01 Jan 2002

Proper Orthogonal Decomposition Based Feedback Optimal Control Synthesis of Distributed Parameter Systems Using Neural Networks

Radhakant Padhi

S. N. Balakrishnan
Missouri University of Science and Technology, bala@mst.edu

Proceedings of the American Control Conference
Anchorage, AK May 8-10, 2002

Proper Orthogonal Decomposition Based Feedback Optimal Control Synthesis of Distributed Parameter Systems Using Neural Networks

Radhakant Padhi S. N. Balakrishnan
Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri - Rolla, MO 65409, USA
padhi@umr.edu bala@umr.edu

Abstract

A new method for optimal control design of distributed parameter systems is presented in this paper. The concept of proper orthogonal decomposition is used for the model reduction of distributed parameter systems to form a reduced order lumped parameter problem. The optimal control problem is then solved in the time domain, in a state feedback sense, following the philosophy of 'adaptive critic' neural networks. The control solution is then mapped back to the spatial domain using the same basis functions. Numerical simulation results are presented for a linear and a nonlinear one-dimensional heat equation problem in an infinite time regulator framework.

1. Introduction

Distributed Parameter Systems (DPS) are governed by a set of partial differential equations. There exist theoretical methods for the control of distributed parameter systems [Curtain] in the infinite dimensional operator theory framework. While there are many advantages, these approaches are mainly confined to the linear systems besides having the usual difficulties in control implementation through an infinite dimensional operator. An engineering approach to deal with the infinite dimensional systems is to have a finite dimensional approximation of the system using a set of orthogonal basis functions before control design. Attention is being increasingly focused in the recent literature on the technique of *proper orthogonal decomposition* [Burns, Ravindran].

Many difficult real-life optimal control problems can be formulated in the framework of dynamic programming [Bryson], which handles this problem by producing a family of optimal paths, or what is known as the "*field of extremals*". One great drawback of this approach, however, is that it requires a prohibitive amount of computation and storage in producing this entire field of extremals. Towards designing a computational tool for finding a feedback form of the optimal control solution for nonlinear lumped parameter systems, an *approximate dynamic programming* approach, followed by the *adaptive critic* neuro control synthesis has been proposed in the literature [Balakrishnan, Werbos]. This makes it possible to synthesize the feedback optimal controllers for complex system. It allows the philosophy of dynamic programming to be carried out without the need for impossible computation and storage requirements.

In this paper an attempt has been made to combine the ideas of proper orthogonal decomposition and adaptive critic based optimal control synthesis to come up with a powerful computational tool for the optimal control of DPS. We have presented numerical simulation results for one-dimensional linear and nonlinear heat equation problems, with an infinite time optimal control formulation. We have compared the numerical results with the closed form solution for the linear problem, which shows good agreement.

2. Proper Orthogonal Decomposition: A Review

In this section we briefly summarize the process of proper orthogonal decomposition. For further readings, an interested reader can refer to [Burns, Ravindran].

Let $\{U_i(y):\ 1 \le i \le N,\ y \in \Omega\}$ be a set of N *snapshot* solutions (observations) of some physical process over the domain Ω at arbitrary instants of time. The goal of the POD technique is to design a basis function Φ that has the largest mean square projection on the snapshots. As a standard notation the L^2 inner product is defined as $\langle \Phi, \Psi \rangle = \int_\Omega \Phi \Psi \, dy$. We seek $\Phi = \sum_{i=1}^{N} w_i U_i$ where the coefficients w_i are to be determined such that Φ maximizes $(1/N)\left(\sum_{i=1}^{N} \left|\langle U_i, \Phi \rangle\right|^2 / \langle \Phi, \Phi \rangle \right)$. After some algebra it can be shown [Ravindran] that this leads to:

$$
\begin{aligned}
&CW = \sigma W \\
&C = \left[c_{ij}\right],\ c_{ij} = \frac{1}{N}\int_\Omega U_i(y)\, U_j(y)\, dy
\end{aligned}
\tag{1}
$$

In Eq.(1) $\sigma \in \mathbb{R}$ and $W = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T$. So, we have a standard matrix eigenvalue and eigenvector problem to find W. Matrix C has N *non-negative real* eigenvalues and N orthogonal eigenvectors. Sorting the eigenvectors in descending order, we can write $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_N \ge 0$. Let the corresponding eigenvectors be $W^1 = \begin{bmatrix} w_1^1 & \cdots & w_N^1 \end{bmatrix}^T$, $W^2 = \begin{bmatrix} w_1^2 & \cdots & w_N^2 \end{bmatrix}^T \ldots W^N = \begin{bmatrix} w_1^N & \cdots & w_N^N \end{bmatrix}^T$. It can be noted that the eigenspectrum can be truncated judiciously such that $\sum_{j=1}^{\tilde{N}} \lambda_j \approx \sum_{j=1}^{N} \lambda_j$. In that case we obtain $\tilde{N}$ orthonormal eigenfunctions as:

$$
\Phi_1 = \sum_{i=1}^{N} w_i^1\, U_i(y),\ \cdots,\ \Phi_{\tilde{N}} = \sum_{i=1}^{N} w_i^{\tilde{N}}\, U_i(y) \tag{2}
$$

The $\|\Phi\| = 1$ condition is met when we normalize W^js by forcing

$$
\langle W^j, W^j \rangle = 1/\left(N\lambda_j\right) \tag{3}
$$

3. Finite Dimensional Approximations

We consider the system described by:

$$\frac{\partial x}{\partial t} = f\left(x,\ \partial x/\partial y,\ \partial^2 x/\partial y^2,\ \ldots, u\right) \tag{4}$$

with appropriate *boundary conditions* that minimizes the performance index:

$$J = \int_0^{t_f} \int_0^{L} L(x,\ u)\, dy\, dt \tag{5}$$

Here the state x and control u are functions of time t and spatial variable y, $0 \le y \le L$. The process of getting the snapshot solutions for our example problem will be discussed later in Section 6.3. With the snapshot solutions we design the POD basis functions following the idea from Section 2. After obtaining the basis functions, we propose to write:

$$x = \sum_{j=1}^{\tilde{N}} \hat{x}_j(t)\,.\,\Phi_j(y),\ \ u = \sum_{j=1}^{\tilde{N}} \hat{u}_j(t)\,.\,\Phi_j(y) \tag{6}$$

We have assumed the same basis functions for x and u. In other words, we assume that the basis functions for the state are capable of representing the control as well. This is because our final aim is to design a *state feedback controller*. Substituting Eq.(6) in Eq.(4) and taking the inner product of this equation on a specific basis function Φ_i we can write:

$$\dot{\hat{x}}_i = \hat{F}_i\left(\hat{x}_j, \hat{u}_j,\ j = 1, 2, \ldots, \tilde{N}\right) \tag{7}$$

One can notice that by the definition of inner product all functionality dependence on y is now absorbed in the integrals. Collecting all equations for $i = 1, 2, \cdots, \tilde{N}$ we can write a $\tilde{N}$ dimensional lumped model for the system as:

$$\dot{\hat{X}} = \hat{F}\left(\hat{X}, \hat{U}\right) \tag{8}$$

Similarly, we can substitute for x and u from Eq.(8) in the expression for the performance index in Eq.(7) to obtain:

$$J = \int_0^{t_f} \hat{L}\left(\hat{X},\ \hat{U}\right)\, dt \tag{9}$$

Eq.(8) and (9) formulate an analogous optimal control problem in the time domain. We point out that the boundary conditions of the PDE are absorbed in Eq.(8).

4. Optimality Conditions

The necessary conditions of optimality for a lumped problem to minimize the performance index in Eq.(9) for a system driven by the dynamics in Eq.(8) is well known [Bryson]. First a *Hamiltonian* variable H is defined as:

$$H = \hat{L} + \lambda^T \hat{F} \tag{10}$$

where λ is the Lagrange multiplier variable. Then the so-called optimal control and costate equations are given by:

$$\partial H / \partial \hat{U} = 0 \tag{11}$$

$$\dot{\lambda} = -\ \partial H / \partial \hat{X} \tag{12}$$

For the solution of optimal control, Eq.(8), (11) and (12) need to be solved simultaneously along with appropriate boundary conditions. The adaptive critic methodology using a set of neural networks needs discrete version of the optimality conditions. In this connection, it is well known that the state equation Eq.(8) develops forward, while the costate equation Eq.(12) develops backward in time. For this reason, at any instant of time k we can write:

$$\hat{X}_{k+1} = \hat{F}_d\left(\hat{X}_k, \hat{U}_k\right) \tag{13}$$

$$\lambda_k = \hat{G}_d\left(\hat{X}_k, \hat{U}_k, \lambda_{k+1}\right) \tag{14}$$

where $\hat{F}_d$ and $\hat{G}_d$ are the resulting algebraic functions of their arguments. In this paper we consider the larger class of problems for which at the time instant k the optimal control $\hat{U}_k$ can explicitly solvable in terms of $\hat{X}_k$ and λ_{k+1} as:

$$\hat{U}_k = \xi\ \left(\hat{X}_k, \lambda_{k+1}\right) \tag{15}$$

We have used the standard fourth order Runge-Kutta method to get Eq.(13) and (14) from Eq (8) and (12) respectively.

5. Simplified Adaptive Critic Synthesis

We propose a set of neural networks, which solve the optimal control problem contained in Eq. (17), (18) & (19), together with appropriate boundary conditions. This control synthesis is essentially obtained through a set of *critic networks*. This is to retain the terminology of the *adaptive critic*, outlined earlier in [Balakrishnan, Werbos]. However, we have eliminated the need of the so-called *action networks* and hence, the need of iterative training between action and critic networks as well. This saves a lot of computations, besides eliminating the functional approximation error for having additional neural networks.

5.1 State generation for neural network training

Once the snapshot solutions are generated and POD basis functions are designed, we observe that

$$\hat{x}_{jk} = \left\langle x_k(y), \Phi_j(y)\right\rangle \tag{16}$$

Using the snapshot solutions in Eq.(16) we fix the minimum and maximum values for the individual elements of $\hat{X}_k$. Let $\hat{X}_{\min}$ denote the vector for minimum values for $\hat{X}_k$ and $\hat{X}_{\max}$ denote the vector of maximum values. Then fixing a positive constant $0 \le C_i \le 1$, we select $\hat{X}_k \in C_i.\left[\hat{X}_{\min}, \hat{X}_{\max}\right]$. Let $S_i = \left\{\hat{X}_k : \hat{X}_k \in C_i.\left[\hat{X}_{\min}, \hat{X}_{\max}\right]\right\}$. One can notice that for $C_1 \le C_2 \le \ldots$, $S_1 \subseteq S_2 \subseteq \ldots$ Thus, for some $i = I$, $C_I = 1$

and S_I will include the domain of interest for initial conditions. In this paper, we have chosen $C_1 = 0.05$, $C_i = C_1 + 0.05\,(i-1)$ for $i = 2, 3, \ldots$ till $i = I$

5.2 Neural network training

Fix C_i and generate S_i. For each element $\hat{X}_k$ of S_i follow the steps below [Figure 1].

Input $\hat{X}_k$ to the networks to get λ_{k+1}. Let us denote it as λ^{a}_{k+1}. Calculate $\hat{U}_k$, knowing $\hat{X}_k$ and λ_{k+1}, from *optimal control equation* Eq.(15). Get $\hat{X}_{k+1}$ from the *state equation* (13), using $\hat{X}_k$ and $\hat{U}_k$. Input $\hat{X}_{k+1}$ to the networks to get λ_{k+2}. Calculate λ_{k+1}, form the *costate equation* (14). Let us denote this as λ^{t}_{k+1}. Train the networks, with all $\hat{X}_k$ as input and all corresponding λ^{t}_{k+1} as output. If proper convergence is achieved, stop and revert to step 1, with S_{i+1}. If not, go to step 1 and retrain the networks with a new S_i.

One can notice that for faster convergence, one can take the convex combination $\left[\beta\ \lambda^{t}_{k+1} + (1-\beta)\ \lambda^{a}_{k,j+1}\right]$ as the target output for training, where $0 < \beta < 1$ is the learning rate for the neural network training. Moreover, to minimize the chance of getting trapped in a local minimum, we have followed the *batch training* philosophy. For the heat conduction examples presented in this paper, we have chosen $\beta = 0.5$.

5.3 Convergence check

First, fix a tolerance value (we have fixed $tol = 0.1$, for the heat conduction problem). By using the profiles from S_i^c, generate the target outputs, as described in Section 5.2. Say the outputs are $\lambda^{t_i}_{1}, \lambda^{t_i}_{2}, \ldots, \lambda^{t_i}_{\tilde{N}}$. Generate the actual output from the networks, by simulating the *trained* networks with the profiles from S_k^c. Say the values of the outputs are $\lambda^{a_i}_{1}, \lambda^{a_i}_{2}, \ldots, \lambda^{a_i}_{\tilde{N}}$. Check whether simultaneously $\left\|\lambda^{t_i}_{j} - \lambda^{a_i}_{j}\right\|_2 / \left\|\lambda^{t_i}_{j}\right\|_2 < tol,\ \forall\ j = 1, 2, \ldots, \tilde{N}$. If yes, we assume that the networks have converged.

6. Motivating Examples: One-Dimensional Heat Conduction Equations

6.1 Problem Description

We consider a nonlinear one-dimensional heat conduction problem given by:

$$\partial x / \partial t = \partial^2 x / \partial y^2 - x^3 + u \qquad (17)$$

The linear version of the problem is given by:

$$\partial x / \partial t = \partial^2 x / \partial y^2 + u \qquad (18)$$

We consider the infinite time quadratic regulator problem, for which the performance index to be minimized is given by:

$$J = \frac{1}{2} \int_0^{t_f \to \infty} \int_0^{L} \left(q x^2 + r u^2\right) dy\, dt \qquad (19)$$

Boundary and Initial Conditions

We assume that the boundary conditions are given by $\frac{\partial x}{\partial y}(t, 0) = \frac{\partial x}{\partial y}(t, L) = 0$ and the initial condition can be *any* profile from the *domain of interest*.

6.2 Domain of interest and state profile generation

We assume an envelope profile

$$f_{env}(y) = a + A\,Cos\left(-\pi + (2\pi y / L)\right) \qquad (20)$$

We define $S_I \equiv \left\{ \begin{array}{c} x : \| x \| \le \| f_{env} \|,\ \| x'' \| \le \| f''_{env} \| \\ and\ \ x'(t,0) = x'(t,L) = 0 \end{array} \right\}$ as the domain of interest. The conditions put on x ensure that the profiles are smooth and they satisfy the boundary conditions. For our numerical experiments, we have chosen $a = A = 0.25$. For the envelope profile chosen we have

$$\| f_{env} \|^2 = \left(a^2 + A^2/2\right) L, \quad \| f''_{env} \|^2 = A^2 \pi^4 (2/L)^3 \qquad (21)$$

After fixing $0 \le C_i \le 1$, we assume

$$\|x\|^2_{\max} = C_i \|f_{env}\|^2,\ \|x''\|^2_{\max} = \|f''_{env}\|^2 \qquad (22)$$

We assume a Fourier cosine series expansion for $x(y)$:

$$x = a_0 + \sum_{n=1}^{N_f} a_n Cos(n\pi y / L) \qquad (23)$$

where N_f is a large number. After some algebra, we observe:

$$\begin{aligned} \frac{L}{2}\left(2a_0^2 + \sum_{n=1}^{N_f} a_n^2\right) &\le C_i\left(a^2 + A^2/2\right)L \\ \frac{L}{2}\left(\sum_{n=1}^{N_f} n^4 a_n^2\right)\left(\frac{\pi}{L}\right)^4 &\le A^2\pi^4 (2/L)^3 \end{aligned} \qquad (24)$$

So we select *random* coefficients $a_n,\ n = 0, 1, \ldots, N_f$ to satisfy both the inequalities of Eq.(24) and generate a state profile using Eq. (23).

6.3 Snapshot solution generation

We have followed the steps below to generate the snapshot solutions.

Fix $0 \le C_i \le 1$ and generate a random initial state profile $x(0, y)$. Generate a random control profile as well, similar to the state profile generation. This is done under the assumption that the controller will satisfy $\|u\| \le \|x(0, y)\|$ and $\|u''\| \le \|x''(0, y)\|$. Holding the control as constant, simulate the original system Eq.(17). Randomly select some profiles at arbitrary instants of time and assume that those are the snapshot solutions. We propose to repeat the steps outlined above a number of times and to collect some snapshot solutions each time, till enough number of snapshots is collected.

6.4 Finite dimension approximations

First the snapshot solutions are generated and POD basis functions are designed. Substituting Eq.(6) in Eq.(17), taking the inner product with Φ_i we get:

$$\dot{\hat{x}}_i = \sum_{j=1}^{\bar{N}} \langle \Phi_j'', \Phi_i \rangle \hat{x}_j - \int_0^L \left(\sum_{j=1}^{\bar{N}} \hat{x}_j \, \Phi_j \right)^3 \Phi_i \, dy + \hat{u}_i \tag{25}$$

Using the boundary conditions after some algebra, it leads to:

$$\dot{\hat{X}} = A\hat{X} + f^{nl}\left(\hat{X}\right) + B\hat{U} \tag{26}$$

where,

$$A \equiv \left[a_{ij}\right], \; a_{ij} = -\left\langle \Phi_i', \Phi_j' \right\rangle, \quad B \equiv I$$
$$f_i^{nl}\left(\hat{X}\right) \equiv -\int_0^L \left(\sum_{j=1}^{\bar{N}} \hat{x}_j \, \Phi_j \right)^3 \Phi_i \, dy \tag{27}$$

$f^{nl}\left(\hat{X}\right)$ is a nonlinear function that comes from the nonlinear term in Eq.(17). For the linear problem in Eq.(18), this term will be absent and we will have the following system dynamics.

$$\dot{\hat{X}} = A\hat{X} + B\hat{U} \tag{28}$$

For the performance index, we observe:

$$q\langle x, x\rangle = \hat{X}^T Q \hat{X}, \; r\langle u, u\rangle = \hat{U}^T R \hat{U}$$
$$\text{where } Q = q\,I, R = r\,I \tag{29}$$

Using Eq.(29), the performance index in Eq.(19) can be written as:

$$J = \frac{1}{2} \int_0^{t_f \to \infty} \left(\hat{X}^T Q \hat{X} + \hat{U}^T R \hat{U} \right) dt \tag{30}$$

6.5 Optimality conditions

Using Eq.(30), (10), (11) and (12), the optimal control and costate equations can be derived as

$$\hat{U} = -R^{-1} B^T \lambda \tag{31}$$

$$\dot{\lambda} = -Q\hat{X} - \left(A^T + \partial f^{nl} / \partial \hat{X}\right)\lambda \tag{32}$$

For the linear problem, the optimal control equation remains same as Eq. (31). However, the costate equation is given by:

$$\dot{\lambda} = -Q\hat{X} - A^T \lambda \tag{33}$$

Then, as pointed out in Section 4, Runge-Kutta method was used to derive the discrete versions of the optimality conditions.

6.6 Closed form solution for linear problem

For the linear problem, the closed form solution for the optimal control can be derived [Curtain]. The solution is given by:

$$u(t, y) = -\frac{1}{L}\int_0^L x(t, \bar{y})\, d\bar{y} - \frac{2}{L}\sum_{n=1}^{\infty} \left\{ -\left(\frac{n\pi}{L}\right)^2 + \sqrt{-\left(\frac{n\pi}{L}\right)^4 + 1} \right\} \left\{ \int_0^L x(t, \bar{y})\, Cos\left(\frac{n\pi \bar{y}}{L}\right) d\bar{y} \right\} Cos\left(\frac{n\pi y}{L}\right) \tag{34}$$

6.7 Choice of neural network structure

We have taken five $\pi_{5,5,5,1}$ neural networks, one each for each of the costates. A $\pi_{5,5,5,1}$ neural network means 5 neurons in the input layer, 5 neurons in the first hidden layer, 5 neurons in the second hidden layer and 1 neuron in the output layer. For activation functions, we have taken a *tangent sigmoid* function for all the hidden layers and a *linear* function for the output layer.

6.8 Numerical results

For the numerical experimentation we chose $q = r = 1$, $L = 4$. For implementing the control we assumed a control update scheme with $\Delta t = 0.1$. In the finite difference scheme for generating the snapshot solutions we assumed $\Delta t = 0.002$, $\Delta y = 0.1$. However for simulating the system after control synthesis, we have assumed $\Delta t = 0.001$, $\Delta y = 0.05$. The choice of this two different set of values was to emphasize the point that the control synthesis methodology presented is independent of the grid size. This was also to see that the results are not bad because of the spillover effects, by assuming a particular grid size for generating the snapshot solutions and hence the basis functions. However to compute the values of the basis functions at a location other than where it was constructed, we have opted for an interpolation scheme based on the Fourier cosine series having the same number of terms as the number of points for which the function values exist.

The first objective was to show that the approach is a viable tool for the optimal control synthesis of the distributed parameter systems. We notice that the problems we considered for numerical experimentation represent *infinite time regulator* problems. So both the state and control over the entire spatial domain should proceed towards zero as time progresses. In Figures 2 and 3, we present a typical result for the *nonlinear problem*, after synthesizing the POD based optimal neuro control. As expected, state and control histories from a random initial condition develop towards zero with the increase of time. We present the simulation results for the *linear problem* in Figures 4 and 5 from another random initial condition. These figures again show the same expected trend that the state and control develop towards zero as time increases.

To boost the confidence on the methodology presented, we simulated the system with this closed form control starting from the same initial state profiles. The state and control results are presented in Figures 6 and 7 respectively. Comparing Figures 4 and 6 as well as 5, 7 the closeness of the results is obvious. However, we have exclusively plotted the errors between the states and controls coming from the two approaches in Figures 8 and 9. From these plots it can be clearly seen that the magnitude of the *error in states* is two-order less, as compared to the magnitude of states. Similarly, the magnitude of the *error in control* is one-order less, as compared to the magnitude of control. This good agreement of results successfully verifies the proposed technique of optimal control synthesis for DPS.

Even though we have presented the results only from a single initial condition (because of space limitations), the same behavior was observed from a large number of arbitrarily chosen random initial conditions in the domain of interest. This shows that the control synthesis methodology presented can be implemented in a feedback sense. We also point out that computation of the control only uses the neural networks already synthesized off-line. Since using the networks are not at all computationally intensive, this methodology can be implemented on-line.

7. Conclusions

In this paper a systematic computational tool for the optimal control synthesis of distributed parameter systems is presented. Using the concept of POD a low-dimensional lumped model representation of the infinite dimensional system was developed. This low dimensional model was used to synthesize the optimal control, in a state feedback sense, following the philosophy of adaptive critic neural networks. The synthesized control in time domain was then extended to the spatial domain using the same basis functions. We have synthesized the optimal control for a one-dimensional nonlinear and a linear heat conduction problem. Simulations show good results for all cases. The results for this case were successfully compared with the closed form solution, for some typical initial conditions, showing close matching.

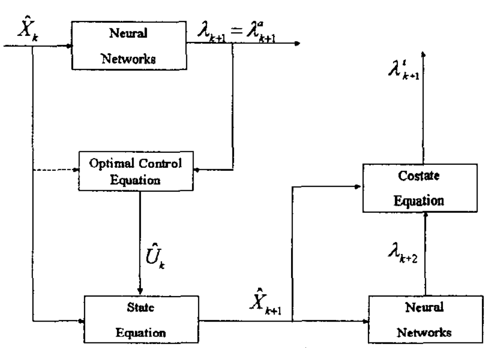

Figure 1: Schematic of simplified adaptive critic neural network synthesis

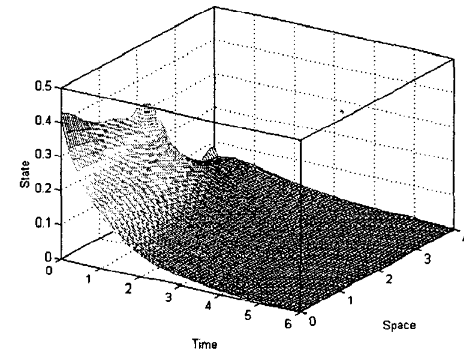

Figure 2: State of the nonlinear system from a random initial condition

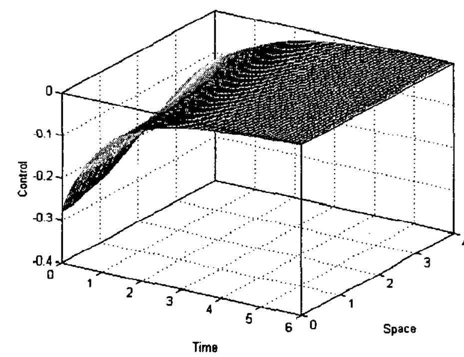

Figure 3: Associated control of the nonlinear system from the random initial condition

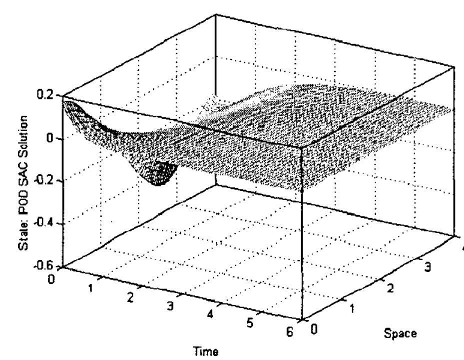

Figure 4: State of the linear system from a random initial condition with POD control

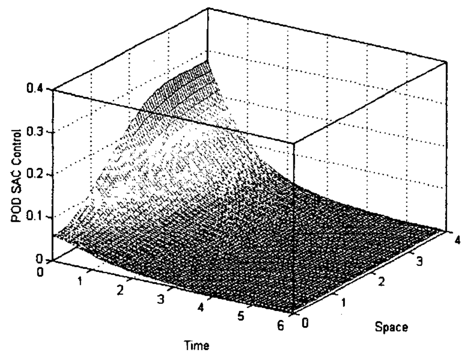

Figure 5: Associated POD control of the linear system for the random initial condition

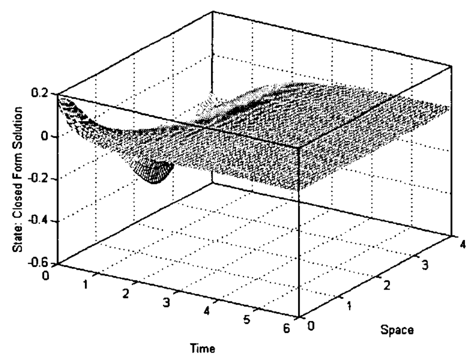

Figure 6: State of the linear system from a random initial condition with closed form control

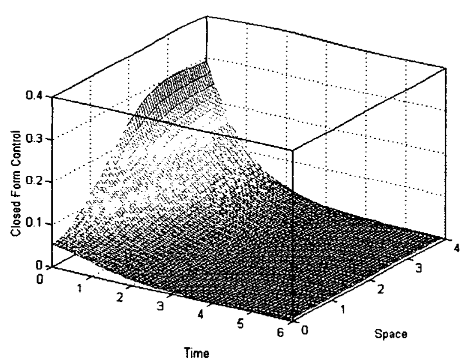

Figure 7: Associated closed form control of the linear system for the random initial condition

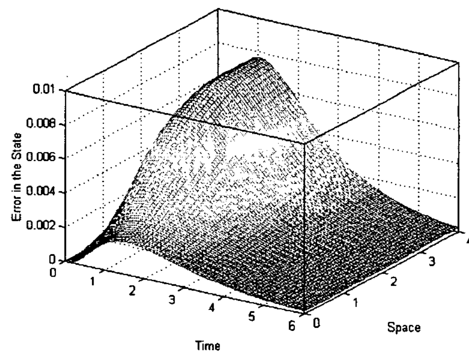

Figure 8: Error in state of the linear system for the random initial condition

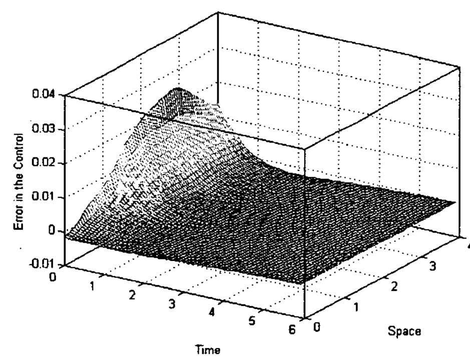

Figure 9: Error in control of the linear system for the random initial condition

Acknowledgement

This research was supported by NSF-USA grant ECS 9976588.

References

1. Balakrishnan S. N. and Biega V., Adaptive-Critic Based Neural Networks for Aircraft Optimal Control, *Journal of Guidance, Control and Dynamics, Vol. 19, No. 4, July-Aug. 1996, pp. 893-898.*

2. Bryson A. E. and Ho Y. C., Applied Optimal Control, *London: Taylor and Francis, 1975.*

3. Burns J. and King B. B., A Reduced Basis Approach to the Design of Low-order Feedback Controllers for Nonlinear Continuous Systems, *Journal of Vibration and Control, Vol. 4, No. 3, 1998, pp. 297-323.*

4. Curtain R. F. and Zwart H. J., An Introduction to Infinite Dimensional Linear Systems Theory, *Springer-Verlag, New York, 1995.*

5. Ravindran S. S., Proper Orthogonal Decomposition in Optimal Control of Fluids, *NASA/TM-1999-209113.*

6. Werbos P. J., Neuro control and Supervised Learning: An Overview and Evaluation, *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, White D. and Sofge D. (Eds.), Van Nostrand Reinhold, 1992.*