold. Also consider the nonlinear output feedback controller that is designed based on Eq. 6.20:

\[ u = -k(x_s, c_0) L_{B_s} V_{\hat{w}}(x_s) \]  \hspace{1cm} (B.1)

where

\[ k(x_s, c_0) = \begin{cases} c_o + \frac{L_{F_s} V_{\hat{w}}(x_s) + \sqrt{(L_{F_s} V_{\hat{w}}(x_s))^2 + (L_{B_s} V_{\hat{w}}(x_s))^2}}{L_{B_s} V_{\hat{w}}(x_s)}}, L_{B_s} V_{\hat{w}}(x_s) \neq 0 \\ c_o \hspace{2cm} L_{B_s} V_{\hat{w}}(x_s) = 0 \end{cases} \]
\[ \tilde{x}_s = S_m^+ y_m, \quad F_s = A \tilde{x}_s + f_s(\tilde{x}_s, 0), \quad L_F, V_{\tilde{w}} = \frac{\partial V_{\tilde{w}}}{\partial \tilde{x}_s} F_s, \quad L_B, V_{\tilde{w}} = \frac{\partial V_{\tilde{w}}}{\partial \tilde{x}_s} B_s, \quad V_{\tilde{w}}(\tilde{x}_s(t_i)) = \frac{\zeta}{2} \tilde{x}_s(t_i)^T \tilde{x}_s(t_i) \]

where \( \zeta \) is a periodically updated used-defined parameter. Then the controller in Eq. B.1 asymptotically stabilizes the system in Eq. 6.7.

**Proof.** Initially, we use Lyapunov arguments to prove that the closed-loop system of Eq. 6.20 is asymptotically stable. We then utilize multiple Lyapunov function analysis (from hybrid systems theory) to show that the systems remains stable during the periodic updates of the ROM. The above two parts are then synthesized together in part 3 to establish the asymptotic stability of the closed-loop system Eq. 6.7.

**Part-1:** *(Closed loop stability of system, Eq. 6.7, between ROM revisions)*

Consider system in Eq. 6.20 of dimension \( \tilde{w} \) for which assumption 5 holds. Evaluating the time-derivative of the Lyapunov function candidate

\[ V_{\tilde{w}}(\tilde{x}_s(t)) = \frac{\zeta}{2} \tilde{x}_s(t)^T \tilde{x}_s(t) \]  

(B.2)

along the trajectories of the system, we obtain

\[ \dot{V}_{\tilde{w}} = L_F V_{\tilde{w}}(\tilde{x}_s(t)) + L_B V_{\tilde{w}}(\tilde{x}_s(t)) u \]  

(B.3)

where \( \tilde{x}_s = S_m^+ y_m = S_m^+ S_m(x_s + x_f) = x_s \) and \( \zeta \) is an appropriately chosen positive number (discussed later). Substituting the controller in Eq. B.1 into the above
equation yields, after some algebraic manipulations.

\[
\dot{V}_{w}(\bar{x}_s(t)) = -c_0\left(L_B V_{\bar{w}}(\bar{x}_s(t))\right)^2 - \sqrt{(L_F V_{\bar{w}}(\bar{x}_s(t)))^2 + (L_B V_{\bar{w}}(\bar{x}_s(t)))^4} \tag{B.4}
\]

From the above equation, it is clear that \(\dot{V}_{w}(\bar{x}_s) < 0\). Therefore, given an initial condition \(|\bar{x}_s(0)| < \delta_s\), the closed-loop system of Eq. 6.20 is asymptotically stable under the controller of Eq. B.1 [Theorem 4.1] [27].

Substituting the controller in Eq. B.1 into Eq. 6.19, the closed-loop PDE system of Eq. 6.7 can be written in the following form:

\[
\frac{dx_s}{dt} = A_s(x_s, x_f) - B_s k(\bar{x}_s, c_0) L_B V_{\bar{w}}(\bar{x}_s) + f_s(x_s, x_f)
\]

\[
\frac{dx_f}{dt} = A_f(x_s, x_f) - B_f k(\bar{x}_s, c_0) L_B V_{\bar{w}}(\bar{x}_s) + f_f(x_s, x_f)
\]

Employing assumption 3 the above system can be expressed in the following singular-perturbation form:

\[
\frac{dx_s}{dt} = A_s(x_s, x_f) - B_s k(\bar{x}_s, c_0) L_B V_{\bar{w}}(\bar{x}_s) + f_s(x_s, x_f)
\]

\[
\epsilon_s \frac{dx_f}{dt} = \epsilon_s A_f(x_s, x_f) + \epsilon_s f_f^*(x_s, x_f)
\]

where \(\epsilon_s\) is a small number quantifying the separation between the dominant and non-dominant eigenmodes of the spatial operator and \(f^*\) is defined as \(f_f^*(x_s, x_f) = -B_f k(\bar{x}_s, c_0) L_B V_{\bar{w}}(\bar{x}_s) + f_f(x_s, x_f)\). By the construction of the controller in Eq. 6.24 the term \(f_f^*(x_s, x_f)\) does not contain terms of the form \(O(1/\epsilon_s)\).
Then, introducing the fast time-scale \( \tau = t/\epsilon_s \) and setting \( \epsilon_s = 0 \), we obtain the following infinite-dimensional fast subsystem from the above equation.

\[
\frac{\partial x_f}{\partial \tau} = A_{f\epsilon_s} x_f
\]  

(B.7)

where \( A_{f\epsilon_s} = \epsilon_s A_f \). Note that \( A_{f\epsilon_s} \) is of order \( O(1) \) and the above system is locally exponentially stable by construction of system in Eq. 6.19 and assumption 4. Consequently after a period of time, \( t_f \), during which the fast dynamics relax to zero, we may assume \( x_f = 0 \) [25]. The closed-loop PDE system in Eq. B.6 then reduces to the finite-dimensional loop slow system of Eq. 6.20, which was already proven to be asymptotically stable.

Part-2: (Stability of hybrid system)

In general during the closed-loop process evolution, the periodic updates of ROM using APOD alters the underlying dominant subspace \( \mathbb{P} \). As a result the Lyapunov function \( V_{\tilde{w}} \) and the controller in Eq. B.1 are redesigned based on the updated ROM. Consequently, the stability aspects of the closed-loop system needs to be assessed using hybrid systems theory. To this end multiple Lyapunov functions are introduced of the same general form of B.2. Under the assumption of finite number of ROM updates and finite time interval between ROM updates the following additional restriction on the Lyapunov functions guarantees the switched system
to be Lyapunov stable [Theorem 3.2][116]:

\[ V_{\tilde{w}_1}(\tilde{x}_s(t_i)) < V_{\tilde{w}_2}(\tilde{x}_s(t_{i-1})) \]  

(B.8)

where \( i > 1 \) and \( V_{\tilde{w}_1}(\tilde{x}_s(t_i)) \) corresponds to the value of the Lyapunov function at the beginning of the interval \( i \) for which the ROM is of dimension \( \tilde{w}_1 \).

Part-3: *(Combined analysis)*

The above condition is directly enforced in the APOD methodology during the ROM updates through the appropriate choice of \( \zeta \) and \( \epsilon \) and the periodic update of \( \zeta \) value so that inequality Eq.B.8 remains valid. This update can be automatically enforced. Thus using the above condition under assumption 4 and the fact that the closed-loop system in Eq. B.6 is asymptotically stable, we can conclude that the designed controller in Eq. B.1 asymptotically stabilizes the system in Eq. 6.7. \( \square \)


ternational Symposium on Advanced Control of Chemical Processes, Kyoto, Japan, pp. 203–214.


[114] **Demetriou, M. A.** (2008) “Guidance of a moving collocated actuator/sensor for improved control of distributed parameter systems,” in *Pro-
ceedings of 47\textsuperscript{rd} IEEE Conference on Decision and Control, Cancun, Mexico, pp. 215–220.


Vita
Sivakumar Pitchaiah

Education


B.Tech., Chemical Engineering, Sri Venkateswara University, Tirupathi, May 2003.

Awards and Honors

1. Advanced process control research internship position with core process controls group with ExxonMobil chemical company (EMCC), 2010.

2. Computing & Systems technology (CAST), AICHE; Graduate Travel Award, Philadelphia, 2008.


Journal Papers


