15th Triennial World Congress, Barcelona, Spain

PROPER ORTHOGONAL DECOMPOSITION BASED OPTIMAL CONTROL DESIGN OF HEAT EQUATION WITH DISCRETE ACTUATORS USING NEURAL NETWORKS

Radhakant Padhi[1] **S. N. Balakrishnan**[2]
Dept. of Mechanical and Aerospace Engineering and Eng. Mechanics
University of Missouri-Rolla, USA

Abstract: A new method is presented for the optimal control design of one-dimensional heat equation with the actuators being concentrated at discrete points in the spatial domain. This systematic methodology incorporates the advanced concept of proper orthogonal decomposition for the model reduction of distributed parameter systems. After deigning a set of problem oriented basis functions an analogous optimal control problem in the lumped domain is formulated. 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. *Copyright © 2002 IFAC*

Keywords: distributed parameter systems, optimal control, feedback control, neural networks, nonlinear systems

1. INTRODUCTION

Distributed Parameter Systems (DPS) are governed by a set of partial differential equations. One approach to deal with the distributed parameter systems is to have a finite dimensional approximation of the system using a set of orthogonal basis functions via Galerkin projection (Holmes, 1996). The methodology of Galerkin projection normally leads to high order lumped system representations to adequately represent the properties of the original system, if arbitrary orthogonal functions are used as the basis functions. For this reason attention is being increasingly focused in the recent literature on the technique of *Proper Orthogonal Decomposition (POD)* (Banks *et. al.,* 2000; Holmes *et. al.,* 1996; Ravindran, 1999; Singh *et. al. 2001*).

The synthesis of various nonlinear control laws using neural networks has been demonstrated in a variety of applications (Hunt, 1992; White, 1992). Towards designing a computational tool for finding a feedback form of the optimal control solution for nonlinear lumped parameter systems, the *Adaptive Critic* neuro control methodology has been proposed in the literature (Balakrishnan & Biega, 1996; Werbos, 1992). This methodology comes up with a state feedback control law by the off-line training of the so-called 'action' and 'critic' networks, for an entire envelope of states. 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.

This paper is an attempt to combine the ideas of POD and adaptive critic synthesis to come up with a powerful computational tool for the optimal control of one-dimensional heat equation. First the problem oriented basis functions are designed from a set of snapshot solutions, following the idea of POD. Then an analogous finite dimensional optimal control problem is formulated in the time domain. After synthesizing the control in the time domain we generate the control function in the spatial domain by using the basis functions. We have presented numerical simulation results for one-dimensional linear and nonlinear heat equation problems, with an infinite time optimal control formulation. The control synthesis was carried out assuming *point actuators* in the spatial domain.

The neuro optimal control methodology described in this paper retains all the powerful features of the adaptive critic methodology. However, we have been successful in completely eliminating the action networks. As an added benefit, we no more require the iterative training loops between the action and critic networks. So the methodology presented in this paper leads to a considerable saving of computations

[1] Lecturer and Post Doctoral Fellow (padhi@umr.edu)
[2] Professor (Contact Person, bala@umr.edu)

besides eliminating the error associated with the additional neural network approximations. For this reason, this paper can also be thought of as an improvement of the adaptive critic technique.

To the knowledge of the authors, this is the first neural network paper to present a systematic computational tool for the feedback optimal control synthesis of distributed parameter systems that incorporates the powerful technique of proper orthogonal decomposition. It is also the first paper to present a viable computational tool to address discrete (point) controllers in the spatial domain.

2. ONE-DIMENSIONAL HEAT CONDUCTION PROBLEM WITH POINT ACTUATORS

2.1 Problem Description

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

$$\frac{\partial x(t,y)}{\partial t} = \frac{\partial^2 x(t,y)}{\partial y^2} - x^3(t,y) + u(t,y) \qquad (1)$$

The linear version of the problem is without the $x^3(t,y)$ term. We assume that the control is excited by a set of actuators concentrated at discrete points. So, $u(t,y)$ is given by:

$$u(t,y) = \sum_{m=1}^{M_u} v(t,y).\delta(y - y_m) \qquad (2)$$

where $v(t,y)$ is a continuous function both in time t and space y. We define the $\delta(.)$ function as follows.

$$v(t,y).\delta(y-y_m) = \begin{Bmatrix} v(t,y_m), & y = y_m \\ 0, & y \neq y_m \end{Bmatrix} \qquad (3)$$
$$\int_0^L v(t,y).\delta(y-y_m)\,dy = v(t,y_m)$$

We consider the cost function to be minimized is given by:

$$J = \frac{1}{2}\int_0^{t_f \to \infty}\int_0^L \left(qx^2 + ru^2\right) dy\,dt \qquad (4)$$

We assume that the boundary conditions are given by the boundary condition $x'(t,0) = x'(t,L) = 0$; *i.e.* both the ends are insulated. For initial profiles, we assume that the profiles can be *any* profile from the *domain of interest*.

2.2 Domain of Interest and State Profile Generation

We assume an envelope profile

$$f_{env}(y) = a + A\,Cos\left(-\pi + (2\pi y / L)\right) \qquad (5)$$

Then we define

$$S_I \equiv \begin{Bmatrix} x(t,y) : \| x(t,y) \| \le \| f_{env}(y) \|, \\ \| x''(t,y) \| \le \| f''_{env}(y) \| \\ and \;\; x'(t,0) = x'(t,0) = 0 \end{Bmatrix} \qquad (6)$$

as the domain of interest. After fixing $0 \le C_i \le 1$, we assume

$$\|x\|_{\max}^2 = C_i \|f_{env}\|^2, \quad \|x''\|_{\max}^2 = \|f''_{env}\|^2 \qquad (7)$$

To satisfy the boundary conditions, we assume a Fourier cosine series expansion approximation

$$x(t,y) = a_0 + \sum_{n=1}^{N_f} a_n Cos\left(n\pi y / L\right) \qquad (8)$$

After some algebra, we can write:

$$\begin{aligned} &\frac{L}{2}\left(2a_0^2 + \sum_{n=1}^{N_f} a_n^2\right) \le C_i\left(a^2 + A^2/2\right)L \\ &\frac{L}{2}\left(\sum_{n=1}^{N_f} n^4 a_n^2\right)\left(\frac{\pi}{L}\right)^4 \le A^2\pi^4 (2/L)^3 \end{aligned} \qquad (9)$$

To satisfy both the inequalities of (9), we select *random* coefficients a_n, $n = 0,1,\ldots,N_f$ and generate a state profile using Eq. (8). Such a profile is *guaranteed* to lie within our definition of the domain of interest (6).

2.3 Snapshot Solution Generation

To generate snapshot solutions we follow the procedure outlined below.

- Fix $0 \le C_i \le 1$ and generate a random state profile.
- Generate a random control profile as well, similar to the state profile generation and select the values at the control application points.
- Holding the control as constant, simulate the original system Eq.(1), possibly using a finite difference technique [Smith], for some finite time.
- 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 snapshots are collected.

2.4 Finite Dimension Approximations

With the snapshot solutions we design the problem oriented POD basis functions (Ravindran, 1999; Holmes, *et. al.* 1996]. Then we expand $x(t,y)$ and $v(t,y)$ as:

$$x(t,y) = \sum_{j=1}^{\tilde{N}} \hat{x}_j(t).\Phi_j(y), \quad v(t,y) = \sum_{j=1}^{\tilde{N}} \hat{v}_j(t).\Phi_j(y) \qquad (10)$$

Substituting Eq.(10) and Eq.(2) in Eq.(1), taking the inner product with Φ_i we get

$$\left\langle \sum_{j=1}^{\tilde{N}} \dot{\hat{x}}_j \Phi_j, \Phi_i \right\rangle = \left\langle \frac{\partial^2}{\partial y^2}\left(\sum_{j=1}^{\tilde{N}} \hat{x}_j \Phi_j, \Phi_i \right) \right\rangle - \left\langle \left(\sum_{j=1}^{\tilde{N}} \hat{x}_j \Phi_j \right)^3, \Phi_i \right\rangle + \left\langle \sum_{j=1}^{\tilde{N}} u(t,y), \Phi_i \right\rangle \quad (11)$$

Using the boundary conditions, we have $\Phi'_j(t,0) = \Phi'_j(t,L) = 0$, which leads to

$$\left\langle \Phi''_j, \Phi_i \right\rangle = -\left\langle \Phi'_j, \Phi'_i \right\rangle \quad (12)$$

We define a nonlinear function:

$$f_i^{nl}\left(\hat{X}\right) = -\int_0^L \left(\sum_{j=1}^{\tilde{N}} \hat{x}_j \Phi_j \right)^3 \Phi_i \, dy \quad (13)$$

We assume

$$v(t,y) = \sum_{j=1}^{\tilde{N}} \hat{v}_j(t)\, \Phi_j(y) \quad (14)$$

After some algebra, we can write:

$$\begin{aligned} &\left\langle u(t,y), \Phi_i(y) \right\rangle \\ &= \sum_{m=1}^{M_u} \sum_{j=1}^{\tilde{N}} \Phi_j(y_m)\Phi_i(y_m)\, \hat{v}_j(t) \\ &= \sum_{j=1}^{\tilde{N}} \left[\sum_{m=1}^{M_u} \Phi_j(y_m)\Phi_i(y_m) \right] \hat{v}_j(t) \end{aligned} \quad (15)$$

We define

$$\begin{aligned} A &= \left[a_{ij} \right]_{\tilde{N}\times\tilde{N}}, \quad a_{ij} = -\left\langle \Phi'_i, \Phi'_j \right\rangle \\ B &= \left[b_{ij} \right]_{\tilde{N}\times\tilde{N}}, \quad b_{ij} = \sum_{m=1}^{M_u} \Phi_j(y_m)\Phi_i(y_m) \end{aligned} \quad (16)$$

Using Eq. (11), (13) and (16) for $i = 1, 2, \cdots, \tilde{N}$, we obtain:

$$\begin{aligned} &\dot{\hat{X}} = A\hat{X} + f^{nl}\left(\hat{X}\right) + B\hat{U} \\ &\text{where } \hat{U} \equiv \left[\hat{v}_1 \quad \hat{v}_2 \quad \cdots \quad \hat{v}_{\tilde{N}} \right] \end{aligned} \quad (17)$$

where $f^{nl}\left(\hat{X}\right)$ is a nonlinear function that comes from the nonlinear term in Eq.(1). For the linear problem this term will be absent. For the cost function, we observe:

$$\begin{aligned} q\left\langle x, x \right\rangle &= q\left\langle \left(\sum_{i=1}^{\tilde{N}} \hat{x}_i \Phi_i \right), \left(\sum_{j=1}^{\tilde{N}} \hat{x}_j \Phi_j \right) \right\rangle \\ &= q\sum_{j=1}^{\tilde{N}} \hat{x}_j\, \hat{x}_j = \hat{X}^T Q\, \hat{X} \end{aligned} \quad (18)$$

where $Q \equiv diag\left(q_1 \quad q_2 \quad \cdots \quad q_{\tilde{N}} \right)$

Similarly,

$$\begin{aligned} &r\left\langle u, u \right\rangle = \hat{X}^T R\, \hat{X} \\ &\text{where } R = r\,B \end{aligned} \quad (19)$$

Thus the cost function in Eq.(4), can be written as:

$$J = \frac{1}{2} \int_0^{t_f \to \infty} \left(\hat{X}^T Q \hat{X} + \hat{U}^T R \hat{U} \right) dt \quad (20)$$

From Eq.(17) and (20) we have an optimal control formulation in the lumped parameter framework.

2.5 Optimality Conditions

Following the standard optimal control theory for lumped systems (Bryson, 1975). The optimal control equation can be derived as:

$$\hat{U} = -R^{-1} B^T \lambda \quad (21)$$

Similarly the costate equation can be derived as:

$$\dot{\lambda} = -Q\hat{X} - \left(A^T + \left[\frac{\partial f^{nl}}{\partial \hat{X}} \right] \right) \lambda \quad (22)$$

where λ is the Lagrange multiplier. For the linear problem the optimal control equation remains same as Eq. (21). However, the costate equation (22) does not contain $\partial f^{nl} / \partial \hat{X}$.

2.6 Choice of Neural Network Structure

For this particular problem 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.

3. NEURAL NETWORK SYNTHESIS

We propose a set of neural networks, which solve the optimal control problem contained in Eq. (17), (21) & (22), with appropriate boundary conditions.

3.1 State generation for neural network training

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

$$\hat{x}_{j_k} = \left\langle x_k(y), \Phi_j(y) \right\rangle \quad (23)$$

So we use all the snapshots in Eq.(23) and fix the minimum and maximum values for the individual elements of $\hat{X}_k$. Let $\hat{X}_{\max}$ denote the vector of maximum values for $\hat{X}_k$ and $\hat{X}_{\min}$ the vector for minimum 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 $C_1 \le C_2 \le C_3 \le \ldots$, $S_1 \subseteq S_2 \subseteq S_3 \subseteq \ldots$. Thus, for some $i = I$, $C_I = 1$ and S_I will include the domain of interest for initial conditions. Hence, to begin the synthesis procedure, we fix a small value for the constant C_1 and train the networks for the states, randomly generated within S_1. Once the critic networks converge for this set, we choose C_2 close to C_1 and again train the networks for the profiles within S_2 and so on. We keep on increasing the constant C_i this way till the set S_i includes domain of interest for the initial conditions. In this paper, we have chosen $C_1 = 0.05$, $C_i = C_1 + 0.05\ (i-1)$ for $i = 2, 3, \ldots$ and continued till $i = I$, where $C_I = 1$. However, any other scheme should also be fine.

3.2 Neural Network Training

For better capturing of the relationship between $\hat{X}_k$ and λ_{k+1}, we have synthesized separate networks for each element of the vector λ_{k+1}. We synthesize the neural networks in the following manner [Figure 1].

1. Fix C_i and generate S_i
2. For each element $\hat{X}_k$ of S_i follow the steps below
   - 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* (21)
   - Get $\hat{X}_{k+1}$ from the *state equation* (17), 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* (22). Denote this target output as λ^{t}_{k+1}
3. Train the networks, with all $\hat{X}_k$ as input and all corresponding λ^{t}_{k+1} as output
4. 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.

We have taken 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, where the network is trained for all of the elements of S_i together. For our heat conduction example problems, we have chosen $\beta = 0.5$.

One can notice, since $\hat{U}_k$ is supposed to be a known function of $\hat{X}_k$ and λ_{k+1}, after successful training of the networks, we can directly calculate the associated optimal control $\hat{U}_k$ from Eq.(21) and hence $v(t,y)$ from Eq.(14) and $u(t,y)$ from Eq.(2).

3.3 Convergence Condition

Before changing C_i to C_{i+1} and generating new profiles for further training, it should be assured that proper convergence is arrived for C_i. For this purpose C_i is fixed to the same values that have been used for the training of the networks. Generate a set S_i^c of profiles, exactly the same manner used to generate S_i. Moreover, fix a tolerance value (we have fixed $tol = 0.1$)

1. By using the profiles from S_i^c, generate the target outputs, as described in Section 3.2. Say the outputs are $\lambda^{t_i}_1, \lambda^{t_i}_2, \ldots, \lambda^{t_i}_{\tilde{N}}$
2. 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 $\lambda^{a_i}_1, \lambda^{a_i}_2, \ldots, \lambda^{a_i}_{\tilde{N}}$.
3. Check whether simultaneously $\left\|\lambda_j^{t_i} - \lambda_j^{a_i}\right\|_2 / \left\|\lambda_j^{t_i}\right\|_2 < tol,\ \forall\ j = 1, 2, \ldots, \tilde{N}$.

4. NUMERICAL RESULTS

For the numerical experimentation we have assumed all variables in a compatible unit system. We chose $q = r = 1$, $L = 4$. We assumed that the controllers are separated in the spatial dimension with $\Delta\, y_u = 1$. We assumed a control update scheme with $\Delta t = 0.1$. Accordingly in the fourth order Runge-Kutta method for the time integration of state and costate equations in the neural network synthesis process, we assumed $\Delta t = 0.1$. In our simulations of the systems, we have collected the state profile at every $\Delta t = 0.1$ for control calculations and held that control profile constant in an explicit finite difference simulations [Smith] till next update of the control. In the finite difference scheme for generating the snapshot solutions we assumed $\Delta t = 0.002, \quad \Delta y = 0.1$. However for simulating the system after control synthesis, we have assumed $\Delta t = 0.001, \quad \Delta y = 0.05$. It should be noted that the choice of values of Δt and Δy satisfies the standard CFL condition for numerical stability for linear parabolic systems $\Delta t / (\Delta y)^2 < 0.5$ [Smith] in both the cases.

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. Further since the aim was to present a synthesis tool in a state feedback sense in a domain of interest, this feature should be observed for all initial conditions in the domain of interest. However, since its impossible to present separate results for a large number of initial profiles (because of space limitations), we chose some representative profiles as initial conditions for the simulation purposes. One of the two such test profiles was generated with $x(0,y) = 0.2 + 0.2\,Cos(-\pi + 2\pi y / L)$. The other initial profile was generated at random.

In Figures 2-9 we present the state and associated control histories for the two representative initial conditions. We can see the expected trend of the state and control developing towards zero with the increase of time. The task of driving the state to zero in the entire spatial domain was achieved with no difficulty. Even though we have presented the results only from limited typical initial profiles, the same behaviour 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.

5. CONCLUSIONS

In this paper a systematic computational tool for the optimal control synthesis of a one-dimensional nonlinear and a linear heat conduction problem has been presented. The powerful proper orthogonal decomposition methodology has been used in designing problem-oriented basis functions, which were used in a Galerkin projection to come up with a low-dimensional lumped model representation of the infinite dimensional system. Using this low dimensional model we synthesized the optimal control, in a state feedback sense in the domain of interest, 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. This was done assuming a set of discrete controllers in the spatial domain. We point out that the neural networks synthesized offline can be implemented online, since the computation of control only uses the networks.

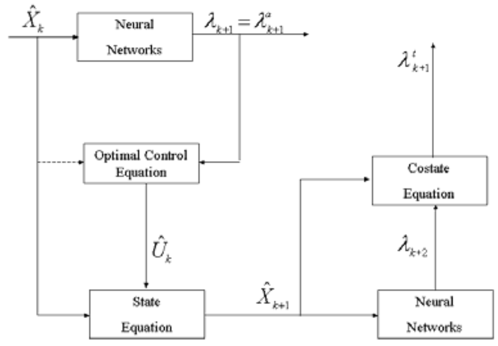

Figure 1: Schematic of neural network synthesis

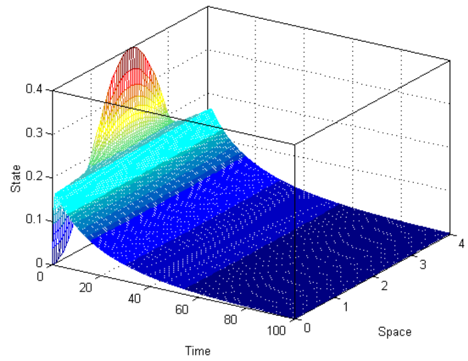

Figure 2: State of the nonlinear system from a sinusoidal initial condition

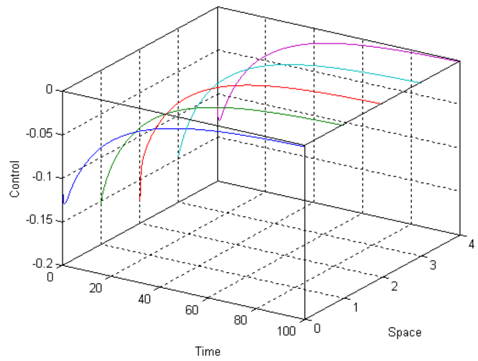

Figure 3: Associated control of the nonlinear system from the sinusoidal initial condition

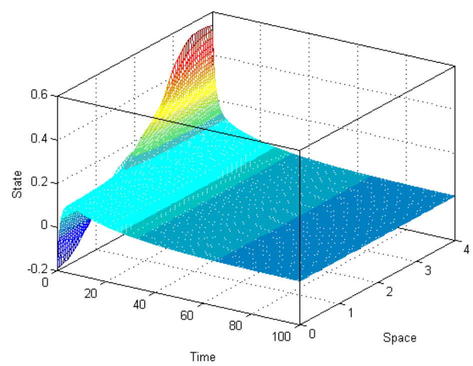

Figure 4: State of the nonlinear system from a random initial condition

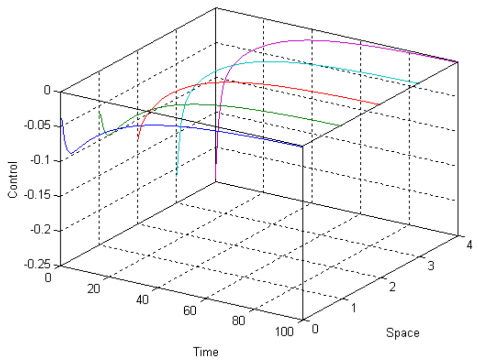

Figure 5: Associated control of the nonlinear system from the random initial condition

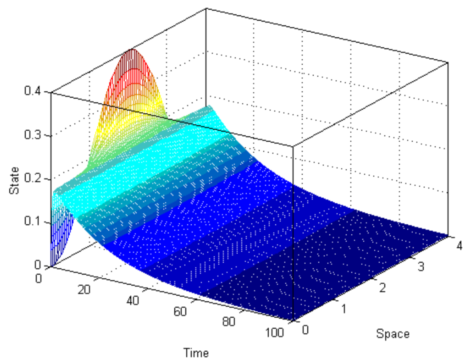

Figure 6: State of the linear system from a sinusoidal initial condition

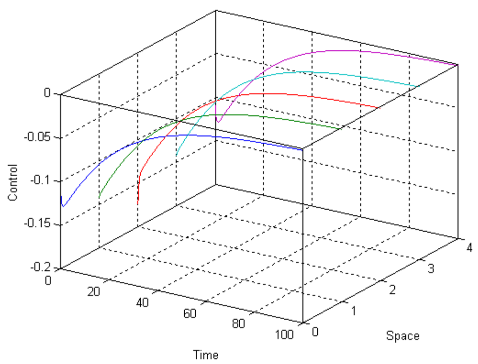

Figure 7: Associated control of the linear system from the sinusoidal initial condition

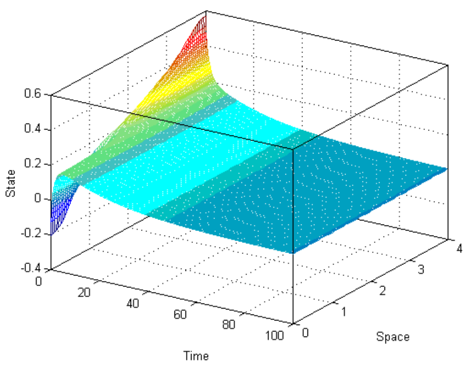

Figure 8: State of the linear system from a random initial condition

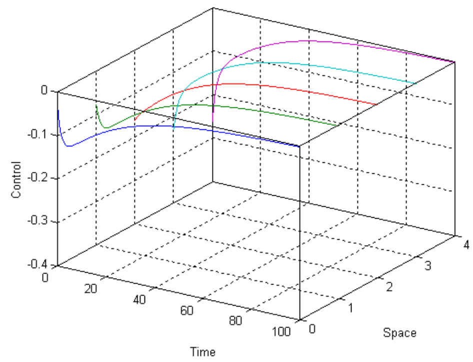

Figure 9: Associated control of the linear system from the random initial condition

ACKNOWLEDGEMENT

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

REFERENCES

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

Banks H. T., Rosario R. C. H and Smith R. C.(2000), Reduced-Order Model Feedback Control Design: Numerical Implementation in a Thin Shell Model, *IEEE Transactions on Automatic Control, Vol. 45, pp. 1312-1324.*

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

Holmes P., Lumley J. L. and Berkooz G. (1996), Turbulence, Coherent Structures, Dynamical Systems and Symmetry, *Cambridge University Press, pp. 87-154.*

Hunt K. J. (1992), Neural Networks for Control Systems – A Survey, *Automatica, Vol. 28, No. 6, pp. 1083-1112.*

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

Singh S. N., Myatt J. H. and Addington G. A (2001), Adaptive Feedback Linearizing Control of Proper Orthogonal Decomposition Nonlinear Flow Models, Proceedings of the American Control Conference, pp. 1533-1538.

Smith G. D., Numerical Solution of Partial (1985), Differential Equations: Finite Difference Methods, *Clarendon Press, Oxford*.

Werbos P. J. (1992), 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.*

White D. and Sofge D. (Eds.) 1992, Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, *Van Nostrand Reinhold.*