2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

Lumped-Parameter Model Development and Robust Control of Systems Governed by 2-D Parabolic Convection-Diffusion Equation

Mark Trudgen[1] Javad Mohammadpour[2]

Abstract— **In the present paper, proper orthogonal decomposition (POD) method is employed to derive a lumped-parameter model for systems governed by two-dimensional (2-D) parabolic convection-diffusion (PCD) equation. The POD method employs singular value decomposition (SVD) to explore the content of a data set in order to identify the most and least variation to choose lower-order basis functions that provide close approximations of the original data set. In this work, POD is utilized to determine a low-order model that is suitable for control design purposes; using the low-order model, an H_∞ controller is then designed to ensure closed-loop system stability and reference tracking. This control design framework is chosen since the low-order model presents both parametric uncertainty and unmodeled high frequency dynamics arising from the derivation of the low-order model. The loop-shaping method is adopted for the design a robust H_∞ controller to achieve desired tracking and disturbance rejection in the closed-loop system. The simulation results show that the robust controller designed on the basis of the low-order model provides satisfactory reference tracking performance for the system described by the full-order PCD model.**

I. Introduction

Recent developments in the control system community with respect to model order reduction techniques have generated interest in applying relevant tools to fluid dynamic systems governed by partial differential equations (PDEs) [1], [2]. Often, the fluid flows are described by high-dimensional and/or nonlinear equations, but the fully described fluid dynamical models are not suited for control design purposes [3], [4]. Fundamentally, we find the need to develop low-order models that can capture and accurately represent the system dynamics so that a controller of reasonable order can be designed [4].

Convection-diffusion equations are representative of a class of fluid dynamic systems that accurately describe a particular flow phenomenon, but are mathematically not suited for control design purposes [5]. Therefore, in order to arrive at lower-order models tractable for control design purposes, there is a need to use a suitable projection method [1]. Convection-diffusion processes are widespread in occurrence in many scientific and engineering fields. A few common applications are: pollutant dispersal, vorticity transport in the incompressible Navier-Stokes equations, atmospheric pollution, semiconductor equations, the Stephan problem on a variable mesh, and viscous compressible flow [5]. The authors in [6], [7] used finite element method to control the convection-diffusion equation; however, their primary focus was on the mesh constraints related to optimality. Here, we extend the control design problem governed by the convection-diffusion equation using a robust H_∞ controller.

A great deal of work developed for model order reduction techniques has come within the controls context, and many methods for model order reduction have been suggested [2]. Proper orthogonal decomposition (POD) has become a well-accepted technique to obtain optimal basis functions that lead to low-order models to accurately represent the original full-order models [1], [2], [3], [4], [8], [9], [10], [11]. POD was first introduced by Lumley [12] in the context of turbulence, and the method of snapshots was first suggested by Sirovich [13]. POD has the intrinsic property that it is completely data dependent, and the modal decomposition used does not assume prior knowledge of how the data is generated [8]. This property is advantageous as no *apriori* information is needed to choose an ideal set of basis functions.

In this paper, POD is utilized as the basis for modal decomposition of an ensemble of functions. Given a set of data that lies in a vector space, the problem is to find a subspace of fixed dimension that minimizes the error between the two vector spaces [1]. We describe an infinite-dimensional parabolic convection-diffusion (PCD) equation (see, e.g., [14]) and then represent it in the discrete time domain. Using the method of snapshots to align the data together, we reduce the problem to a singular value decomposition (SVD) analysis. SVD is a valuable tool for identifying dimensions inside data sets that have the most and least variation [15]. We then form a reduced-order model using Galerkin projection [3]. Lastly, we design an H_∞ controller and show that this low-order controller is a suitable choice to control the high-order PCD model as it is robust to the natural parametric uncertainties within the flow that the PCD equation describes, and is also robust to the unmodeled high frequency dynamics that arise due to the use of POD for model reduction [16].

This paper is organized as follows: Section II describes the two-dimensional parabolic convection-diffusion equation and the associated discretized state-space model. In Section III, the high-order model is reduced via the application of POD. Section IV describes the associated H_∞ controller design process and simulation results, and Section V draws conclusions.

[1]College of Engineering, The University of Georgia, Athens, GA 30602 USA. Email: mtrudgen@uga.edu.

[2]Corresponding author; College of Engineering, The University of Georgia, Athens, GA 30602 USA. Email: javadm@uga.edu.

II. Full-Order PCD Model and the Discretized Model

In this paper, we consider the two-dimensional parabolic convection-diffusion (PCD) equation as [14]

$$\omega_t = \mu(\omega_{xx} + \omega_{yy}) - c_1(x,y)\omega_x - c_2(x,y)\omega_y + b(x,y)u(t), \tag{1}$$

over the spatial domain x, $y \in [0,1]$, where ω is the velocity field [5], $u(t)$ is the source, which is treated as the control input, and the convection coefficients are

$$c_1(x,y) = -x * \sin(2\pi x)\sin(\pi y), \tag{2a}$$
$$c_2(x,y) = -y * \sin(\pi x)\sin(2\pi y), \tag{2b}$$
$$b(x,y) = 5 * \sin(\pi x)\sin(\pi y). \tag{2c}$$

The expression for $b(x,y)$ is valid for $x \geq 0.5$, and otherwise, $b(x,y) = 0$. The boundary conditions for (1) are given as

$$\omega_x(t,0,y) = 0, \tag{3a}$$
$$\omega(t,x,0) = 0, \tag{3b}$$
$$\omega(t,x,1) = 0, \tag{3c}$$
$$\omega(t,1,y) = 0. \tag{3d}$$

Finally, the initial condition is assumed to be

$$\omega(0,x,y) = 0. \tag{4}$$

A. Discretization of the PCD Model

The partial differential equation (PDE) in (1)-(2) along with the boundary conditions (3) is discretized using a forward time-center space (FTCS) discretization method to give

$$\begin{aligned}\frac{\omega_{i,j}^{k+1} - \omega_{i,j}^{k}}{\Delta t} = \mu\Bigg[&\frac{\omega_{i+1,j}^{k} - 2\omega_{i,j}^{k} + \omega_{i-1,j}^{k}}{(\Delta x)^2} \\ &+ \frac{\omega_{i,j+1}^{k} - 2\omega_{i,j}^{k} + \omega_{i,j-1}^{k}}{(\Delta y)^2}\Bigg] \\ &- c_1(x_i,y_j)\left[\frac{\omega_{i+1,j}^{k} - \omega_{i-1,j}^{k}}{2\Delta x}\right] \\ &- c_2(x_i,y_j)\left[\frac{\omega_{i,j+1}^{k} - \omega_{i,j-1}^{k}}{2\Delta y}\right] \\ &+ b(x_i,y_j)u^k,\end{aligned} \tag{5}$$

where Δx and Δy represent the discretization step size in spatial directions, and Δt is the time step; i and j represent the two spatial indices in the x and y dimensions, respectively, and k represents the time index. Following the discretization procedure, x_i and y_j, appearing in the convection coefficients and input distribution function, now represent the unique spatial location that corresponds to the current i^{th} or j^{th} step, respectively.

A simulation result of (5) at steady-state is shown in Figure 1, where $\Delta x = \Delta y = 1/30$, and $\Delta t = 0.0025$ sec. A nominal value of $\mu = 0.1$ is chosen for the diffusion coefficient, and the control input is chosen to be a unit step function. These conditions are chosen to examine the open-loop response to a constant input signal. In Figure 1 we see the boundary conditions and convection coefficients represented.

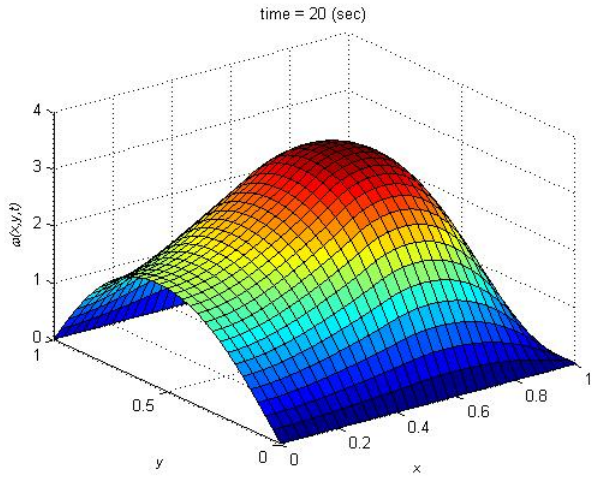

Fig. 1. Simulation of PCD equation.

B. State-space Model Derivation

Next, using the discretized model (5), we derive a discrete time state-space model of the form

$$\begin{aligned} z(k+1) &= Az(k) + Bu(k) \\ y(k) &= Cz(k). \end{aligned} \tag{6}$$

The state vector $z(k)$ consists of the discretized PDE variable ω, with n being the total number of steps in the x direction, and m the number of steps in the y direction; the state vector is

$$z(k) = [z_1(k), \ldots, z_{n\times m}(k)]^T. \tag{7}$$

The state vector $z(k)$ is arranged with respect to the spatial coordinates, and thus the structure of the elements of the state vector is as follows

$$z(k) = [\omega_{1,1}(k), \ldots, \omega_{1,m}(k), \ldots, \omega_{n,1}(k), \ldots, \omega_{n,m}(k)]^T. \tag{8}$$

Additionally, the state vector $z(k)$ is arranged with respect to the given boundary conditions in (3). The elements $[\omega_{1,1}^k, \ldots, \omega_{1,m}^k]$ in (8) correspond to the Neumann boundary condition in (3a). The elements $[\omega_{2,1}^k, \ldots, \omega_{n-1,m}^k]$ correspond to the Dirichlet boundary conditions given in (3b)-(3c) for the inner elements. Finally, the elements $[\omega_{n,1}^k, \ldots, \omega_{n,m}^k]$ correspond to the Dirichlet boundary condition given in (3d) for the outer elements. This same generalized structure, representing the given boundary conditions in (3), can be seen in the system matrix A. First, we introduce $\mu = \bar{\mu} + \Delta\mu$, where $\bar{\mu}$ is the nominal value of μ in order to write the A matrix, in which $\bar{\lambda}_x = \frac{\bar{\mu}\Delta t}{(\Delta x)^2}$, $\bar{\lambda}_y = \frac{\bar{\mu}\Delta t}{(\Delta y)^2}$, $\alpha_{i,j} = \frac{\Delta t c_2(x_i,y_j)}{2\Delta y}$, and $\beta_{i,j} = \frac{\Delta t c_1(x_i,y_j)}{2\Delta x}$. Additional discussion on $\bar{\mu}$ is given in Section 4. The corresponding state matrix is given in (9).

The input matrix B is prearranged by populating its

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\bar{\lambda}_y-\alpha_{2,1} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y+\alpha_{2,3} & 0 & \cdots & 2\bar{\lambda}_x & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \bar{\lambda}_y-\alpha_{3,2} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y+\alpha_{3,4} & \cdots & 0 & 2\bar{\lambda}_x & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & \bar{\lambda}_x+\beta_{n+2,2} & \cdots & \bar{\lambda}_y+\alpha_{n+2,m+1} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y-\alpha_{n+2,m+3} & \cdots & 0 & \bar{\lambda}_x-\beta_{n+2,2m+2} & 0 & \cdots & 0 \\
0 & 0 & \bar{\lambda}_x+\beta_{n+3,3} & \cdots & \bar{\lambda}_y+\alpha_{n+3,m+2} & 1-2\bar{\lambda}_x-2\bar{\lambda}_y & \bar{\lambda}_y-\alpha_{n+3,m+4} & \cdots & 0 & \bar{\lambda}_x-\beta_{n+3,2m+3} & \cdots & 0 \\
0 & 0 & 0 & \ddots & 0 & \ddots & \ddots & \ddots & 0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0_{n(n-1)+1,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0_{n(n-1)+1,m^2} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
0_{n^2,1} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0_{n^2,m^2}
\end{bmatrix}. \tag{9}$$

elements using (2c) as

$$B = \begin{bmatrix}
5\sin(\pi x_1)\sin(\pi y_1) \\
\vdots \\
5\sin(\pi x_1)\sin(\pi y_m) \\
\vdots \\
5\sin(\pi x_n)\sin(\pi y_1) \\
\vdots \\
5\sin(\pi x_n)\sin(\pi y_m)
\end{bmatrix}. \tag{10}$$

The C matrix, representing the system measurements, is formed by assuming that the sensor is located at the middle of the spatial variables range. To verify the state-space representation of the PCD model, i.e., (6), we used the same simulation conditions as before, i.e., with $\Delta x = \Delta y = 1/30$ and $\Delta t = 0.0025$ sec., and the corresponding outputs replicate.

III. Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) method delivers a basis for model decomposition to extract dominant trends and features [3]. POD is a linear procedure and its properties often make it the preferred tool to use in various applications as it is an efficient way to capture the dominant components of even an infinite-dimensional process [2]. Often, this information is captured with only a few modes [3]. POD can be applied to spatio-temporal models, e.g., (1), where the goal is to identify a low-dimensional subspace on which to construct a model via projection of the given state-space model in (6) [4]. Essentially, POD extracts a set of orthonormal basis functions [17]. First, to approximate any function of interest over a domain of interest, we write the ensemble into coefficients to be determined [1]. For our problem on hand, we start the POD procedure by writing

$$\omega(t,x,y) \approx \widehat{\omega}(t,x,y) = \sum_{k=1}^{M} \alpha_k(t)\varphi_k(x,y), \tag{11}$$

where φ_k's define the set of orthonormal basis functions, and α_k's denote the time-dependent coefficients. We employ the method of snapshots - originally suggested by Sirovich [13] - that is a numerical procedure. It solves an eigenvalue problem [3] and only requires an ensemble of appropriately organized data points [17]. The data needed is captured as $\omega_{snap} \in \mathbb{R}^{N\times K}$,

$$\omega_{snap} = \begin{bmatrix} \omega_1^1 & \cdots & \omega_1^K \\ \vdots & \ddots & \vdots \\ \omega_N^1 & \cdots & \omega_N^K \end{bmatrix}, \tag{12}$$

where N corresponds to the number of discretization steps and K corresponds to the number of snapshots. In the finite-dimensional case, POD reduces to an SVD problem. To see this connection, we first start with the data matrix and write it as an eigenvalue problem. For a complete discussion of the mentioned problem, see [3]. The SVD is written for ω_{snap} as

$$\omega_{snap} = \Phi\Sigma V^T. \tag{13}$$

Using SVD, one can recorrelate data sets to expose various relationships within the data [3]. Also, relationships within the data set are identified and ordered from the most variation to the least [15]. This is done by making use of SVD as

$$\omega_{snap} = \Phi\Sigma V^T = \begin{bmatrix}\Phi_r & \Phi_s\end{bmatrix}\begin{bmatrix}\Sigma_r & 0 & 0 \\ 0 & \Sigma_s & 0\end{bmatrix}\begin{bmatrix}V_r^T \\ V_s^T\end{bmatrix}. \tag{14}$$

The columns of Φ from the SVD form the set of basis functions $\{\varphi_1,\cdots,\varphi_N\}$. This type of projection captures the most *energy* for reduced model [3]. In (13), $\Phi \in \mathbb{R}^{N\times N}$ and $V \in \mathbb{R}^{K\times K}$ and the sizes of Φ_r, Σ_r, and V_r each correspond to the M dominant singular values chosen. These basis functions, called POD modes, are used to obtain accurate low-order dynamic models via Galerkin projection [17].

Next, we examine the singular values to produce a reduced-order model. A representation of how much energy is captured by the reduced-order model is given by the differences in the sum of the squared singular values (15). A high percentage of energy preserved is always desired, meaning a larger M, which indicates that the model retains more of the information contained in the original snapshots. The preserved energy percent (PEP) is defined as

$$PEP = 100 \times \frac{\sum_{i=1}^{M}\sigma_i^2}{\sum_{i=1}^{N}\sigma_i^2}. \tag{15}$$

Furthermore, to proceed with the analysis, we choose an input signal to the system. To gain insight into the system dynamics, we analyzed the Bode plot of (6) corresponding to the discretized model with $\Delta x = \Delta y$ = 1/30, and Δt = 0.0025 sec. This corresponds to a high-order model with 961 states. The Bode plot is shown in Figure 2, which represents multiple plots corresponding to $\mu \in [0.06, 0.1]$ as the flow develops.

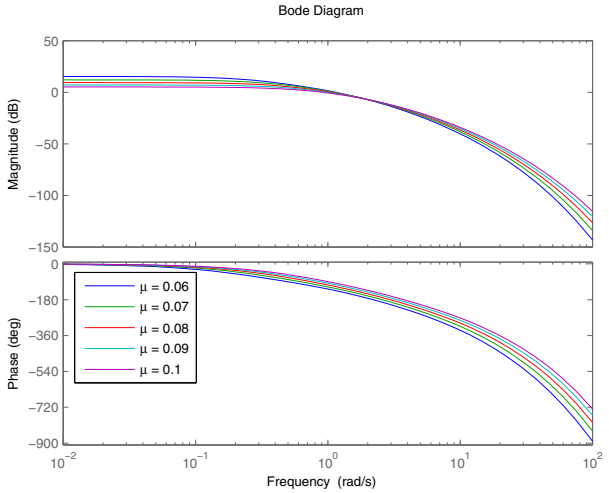

Fig. 2. Bode plot of the discretized model (6).

From the Bode plots depicted in Figure 2, it is observed that the system passes through low-frequency inputs, and that it attenuates high-frequency inputs. Therefore, to appropriately excite the system, we use a low-frequency sinusoidal input with multiple frequency components.

The system experiences greater excitation from a low-frequency multi-layered sinusoidal input rather than a single frequency input so we excited the system with the sum of sinusoidal signals with frequencies ranging from 0 to 50Hz with varying amplitudes. The frequency spectrum of the input signal is shown in Figure 3.

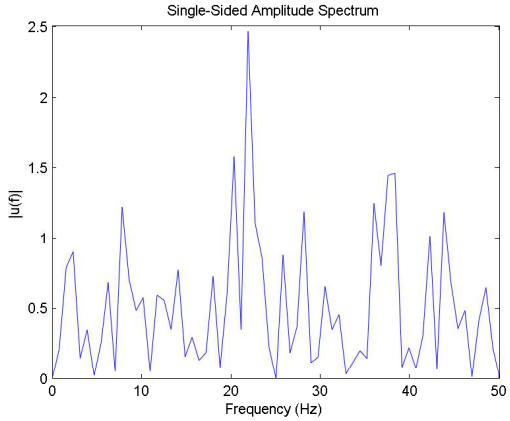

Fig. 3. Frequency content of the system input.

We used this analysis to determine the amount of captured energy that would be present in the reduced-order models. Figure 4 plots the preserved energy percent versus the number of singular values retained (i.e., the order of the reduced-order model or the number of retained POD modes). As observed in Figure 4, the first mode retains 78% of the energy, while using the first three modes retains approximately 94% of the energy, and the first five modes retain over 98% of the energy of the original high-order model.

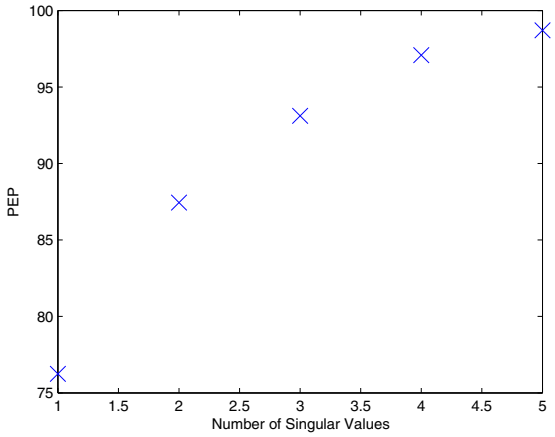

Fig. 4. Percentage of captured energy.

To obtain the reduced order state-space model, (6) is multiplied from both sides by the truncated orthonormal matrix $\Phi_r \in \mathbb{R}^{N\times M}$ as

$$\Phi_r^T z(k+1) = \Phi_r^T A z(k) + \Phi_r^T B u(k). \tag{16}$$

Recalling that $z(k)$ is the state vector of the original high-order approximation, the reduced-order state vector becomes

$$z_r(k) = \Phi_r^T z(k). \tag{17}$$

Since each element of $z_r(k)$ is a linear combination of the elements of $z(k)$, substituting (17) into (6) yields

$$\begin{aligned} z_r(k+1) &= A_r z_r(k) + B_r u(k) \\ y(k) &= C_r z_r(k), \end{aligned} \tag{18}$$

with

$$A_r = \Phi_r^T A \Phi_r \qquad B_r = \Phi_r^T B \qquad C_r = C\Phi_r. \tag{19}$$

We reduce the full-order model to 5^{th} order, which retains 98% of the *energy* of the high-order model. We note that in the low-order model, A_r, B_r, and C_r can be calculated offline. Next, the reduced-order model is simulated to steady-state with the same conditions (Δx = Δy = 1/30, Δt = 0.0025 sec, and $\mu = 0.1$) used to generate Figure 1. Simulation results illustrate a close agreement between the low and high-order models (see Figure 5).

IV. Controller Design and Closed-loop Simulation Results

In this section, we discuss the control design procedure in the robust H_∞ framework, for which we first introduce the sources of uncertainties and describe how to quantify them.

A. Representing Uncertainty

An investigation into the natural phenomena described by the PCD equation revealed that the diffusion term does change as flow develops. We assume the parameter μ to vary in the range $\mu \in [0.06, 0.11]$, with the nominal value of $\bar{\mu} = 0.1$. Indeed, many convection-diffusion equations have time-varying parameters but are instead described with lumped-parameter models for mathematical tractability [5], [18]. However, from a control design perspective, we realize that if we let $\mu = \bar{\mu} + \Delta\mu$, where $\bar{\mu}$ is the nominal value and $\Delta\mu$ represents uncertainty from the steady-state, we can still perform model reduction using $\bar{\mu}$ and later account for the uncertainty $\Delta\mu$. Additionally, the convection coefficients will have a level of uncertainty as (1) is derived from incompressible flow assumptions. In light of the competing trade-offs, we choose to derive a model using $\bar{\mu}$, and then use a robust controller design framework to account for the variations in the parameters. All physical systems realistically have a degree of uncertainty due to unmodeled high-frequency dynamics [16], and this is especially true for (18) as it is a low-order representation of (1). Therefore, we choose to model this as a multiplicative input uncertainty [16].

B. Robust Stability

To guarantee the closed-loop system stability for $\mu \in [0.06, 0.11]$ and in the presence of the model uncertainty due to the low-order approximation, we first write the lumped parameter multiplicative uncertainty as

$$G_p(s) = G(s)(1 + W_p(s)\Delta_p(s)), \tag{20}$$

where $G_p(s)$ represents the high-order model, $G(s)$ represents the low-order model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties [16]. We then simplify to the expression

$$\left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right| \leq |W_p(j\omega)|. \tag{21}$$

Figure 5 shows the Bode plot of the high-order model G_p, and the low-order system G at $\bar{\mu} = 0.1$.

Next, we determine an uncertainty weight W_p that satisfies (21) for various μ's varying in the interval $[0.06, 0.11]$. This yields the high-pass filter given in (22).

$$W_p(s) = \frac{1.85s + 0.1906}{s + 2.097}. \tag{22}$$

C. Loop Shaping for Control Design

Next, we use loop shaping method [16] to design a controller that guarantees tracking with zero steady-state error and a low control effort. The corresponding block diagram shown in Figure 6 depicts how disturbances and noise signals affect the closed-loop system.

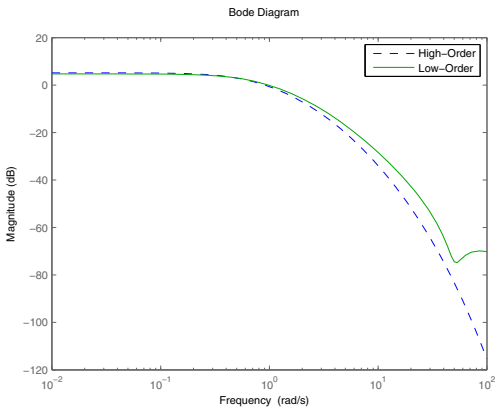

Fig. 5. Bode plot of the low-order vs high-order model.

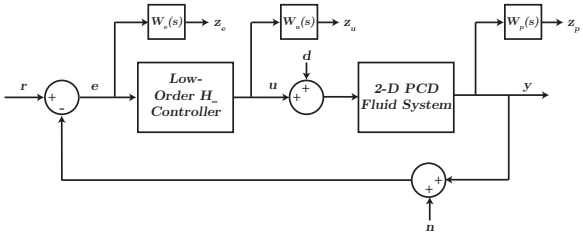

Fig. 6. Configuration of the closed-loop control system.

In standard loop shaping, three weights W_e, W_p and W_u, shown in Figure 6, need to be tuned to penalize the corresponding signals at specific frequency ranges. The weight W_e, selected to be a low pass filter, is tuned to eliminate the steady-state error, W_u to penalize large control inputs to the PCD model, and W_p is a high pass filter used to achieve robustness with respect to uncertainties. W_e is given in (23). The weight W_u is chosen to be a high-pass filter to penalize large controller outputs [19] and is given in (24). The design of the third weight W_p was discussed earlier and the filter is given by (22).

$$W_e(s) = \frac{0.15}{s + 1e^{-5}}, \tag{23}$$

$$W_u(s) = \frac{s + 0.8}{s + 1.1}. \tag{24}$$

D. Simulation Results and Discussion

To design the controller, we used the MATLAB Robust Control toolbox. To this end, we first converted the block diagram shown in Figure 6 into the corresponding linear fractional transformation (LFT) form. Next, we used the command *dhinflmi* to design an 8^{th} order controller. The sampling frequency of the simulation was 400Hz, and to simulate the effect of noise n, we corrupted the output signal y with a band-limited white noise with the power $N_P = 0.02$. The closed-loop system shown in Figure 6 also accounts for process disturbance corrupting $u(k)$. Finally, the high-order PCD model was discretized by considering $\Delta x = \Delta y = 1/30$, and from the corresponding measurement setup

given in (6), the C matrix was obtained. We then used a unit step input, and the simulation result is shown in Figure 7 that demonstrates the desired tracking over the range of variation of μ. The designed robust controller guarantees the closed-loop stability for the high-order PCD model with $\mu \in [0.06, 0.11]$.

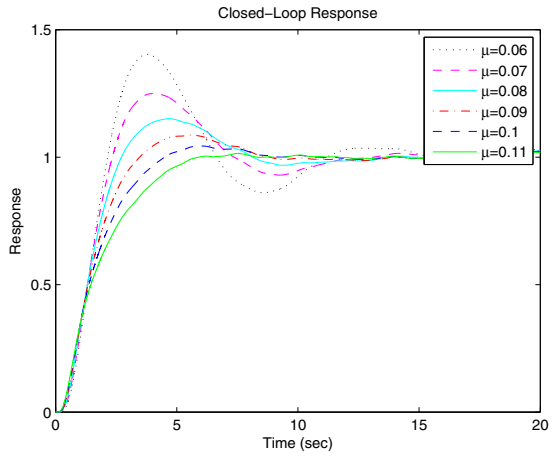

Fig. 7. Closed-loop response to a reference step input for the robust controller.

From Figure 7, it is observed that the robust controller yields a stable closed-loop system. In addition, the H_∞ controller yields an H_∞ norm of $\gamma \approx 1$ for the closed-loop system. A comparison between the nominal and robust controller for the PCD model at $\bar{\mu} = 0.1$ is shown in Figure 8. It shows improved performance by the nominal controller which is not surprising since it is designed specifically for $\bar{\mu} = 0.1$. The robust controller is in fact designed for a region, whereas the nominal controller is only designed for a single operating condition [16], and this allows for a better nominal closed-loop performance at the design point.

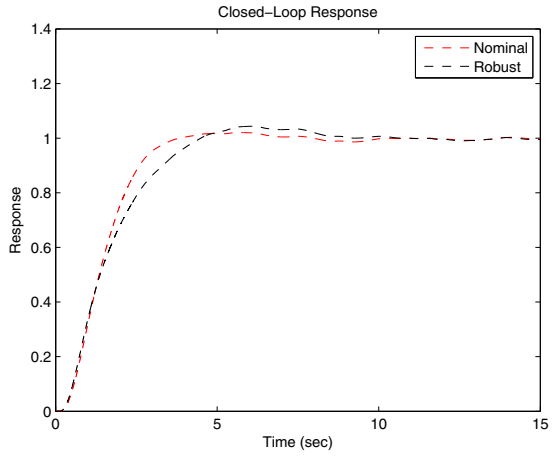

Fig. 8. Closed-loop response to a reference step input comparing the performance of robust vs nominal controller.

V. Concluding Remarks

In this paper, we employed the proper orthogonal decomposition (POD) method to approximate an infinite-dimensional parabolic convection-diffusion equation with a low-order lumped-parameter model while preserving a significant portion of the *energy* of the high-order discretized model. We then synthesized an H_∞ robust controller designed based on the reduced-order model. This was done by implementing loop shaping method to achieve zero steady-state tracking error and robustness against model uncertainties. Simulation results verified that the low-order robust controller could provide the desired tracking for the full-order PCD model with varying coefficients.

References

[1] C. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.

[2] A. Antoulas, D. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," *Contemporary mathematics*, vol. 280, pp. 193–220, 2001.

[3] P. Holmes, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 2012.

[4] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.

[5] K. W. Morton and K. Morton, *Numerical solution of convection-diffusion problems*. Chapman & Hall London, 1996, vol. 12.

[6] A. Valli, G. Carey, and A. Coutinho, "Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction–convection–diffusion processes," *International journal for numerical methods in fluids*, vol. 47, no. 3, pp. 201–231, 2005.

[7] R. Becker and B. Vexler, "Optimal control of the convection-diffusion equation using stabilized finite element methods," *Numerische Mathematik*, vol. 106, no. 3, pp. 349–367, 2007.

[8] M. Rathinam and L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1893–1925, 2003.

[9] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM Journal on Numerical analysis*, vol. 40, no. 2, pp. 492–515, 2002.

[10] S. Volkwein, "Model reduction using proper orthogonal decomposition," *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. see http://www. uni-graz. at/imawww/volkwein/POD. pdf*, 2008.

[11] D. A. Lawrence, J. H. Myatt, and R. C. Camphouse, "On model reduction via empirical balanced truncation," in *American Control Conference, Proceedings of the 2005*, pp. 3139–3144.

[12] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric turbulence and radio wave propagation*, pp. 166–178, 1967.

[13] L. Sirovich, "Turbulence and the dynamics of coherent structures. i-coherent structures. ii-symmetries and transformations. iii-dynamics and scaling," *Quarterly of applied mathematics*, vol. 45, pp. 561–571, 1987.

[14] J. R. Singler, "Balanced POD for model reduction of linear PDE systems: convergence theory," *Numerische Mathematik*, vol. 121, no. 1, pp. 127–164, 2012.

[15] K. Baker, "Singular value decomposition tutorial," *The Ohio State University*, 2005.

[16] S. Skogestad and I. Postlethwaite, *Multivariable feedback control*. John Wiley., 2005.

[17] S. M. Hashemi and H. Werner, "LPV modelling and control of Burgers equation," in *Proceedings 18th IFAC World Congress*, 2011.

[18] E. L. Cussler, *Diffusion: mass transfer in fluid systems*. Cambridge university press, 2009.

[19] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.