

Reduced-Order Model Design for Nonlinear Smart System Models

Stephen F. May ¹ and Ralph C. Smith ²

Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695

Abstract

The use of finite element or finite difference techniques to discretize nonlinear smart material system models can yield full-order numerical models that accurately characterize the system dynamics but do so at significant computational cost. This can preclude the use of these full-order models for uncertainty analysis, sensitivity analysis, system design, or real-time control implementation. In this paper, we discuss the construction of reduced-order system models using proper orthogonal decompositions (POD) with updates. Through the use of snapshots constructed from the full-order models, fundamental physics is retained while significantly improving efficiency for high-speed implementation.

Keywords: Smart materials, proper orthogonal decomposition

1. Introduction

Proper Orthogonal Decomposition (POD), also known as the Karhunen-Loeve procedure, is a basis reduction method that is quite popular among those seeking to implement real-time control on smart material structures (see [1] and the references therein). To implement the POD method, a sufficient amount of solution data for the given smart material system is needed. This data can either be collected experimentally or obtained through high fidelity numerical simulations. Solution data for the system is collected or simulated at several times, and the solution data at a given time is referred to as a snapshot. The POD method extracts characteristic information from the snapshots through an orthogonalization process. Because the basis elements are generated using either high fidelity simulations or actual data from the system, the POD method is able to represent systems very accurately with very few basis elements. This is a large advantage for the POD method over other numerical methods.

A difficulty that arises when trying to implement the POD method regards the manner one chooses to collect the set of snapshots. Using more snapshots than is necessary can lead to redundancy in information and makes the basis larger than is necessary. Not having enough snapshots can lead to gaps in the information about the system and lead to a reduction in accuracy. Generating an effective set of snapshots is an imprecise science and requires great care. It is possible to supplement a set of snapshots as more data becomes available, as is detailed in [7] and [9]. When new data becomes available (i.e., a new snapshot), the oldest snapshot is replaced with the new snapshot and the POD basis is recalculated. This way it is possible to add new information without greatly increasing the cost of computation of the solution.

To illustrate this process, we consider a flat, rectangular beam with one fixed end that has symmetric PZT patches attached to either side. We assume that the thickness of the beam is small relative to the length, and the beam vibrations occur transversely therefore we can use the Euler-Bernoulli beam model. Because of the discontinuities at the ends of the patches, we must consider the weak formulation of the beam model. We can employ a cubic spline basis to find an approximation to the solution of the system. This method of finding an

¹Email: sfmay@ncsu.edu

²Email: rsmith@eos.ncsu.edu, Telephone: (919) 515-7552

approximate solution to this model is shown to be accurate in [2]. Using the solution obtained with the spline basis as the solution data, we can compute a POD solution and compare the results to the full-order solution.

2. Beam Model

In this section, we describe the beam model and how to obtain the full-order approximation to the solution. For a complete derivation see [8]. We consider a flat, rectangular beam with a fixed end that has symmetric PZT patches attached on either side (see Figure 1). Let the fixed end of the beam be at $x = 0$ and the free end at $x = \ell$. To differentiate between the beam and the patch dimensions, we use a subscript b for the beam and p for the patch. Let $w(t, x)$ denote the transverse displacement and $f(t, x)$ the distributed out-of-plane force. Also let ρ , Y , and c denote the effective linear density, Young's modulus, and Kelvin-Voigt damping coefficients. The PZT patches are located on the region $[x_1, x_2]$ and the width of both the beam and the patch is b .

As in [8], balancing forces and moments yields

$$\rho \frac{\partial^2 w}{\partial t^2} + \gamma \frac{\partial w}{\partial t} - \frac{\partial^2 M}{\partial x^2} = f \quad (1)$$

where M denotes the total moment and γ is a proportionality constant relating viscous air damping and the transverse velocity. Here, the linear density function ρ is given by $\rho(x) = h_b b \rho_b + 2\chi_{pe} h_p b \rho_p$ where

$$\chi_{pe}(x) = \begin{cases} 1 & x \in [x_1, x_2] \\ 0 & x \notin [x_1, x_2]. \end{cases}$$

Because of the discontinuity at the patch ends, we need with the weak formulation of (1). As derived in [8], the weak formulation of the beam model is

$$\int_0^\ell \rho \frac{\partial^2 w}{\partial t^2} \phi dx + \int_0^\ell \gamma \frac{\partial w}{\partial t} \phi dx + \int_0^\ell YI \frac{\partial^2 w}{\partial x^2} \frac{d^2 \phi}{dx^2} dx + \int_0^\ell cI \frac{\partial^3 w}{\partial x^2 \partial t} \frac{d^2 \phi}{dx^2} dx = \int_0^\ell f \phi dx + \int_0^\ell k_p V(t) \frac{d^2 \phi}{dx^2} dx \quad (2)$$

where YI is a stiffness parameter, cI is an internal damping parameter, and k_p is an external coupling parameter, are all piecewise constants. The ϕ 's are test functions in the space $H_0^2(0, \ell) = \{\phi \in H^2(0, \ell) | \phi(0) = \phi'(\ell) = 0\}$.

2.1 Full-Order Solution Approximation

As in [8], we want to find an approximate solution using cubic B-splines. We will use this approximation as the full-order solution. To do this we partition the length of the beam, $h = \frac{\ell}{N}$ and $x_j = jh$ where $j = 0, \dots, N$. For $j = -1, 0, \dots, N, N + 1$, cubic B-splines are defined

$$\hat{\phi}_j(x) = \frac{1}{h^3} \begin{cases} (x - x_{j-2})^3, & x \in [x_{j-2}, x_{j-1}) \\ h^3 + 3h^2(x - x_{j-1}) + 3h(x - x_{j-1})^2 - 3(x - x_{j-1})^3, & x \in [x_{j-1}, x_j) \\ h^3 + 3h^2(x_{j+1} - x) + 3h(x_{j+1} - x)^2 - 3(x_{j+1} - x)^3, & x \in [x_j, x_{j+1}) \\ (x_{j+2} - x)^3, & x \in [x_{j+1}, x_{j+2}) \\ 0, & x \notin [x_{j-2}, x_{j+2}). \end{cases}$$

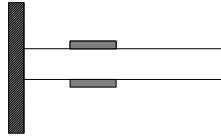


Figure 1: Cantilever beam with PZT patches.

To satisfy the initial conditions $w(t, 0) = \frac{\partial w}{\partial x}(t, 0) = 0$, and for ϕ to be in H_0^2 (which requires $\phi(0) = \phi'(0) = 0$), the basis functions are defined to be

$$\phi_j(x) = \begin{cases} \hat{\phi}_0(x) - 2\hat{\phi}_{-1}(x) - 2\hat{\phi}_1(x) & j = 1 \\ \hat{\phi}_j(x) & \text{otherwise.} \end{cases}$$

Approximate solutions of the model are of the form $w^N(t, x) = \sum_{j=1}^{N+1} w_j(t)\phi_j(x)$. Plugging into (2) and letting the basis functions operate as the test functions yields the system

$$\mathbf{M}\ddot{\mathbf{w}} + \mathbf{Q}\dot{\mathbf{w}} + \mathbf{K}\mathbf{w} = \mathbf{f} + V(t)\mathbf{b} \quad (3)$$

where $\mathbf{w}(t) = [w_1(t), \dots, w_{N+1}(t)]^T$. Here \mathbf{M} , \mathbf{Q} , and \mathbf{K} are the mass, damping, and stiffness matrices and they are defined by

$$\begin{aligned} [\mathbf{M}]_{ij} &= \int_0^\ell \rho\phi_i\phi_j dx \\ [\mathbf{Q}]_{ij} &= \int_0^\ell [\gamma\phi_i\phi_j + cI\phi_i''\phi_j''] dx \\ [\mathbf{K}]_{ij} &= \int_0^\ell YI\phi_i''\phi_j'' dx. \end{aligned}$$

The force vectors \mathbf{f} and \mathbf{b} are defined as

$$\begin{aligned} [\mathbf{f}]_i &= \int_0^\ell f\phi_i dx \\ [\mathbf{b}]_i &= \int_0^\ell \phi_i'' dx. \end{aligned}$$

The second order system (3) can easily be converted into the first order system (see [8]) and then, using an ODE solver, we can generate the approximate solution. This discretized beam model is verified experimentally in [2].

3. Proper Orthogonal Decomposition

Here we discuss reducing the size of the basis used for the simulation using Proper Orthogonal Decomposition (POD). For more detailed information see [1]. To implement the POD method, a large data set must be obtained through collection of data or through high fidelity simulation techniques. The POD method then converts that data into an optimal orthonormal set of N_p POD modes $\Phi_i, i = 1, \dots, N_p$. Here N_p is much smaller than N , the number of basis elements used in the high fidelity solution.

To construct the POD basis, we generate a set of N_s snapshots ($w^N(t_i, x), i = 1, \dots, N_s$) from the collected data. Each of the POD modes is a linear combination of the snapshots

$$\Phi_i(x) = \sum_{j=1}^{N_s} \alpha_j^i w^N(t_j, x). \quad (4)$$

To ensure that the basis elements resemble the data, we require the basis elements $\Phi_i(x)$ to maximize

$$\frac{1}{N_s} \sum_{j=1}^{N_s} |\langle w^N(t_j, \cdot), \Phi_i(\cdot) \rangle| \quad (5)$$

subject to $\langle \Phi_i, \Phi_i \rangle = \|\Phi_i\|^2 = 1$. This also ensures that the first POD element resembles the data the most, followed by the second and so on. To find the POD coefficients (α_j^i 's), we need to construct the covariance matrix (L) of the snapshots

$$[L]_{k,m} = \frac{1}{N_s} \langle w^N(t_k, \cdot), w^N(t_m, \cdot) \rangle, k, m = 1, \dots, N_s. \quad (6)$$

We then find the eigenvalues and eigenvectors of L and we order the eigenvalues from largest to smallest, $\lambda_1 \geq \dots \geq \lambda_{n_s}$. The coefficient α_j^i corresponds to the j th entry of the i th eigenvector. The basis is minimized by finding the smallest N_p such that $\frac{\sum_{i=1}^{N_p} \lambda_i}{\sum_{i=1}^{N_s} \lambda_i} \simeq 1$. Once N_p and the α_j^i 's are computed, the reduced-order basis vectors (Φ_i 's) can be constructed.

3.1 Computing the POD basis for the Beam Equation

To construct the POD basis, we collect or simulate the N_s snapshots Φ_i . Here we use a spline basis for the full-order simulation. The snapshots are of the form $\{w^N(t_j, x) = \sum_{i=1}^N w_i(t) \phi_i(x)\}_{j=1}^{N_s}$ where the ϕ_i 's are the spline basis functions and the w_i 's are the spline coefficients. The covariance matrix \mathbf{L} is defined as

$$\begin{aligned} [\mathbf{L}]_{k,m} &= \frac{1}{N_s} \langle w^N(t_k, \cdot), w^N(t_m, \cdot) \rangle = \frac{1}{N_s} \int_0^l w^N(t_k, x) w^N(t_m, x) dx \\ &= \frac{1}{N_s} \int_0^l \sum_{i=1}^N w_i(t_k) \phi_i(x) \sum_{j=1}^N w_j(t_m) \phi_j(x) dx \\ &= \frac{1}{N_s} \sum_{i=1}^N \sum_{j=1}^N w_i(t_k) w_j(t_m) \int_0^l \phi_i(x) \phi_j(x) dx. \end{aligned}$$

From the spline basis for the simulation, we already have the $N \times N$ matrix that approximates $\int_0^l \phi_i(x) \phi_j(x) dx$, as well as the coefficients w_i 's. Each element of \mathbf{L} is constructed with a vector-matrix-vector multiplication (i.e., L is constructed with simple matrix multiplication using existing matrices from the full-order approximation). The eigenvalues and eigenvectors of \mathbf{L} are ordered from largest to smallest $\lambda_1 \geq \dots \geq \lambda_{n_s}$. After finding the ordered eigenvalues and eigenvectors, we can find N_p , the α_j^i 's, and build the POD basis functions (Φ 's).

Once we have constructed the POD basis functions, we build the system matrices \mathbf{M} , \mathbf{K} , and \mathbf{C} . The matrices for the reduced-order system are constructed with the matrices from the full-order simulation. We will refer to the full-order system matrices as \mathbf{M}_{full} , \mathbf{K}_{full} , and \mathbf{C}_{full} . The mass matrix \mathbf{M} is defined as

$$\begin{aligned} [\mathbf{M}]_{ij} &= \int_0^l \rho \Phi_i \Phi_j dx \\ &= \int_0^l \rho \sum_{k_1=1}^{N_s} \alpha_{k_1}^i w^N(t_{k_1}, x) \sum_{k_2=1}^{N_s} \alpha_{k_2}^j w^N(t_{k_2}, x) dx \\ &= \rho \sum_{k_1=1}^{N_s} \sum_{k_2=1}^{N_s} \alpha_{k_1}^i \alpha_{k_2}^j \int_0^l w^N(t_{k_1}, x) w^N(t_{k_2}, x) dx \\ &= \rho \sum_{k_1=1}^{N_s} \sum_{k_2=1}^{N_s} \alpha_{k_1}^i \alpha_{k_2}^j \int_0^l \sum_{m_1=1}^N w_{m_1}(t_{k_1}) \phi_{m_1}(x) \sum_{m_2=1}^N w_{m_2}(t_{k_2}) \phi_{m_2}(x) dx \\ &= \rho \sum_{k_1=1}^{N_s} \sum_{k_2=1}^{N_s} \alpha_{k_1}^i \alpha_{k_2}^j \sum_{m_1=1}^N \sum_{m_2=1}^N w_{m_1}(t_{k_1}) w_{m_2}(t_{k_2}) \int_0^l \phi_{m_1}(x) \phi_{m_2}(x) dx \end{aligned}$$

We build the reduced-order mass matrix using the full-order mass matrix, the spline coefficients, and the POD coefficients. The POD mass matrix \mathbf{M} is constructed with matrix multiplication $[\mathbf{M}] = [\alpha_j^i][w_i(t)][\mathbf{M}_{\text{full}}][w_i(t)]^T [\alpha_j^i]^T$.

The damping and stiffness matrices as well as the force vectors for the reduced-order system are obtained in the same manner. The POD method has been used with the beam model to experimentally implement control designs as detailed in [5, 6].

4. POD with Updates

Here we discuss a method to update the POD basis as more snapshots become available. We will do this in the same manner as presented in [7, 9]. To use POD, a set of snapshots is required. It is difficult to ensure that all possible behaviors (modes) are captured in the initial set of snapshots. The POD basis that is generated only takes the information quantified in the set of snapshots into account. Hence the POD basis will only be as good as the snapshots used to create it. When new information becomes available, we would like to add that new information to the information that has already been collected and try to fine tune the POD basis.

When a new snapshot becomes available, we need to decide how to include the new information with the old. Just adding the new snapshot to the rest of the snapshots will increase the size of the covariance matrix. This requires solving another even larger eigenvalue-eigenvector problem which can be expensive and slow. In the case that solving the eigenvalue-eigenvector is too expensive to solve several times, we employ an algorithm from [7, 9] that doesn't require the solution of an eigenvalue-eigenvector problem, but still allows new information to be added as it becomes available.

When we initially compute a POD basis, we find a group of N_p dominant eigenvalues and corresponding eigenvectors of the covariance matrix \mathbf{L} . Letting \bar{P} be the subspace of dominant eigenvectors of \mathbf{L} , an orthonormal basis for \bar{P} is $\mathbf{V} = [v_1, \dots, v_{N_p}]$ where the v_i 's are the eigenvectors corresponding to the ordered eigenvalues $\lambda_1, \dots, \lambda_{N_p}$. We refer to the subspace containing the eigenvectors corresponding to the non-dominant eigenvalues as \bar{Q} . The projectors \mathbf{P} and \mathbf{Q} onto the subspaces \bar{P} and \bar{Q} are

$$\mathbf{P} = \mathbf{V}\mathbf{V}^T, \mathbf{Q} = \mathbf{I} - \mathbf{V}\mathbf{V}^T. \quad (7)$$

Here \mathbf{I} denotes the $Ns \times Ns$ identity matrix.

When a new snapshot becomes available, the oldest snapshot is replaced with the update. As a new snapshot is obtained, there are three possible ways that the dominant subspace \bar{P} can change. Another eigenvalue can become dominant and the dimension of \bar{P} can increase, an eigenvalue can go from the dominant subspace to the non-dominant subspace and the dimension of \bar{P} can decrease, or the dimension of \bar{P} can remain the same. In all cases, \mathbf{V} needs to be updated. Even if the dimension of \bar{P} remains the same, \mathbf{V} needs to account for the new snapshot.

The dimension of \bar{P} increases when adding the new snapshot drives the eigenvalue corresponding to λ_{N_p+1} into the dominant subspace. We assume that only one eigenvalue can become dominant during each step. Define the matrix $c_q = \mathbf{Q}\mathbf{L}\mathbf{Q}$. The power iteration

$$q^{(v+1)} = (c_q)^v q^{(0)} \quad (8)$$

produces iterates that asymptotically lie in the dominant eigenspace of c_q .

To decide if the dimension of \bar{P} should be decreased, we define the $N_p \times N_p$ matrix $\mathbf{H} = \mathbf{V}^T\mathbf{L}\mathbf{V}$. The eigenvalues of \mathbf{H} are a subset of the eigenvalues of the covariance matrix \mathbf{L} . Note here that while we need to compute the eigenvalues of \mathbf{H} , $N_p \ll N$ so this computation shouldn't be very expensive. If \hat{N} eigenvalues of \mathbf{H} are dominant, where $\hat{N} < N_p$, then $\text{span}(\mathbf{V}\mathbf{W})$ is a good approximation to the dominant eigenspace of \mathbf{L} . Here \mathbf{W} is the basis for the dominant eigenspace of \mathbf{H} . To reduce the basis we use Gram-Schmidt orthonormalization of $\mathbf{V}\mathbf{W}$. After each step, regardless of how the subspaces change, in order to maintain the accuracy of the basis, we use Gram-Schmidt orthonormalization on $\mathbf{L}\mathbf{V}$.

5. Numerical Results

Once the POD modes and the matrices and vectors in (3) have been generated, we use the trapezoidal method with 1000 time steps to find the approximate solution to the system using both the full and reduced-order approximation. In Figure 2, we illustrate the solutions for both methods for $0 \leq t \leq 2.5$ at $x = \frac{3\cdot\ell}{5}$. Here, the full-order system has 17 modes and the POD system has 1 mode. To pick N_p , we require $\frac{\sum_{i=1}^{N_p} \lambda_i}{\sum_{i=1}^{N_s} \lambda_i} \geq 1 - 10^{-8}$. For the POD solution we use 6 snapshots distributed equally in the time interval. The norm of the difference between the two solutions is $\|\cdot\|_2 = 7.5 \times 10^{-6}$. The full-order solution takes .208 seconds to compute while the POD solution takes .030 seconds. Here the forcing function f , is a given by the curve $f(t) = \sin(10\pi t)$.

Now let the forcing function

$$f(t) = \begin{cases} 0 & t \in [0, .1] \\ \frac{10}{3}(t - 0.1) & t \in (0.1, 0.4] \\ 1 & t \in (0.4, 0.5] \\ \sin(10\pi t) + 1 & t \in (0.5, 2] \\ 1 & t \in (2, 2.1] \\ \frac{10}{3}(2.4 - t) & t \in (2.1, 2.4] \\ 0 & t \in (2.4, 2.5] \end{cases} \quad (9)$$

be the function shown in Figure 3. Because f is piecewise defined, we need to consider the times where the snapshots are collected. It is possible to miss one segment of the forcing function completely with the snapshots. If we don't have a snapshot in the time segment where the forcing function behaves a certain way, it is possible that the POD modes won't show the change in behavior that the beam exhibits from the change in the force.

As we did for the example with the sinusoidal forcing function, we use 6 snapshots distributed equally throughout the time interval and let $\frac{\sum_{i=1}^{N_p} \lambda_i}{\sum_{i=1}^{N_s} \lambda_i} \geq 1 - 10^{-8}$ which yields $N_p = 1$. Comparing the full-order and POD solution for the system with the piecewise forcing function yields a norm of $\|\cdot\|_2 = 9.7 \times 10^{-6}$. The times where we collect the snapshots relative to the behavior of the forcing function is shown in Figure 3. Now we will keep everything the same, but we add 2 updates during the process. Updating at the times $t = 0.405$ and $t = 2.105$, as shown in Figure 3, reduces the differences between the full and reduced-order solutions to $\|\cdot\|_2 = 9.5 \times 10^{-6}$. Here N_p does not change after either of the updates. The full and reduced-order solutions are both shown in Figure 4.

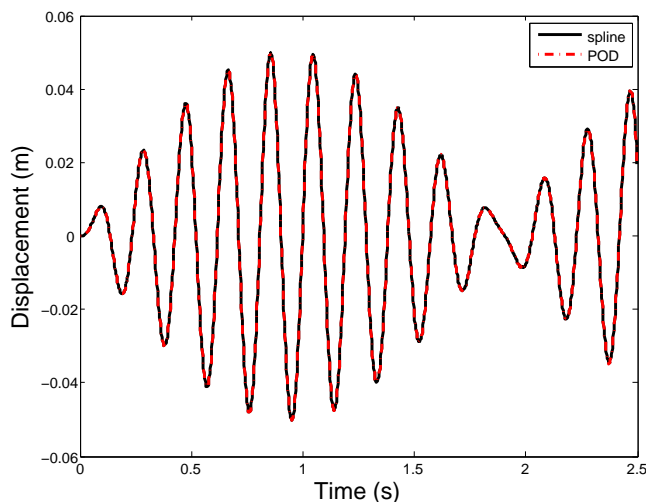


Figure 2: Comparison between the full and reduced-order solutions of (3).

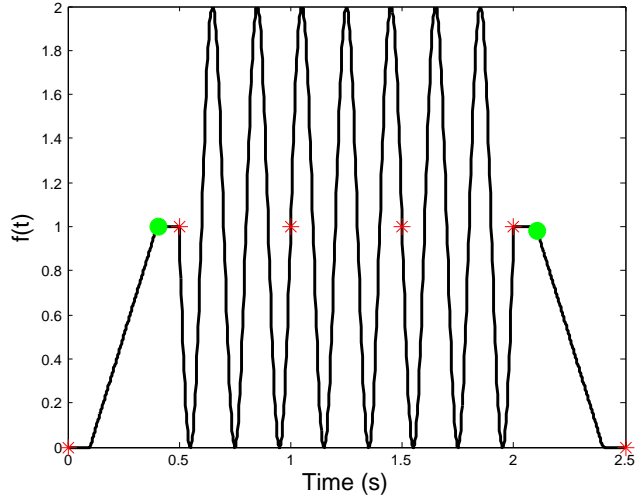


Figure 3: Piecewise forcing function $f(t)$ given by (9) (*-times where snapshots are recorded in full-order solution, o-update times).

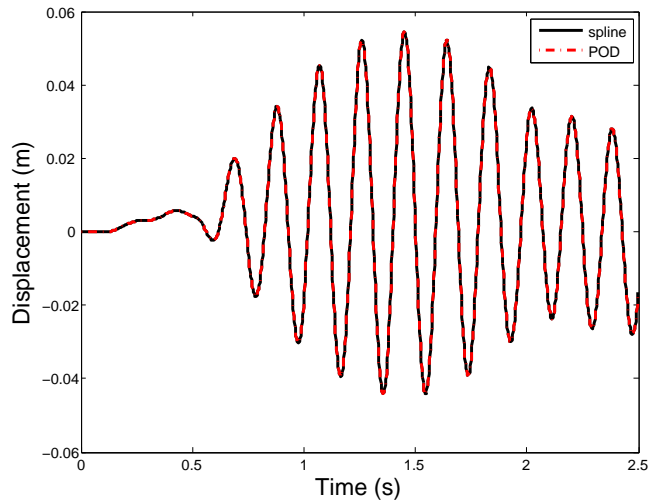


Figure 4: Full and reduced-order solutions of (3) computed using updates to POD.

6. Concluding Remarks

A method for adding updates to the POD basis for the beam model is discussed. We see that even for a simple example, updating with new snapshots can improve the accuracy of the simulation. Updating snapshots in a thoughtful way will be very important in future research in real time control of smart material systems.

Acknowledgements

This research was supported in part by the Air Force Office of Scientific Research through the grants AFOSR-FA9550-04-1-0203 and AFOSR-FA9550-08-1-0348.

References

- [1] Banks, H.T., Beeler, S. C., Kepler, G. M., and Tran, H. T., “Reduced order modeling and control of thin film growth in an HPCVD reactor,” *SIAM Journal on Applied Mathematics*, 62(4), pp. 1251-1280, 2002
- [2] Banks, H. T., Smith, R. C., Wang, Y., *Smart Material Structures: Modeling, Estimation, and Control*, Masson/John Wiley, Paris/Chichester, 1996
- [3] Banks, H.T., and Tran, H.T., “Reduced order based compensator control of thin film growth in a CVD reactor,” *International Series of Numerical Mathematics*, **139**. pp. 1-17, 2001
- [4] Chatterjee, A., “An introduction to the proper orthogonal decomposition,” *Current Science*, 78(7), pp. 808-817, 2000
- [5] del Rosario, R.C.H., Banks, H.T., and Tran, H.T., “Proper orthogonal decomposition-based control of transverse beam vibrations: Experimental implementation,” *IEEE Transactions on Control Systems Technology*, 10(5), pp. 717–726, 2002.
- [6] Hicks, G.P, and Lewis, B.M., “POD based control of beam vibrations: Methodology and experimental implementation,” *Proceedings of the 15th International Symposium on the Mathematical Theory of Networks and Systems*, South Bend, IN, 2002.
- [7] Pitchaiah, S. and A. Armaou, ”Adaptive Model Reduction for Output Feedback Control of dissipative PDE systems,” *Proceedings of the 18th International symposium on Mathematical Theory of Networks Systems*, SSDemi3.1, Blacksburg, Virginia, 2008.
- [8] Smith, R. C., *Smart Material Systems: Model Development*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005
- [9] Varshney, A. and A. Armaou, “Nonlinear control of dissipative PDE systems employing adaptive model reduction,” *Proceedings of the American Control Conference*, 940-947, Seattle, Washington, 2008.