Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

VALIDATION OF ADPTIVE-CRITIC BASED INFINITE TIME OPTIMAL NEURO CONTROL FOR DISTRIBUTED PARAMETER SYSTEMS

RADHAKANT PADHI†, S.N. BALAKRISHNAN††, TIMOTHY W. RANDOLPH‡

University of Missouri - Rolla, MO 65409, USA

† Graduate Student, Dept. of Mechanical and Aerospace Engineering. and Engineering Mechanics, padhi@umr.edu

†† Professor (Contact Person), Dept. of Mechanical and Aerospace Engineering. and Engineering Mechanics, bala@umr.edu, Tel: 1(573) 341-4675 , Fax: 1(573) 341-4115

‡ Associate Professor, Mathemetics and Statistics Department, randolph@umr.edu, Tel: 1(573) 341-4643

Abstract.

Recently the necessary conditions of optimality for distributed parameter systems described in discrete domain have been developed, followed by the synthesis of the *infinite time* optimal neuro-controllers in the framework of *adaptive-critic* design. In this paper, we validate this synthesis methodology by comparing it with two other different approaches already established in the literature.

Key Words. Distributed parameter systems, infinite dimensional systems, partial differential equations, optimal control, dynamic programming, neural networks, adaptive-critic synthesis.

1. INTRODUCTION

Distributed Parameter Systems (DPS) are the processes, which are distributed in space and evolving in time. Unlike the lumped parameter systems, the DPS are described by a set of partial differential equations (PDEs) in the state-space. Examples of such systems include aeroelastic systems, vibration of lightly-damped structures, compliant mechanisms, heat transfer processes etc.

The dynamic programming methodology for the solution of optimal control has many desirable features. But the methodology is often overwhelmed by its computational requirements. However in recent times, an advanced neuro-control methodology called the *adaptive-critic* design has given a new perspective to the associated problems of dynamic programming [1, 3, 4, 6]. The advantages of adaptive-critic design include optimal control of the plant maintaining a feedback structure of the controller, control in real time, control from any initial state in the domain of interest to the desired final state etc. Besides, the methodology can handle linear and nonlinear problems directly. As an added advantage, this powerful neuro-control methodology has mathematical and computational simplicity.

Adaptive-critic optimal control methodology has sucessfully been developed for *distributed parameter systems* in the framework of *infinite time* optimal controllers [4]. However to gain confidence, it is highly desirable to verify this methodology. This can be done by comparing it with some of the other methods available in the literature. If not in a rigorous mathematical analysis sense, at least the results for some bench mark problems can be compared.

We have considered the linear heat conduction/ diffusion equation with the Neumann boundary conditions. For this system, it is possible to synthesize infinite time quadratic regulator controllers as a formulation of standard linear quadratic regulator (LQR) solution in ordinary differential equation sense. This can be done after only spatial discretization [5]. Closed form solution for the optimal control of this particular problem is also available in the literature [2]. Thus, it is possible to compare the results from the adaptive-critic methodology with these established techniques. This is the main aim of this paper. We have validated the adaptive-critic methodology by comparing the results of this linear conduction/diffusion optimal control problem.

2. DYNAMIC PROGRAMMING OF DPS

2.1. System Dynamics (State Equation)

We consider a two-dimensional distributed parameter system. *Two dimension* here means two independent variables, one is time and the other is a spatial variable. The necessary conditions of optimality that have been derived and reported in an earlier paper [4] will be recapitulated here in brief. It should be noted that the formulas mentioned here are a bit more general than those reported in [4], and will appear in detail in the journal version of this paper. The system dynamics we consider over here evolves in time and is given by

$$x_{k+1,j} = f_k\left[x_{k,1}, x_{k,2}, \ldots, x_{k,M},\ u_{k,j}\right] \tag{1}$$

where, the subscripts k accounts for evolution with time (time step) and j for the spatial distribution (nodal number). M denotes the number of nodes in the spatial distribution.

2.2. Cost Function

We consider a general cost function of the following form.

$$J = \sum_{k=1}^{N-1} \sum_{j=1}^{M} \Psi_{k,j}\left(x_{k,j},\ u_{k,j}\right) \tag{2}$$

where, N represents the number of discrete time steps. In agreement with the above definition of the cost function, we denote the *cost function from time step* k as

$$J_k = \sum_{\tilde{k}=k}^{N-1} \sum_{j=1}^{M} \Psi_{\tilde{k},j}\left(x_{\tilde{k},j},\ u_{\tilde{k},j}\right) \tag{3}$$

We **define** the *co-state* at time k and node j as

$$\lambda_{k,j} \equiv \partial J_k \,/\, \partial x_{k,j} \tag{4}$$

2.3. Optimal Control Equation

For optimal control, the necessary condition for optimality is given by $\partial J_k \,/\, \partial u_{k,j} = 0$. After some algebra, the optimal control equation obtained is given by

$$\sum_{\tilde{j}=1}^{M} \left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial u_{k,j}}\right) + \sum_{\tilde{j}=1}^{M} \lambda_{k+1,\tilde{j}} \left(\frac{\partial x_{k+1,\tilde{j}}}{\partial u_{k,j}}\right) = 0 \tag{5}$$

2.4. Co-state Dynamics

Substituting for J_k from Eq.(3), after some manipulation, we get

$$\lambda_{k,j} = \sum_{\tilde{j}=1}^{M} \left[\left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial x_{k,j}}\right) + \lambda_{k+1,\tilde{j}} \left(\frac{\partial x_{k+1,\tilde{j}}}{\partial x_{k,j}}\right)\right] + \sum_{\hat{j}=1}^{M} \left[\sum_{\tilde{j}=1}^{M} \left\{\left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial u_{k,\hat{j}}}\right) + \lambda_{k+1,\tilde{j}} \left(\frac{\partial x_{k+1,\tilde{j}}}{\partial u_{k,\hat{j}}}\right)\right\}\right] \left(\frac{\partial u_{k,\hat{j}}}{\partial x_{k,j}}\right) \tag{6}$$

Thus, we have obtained the *state equation*, *co-state equation* and *optimal control equation.* These equations have to be solved simultaneously to obtain the required optimal control. It may be noted that, *along the optimal trajectory*, by using Eq.(5), Eq.(6) can be simplified to

$$\lambda_{k,j} = \sum_{\tilde{j}=1}^{M} \left[\left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial x_{k,j}}\right) + \lambda_{k+1,\tilde{j}} \left(\frac{\partial x_{k+1,\tilde{j}}}{\partial x_{k,j}}\right)\right] \tag{7}$$

3. THE EXAMPLE PROBLEM

This example is the standard linear diffusion/conduction optimal control problem. We reconsider the controller as an *infinite time* one. The most important motivation for choosing this problem is that it can be solved by using different approaches, and hence we can compare the results of the adaptive-critic methodology.

3.1. The Problem

The problem is described, in continuous time, by

$$\frac{\partial x(t,y)}{\partial t} = \frac{\partial x^2(t,y)}{\partial y^2} + u(t,y) \tag{8}$$

$x(0,y) \equiv$ any initial profile within the interest domain.

$$\left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_0} = 0, \qquad \left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_f} = 0$$

The objective is to find the *optimal control* $u(t,y)$, which minimizes the *quadratic cost function*

$$J = \frac{1}{2} \int_{t_0}^{\infty} \int_{y_0}^{y_f} \left[Qx^2(t,y) + Ru^2(t,y)\right]\, dy\, dt \tag{9}$$

where, $x(t,y)$ and $u(t,y)$ are state and control variables at time t and spatial co-ordinate y, Q is the *weighting factor* on state, R is the *weighting factor* on control; t_0 and $t_f \to \infty$ are initial and

final times; y_0 and y_f are initial and final points on the spatial co-ordinate axis.

3.2. Discrete Formulation

The associated cost-function, to be minimized, is given by

$$J = \frac{1}{2}\left[\sum_{k=1}^{\infty} \sum_{j=1}^{M} \left(Q_D \, x_{k,j}^2 + R_D \, u_{k,j}^2 \right) \right] \quad (10)$$

where, Q_D and R_D are the weighting factors on state and control respectively, *in the discrete domain*. For this particular problem,

$$\Psi_{k,j} = \frac{1}{2}\left[Q_D \, x_{k,j}^2 + R_D \, u_{k,j}^2 \right] \quad (11)$$

Then, by applying Eq.(5) and (7), we arrive at the following set of equations as the necessary conditions for optimality, which are the *state*, *co-state* and *optimal control* equations respective.

$$x_{k+1,j} = x_{k,j} + \Delta t \left[\left(x_{k,j+1} - 2x_{k,j} + x_{k,j-1} \right) / \Delta y^2 + u_{k,j} \right] \quad (12a)$$

$$\lambda_{k,j} = \lambda_{k+1,j} + \Delta t \left[\begin{pmatrix} \lambda_{k+1,j+1} - 2\lambda_{k+1,j} \\ + \lambda_{k+1,j-1} \end{pmatrix} / \Delta y^2 + Q_D \, x_{k,j} \right] \quad (12b)$$

$$u_{k,j}^* = -R_D^{-1} \, \lambda_{k+1,j} \quad (12c)$$

where, Δt and Δy are the step sizes of discretization in time and spatial variables respectively. Together with the necessary conditions of optimality, we have to satisfy the following initial, transversality and boundary conditions.

$x_{0,k} =$ can be *any point* in the domain of interest

$$\lambda_{N,j} = 0, \quad as \; N \to \infty$$

$$x_{k,0} = x_{k,1}, \qquad x_{k,M+1} = x_{k,M} \quad (13)$$

$$\lambda_{k,0} = \lambda_{k,1}, \qquad \lambda_{k,M+1} = \lambda_{k,M}$$

4. SOLUTION TECHNIQUES

4.1. Adaptive-Critic Synthesis

The adaptive-critic synthesis procedure is discussed, in fair detail, in Ref.[4]. However, the core of the technique, which is the iterative training between Critic and Action neural networks, is recapitulated here in brief. The training processes are depicted in Figure-1 and Figure-2.

We synthesize a set of M critic networks, for $k = N-1$, with input $x_{N-1,j}$ and output $\lambda_{N-1,j}$ as per the following steps (Figure-1). Assume $x_{k,j}$. Get $u_{k,j}$ from the trained action networks. Then get $x_{k+1,j}$ from the *state equation* (12a). Input $x_{k+1,j}$ to the trained set of critic networks at $(k+1)^{\text{th}}$ time step, to get $\lambda_{k+1,j}$. Now, with the availability of $x_{k,j}$ and $\lambda_{k+1,j}$, calculate $\lambda_{k,j}$ from the *co-state* equation (12b). Train the set of critic networks with **input** $x_{k,j-1}$, $x_{k,j}$, $x_{k,j+1}$ and **output** $\lambda_{k,j}$ for all the networks related to the *internal node points*. For those intended for the *boundary node points*, we consider either $x_{k,1}$, $x_{k,2}$ or $x_{k,M-1}$, $x_{k,M}$ as the **input**.

After that we focus on action network synthesis. The training process is carried out in the following steps (see Figure-2). Assume random $x_{k,j}$, within the relevant range, and input it to the action networks, to get $u_{k,j}$. Use *state equation* (12a) and the boundary condition [Eq.(13)] to get $x_{k+1,j}$ uniquely. Input $x_{k+1,j}$ to the *trained* set of critic networks to get $\lambda_{k+1,j}$. Get the optimal control $u_{k,j}^*$ from Eq.(12c). Train the networks at k^{th} time step with **input** $x_{k,j-1}$, $x_{k,j}$, $x_{k,j+1}$ and **output** $u_{k,j}^*$ for all the networks related to the *internal node points*. For those intended for the *boundary node points*, we consider either $x_{k,1}$, $x_{k,2}$ or $x_{k,M-1}$, $x_{k,M}$ as the **input**.

Once this process of action synthesis is over, we revert to critic synthesis again. The alternate critic and action network training process is continued till no noticeable change in the output is observed in the outputs in the successive training. Then the networks converge to give the true optimal relationships.

4.2. LQR Solution

The essential idea of this approach can be found in Sage [5]. Essentially, only after spatial discretization of this linear PDE system and using the boundary conditions, we arrive at a system of linear ordinary differential equations (ODEs), in the following form.

$$\dot{X} = AX + BU$$

$$A = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & & \\ & & & \ddots & 1 \\ & & & 1 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \tag{15}$$

The associated cost function takes the form

$$J = \int_0^\infty (X^T Q\, X + U^T R\, U)\, dt \tag{16}$$

where,

$$Q \;=\; R \;=\; (dy/2)\,.\,diag\left(\frac{1}{2}, 1, 1, \;\ldots\;, 1, 1, \frac{1}{2}\right) \tag{17}$$

Then using the standard infinite-time *LQR* theory, solution for the controller can be found easily. Towards this objective, one can simply use the standard *lqr* or *lqr2* functions already available in *MATLAB.*

The controller u is given by:

$$U \;=\; -K\,X \;=\; -R^{-1}B^T S\,X \tag{18}$$

where, S matrix is obtained from the solution of the following algebraic Ricatti equation.

$$SA + A^T S - SBR^{-1}B^T S + Q = 0 \tag{19}$$

4.3. Closed Form Solution

For the particular optimal control problem, closed form solution also exists and have been derived in detail in the literature (Curtain and Zwart [2]). The final form of the control solution, for the infinite-time controller is given by:

$$u^*(t,y) \;=\; -\left[\begin{array}{l} (1/L)\int_0^L x(t,\tilde{y})\, d\tilde{y} \;+ \\ \sum_{n=0}^{\infty}\left\{(2/L)\left(-\left(\frac{n\pi}{L}\right)^2 + \sqrt{\left(\frac{n\pi}{L}\right)^4 + 1}\right).\right. \\ \left.\left(\int_0^L x(t,\tilde{y})\, Cos\left(\frac{n\pi\tilde{y}}{L}\right) d\tilde{y}\right) Cos\left(\frac{n\pi y}{L}\right)\right\} \end{array}\right] \tag{20}$$

As an observation, if the initial profile is *symmetric wrt.* the centre point of the spatial domain, since the boundary conditions are also symmetric, the integral under the infinite sum in Eq.(19) simply becomes zero for each term in the summation and hence the entire sum becomes zero for all time t. Thus, we should arrive at constant controllers for all the node points at any particular time.

5. NUMERICAL RESULTS

For our numerical experients, we set the values as $t_0 = 0$, $y_0 = 0$, $y_f = 4$. For discretization, we have considered $\Delta t = 0.02$, $\Delta y = 1$ (same as Ref.[5]). We have assumed that the initial profiles lie within ± 0.25, and hence, have attempted to solve the problem for all possible initial profiles lying inside this boundaries. Here all values are assumed to be in compatible units.

First of all, it can be noted that the states are driven towards zero as N increases, at all the node points. Same trend can be noticed in the evolution of control with time as well. As a comment, we also observe from Eq.(12c), that $\lambda_N \to 0$ as $N \to \infty$, since $u_N \to 0$ as $N \to \infty$. This way the synthesis process is also seen to satisfy the necessary *transversality* condition for optimality given in Eq.(13).

Figure-3 and Figure-4 gives the comparison of state history and control history solution of the adaptive-critic methodology with the LQR solution, from a sinusoidal initial profile $x(0, y) = 0.25\; Sin\left(\pi y / L\right)$, where $L = 4$ is the length of the finite spatial domain considered. Similarly, Figure-5 and Figure-6 give the comparisons with the closed form solution. It can be observed that the states at all the node points develop quite closely, in both the comparisons. However, the controllers are a bit far off. This is mainly because of the fact that both the LQR as well as the closed form solutions demand the state information in the entire spatial domain for the controller at any node. However, we have purposefully did not synthesize our networks with all the states as the input. This is because in that case, as the node numbers increase, the network size becomes very high. It leads to considerably slower training and possibly inferior optimization.

However, the sub-optimality of the adaptive-critic controllers need not entirely be from not feeding all the states as input to the neural networks. It can also arise because of *insufficient training* of the networks. In fact, as the solution process of adaptive-critic methodology is essentially backward in time, it is highly necessary to cover all the function space in training, to cover all possible profiles as the time evolves. This in fact is a hard task and we are still in a process of searching for a good answer to this question. However, since the synthesis process of the

networks were carried out with *random* profiles, it makes sense to compare the solution trend for some arbitrary random profiles. Such a comparison of the adaptive-critic solution with the LQR results is shown in Figures-7 and 8. From these figures, it is quite clear that the sub-optimal solution of the adaptive-critic methodology is in fact *very close* to the optimal solution from the LQR approach.

As a side observation, since the initial profile is symmetric, as pointed out earlier, the controller in closed form solution is expected to be constant at all the node points at all the time. This infact is clear from Figure-6. As another side note, one can notice that the there is some minor discrepency between the LQR and closed form solutions themselves. It mainly arises due to the process of discritization. This error is also present in the adaptive-critic methodology. However, it can arbitrarily be reduced by choosing proper Δt and Δy, so that the ratio $\frac{\Delta t}{(\Delta y)^2}$ becomes very small.

6. CONCLUSION

After comparing the results from adaptive-critic approach with LQR and closed form solution, it can confidently be claimed that the formulation and approach of the adaptive-critic methodology described in [4] and also outlined briefly in this paper was fine. The minor discrepancies of the results are mainly due to the insufficient training of the networks. Sub-optimality of the adaptive critic solution also arises due to the choice of the network structures, where we have intentionally taken only the states at current and neighbouring nodes as input to the networks. This is mainly to avoid large size of the networks and the associated difficulties. On the other hand both LQR and closed form solution demand full state feedback.

Acknowledgement.

This research was supported by NSF-USA grant ECS-9976588. The authors are thankful to NSF for the support. Many thanks to Dr. P. Werbos, Program Director, ENG/ECS - NSF.

References

1. Balakrishnan S. N. and Biega V., *Adaptive-Critic Based Neural Networks for Aircraft Optimal Control*, J. of Guidance, Control and Dynamics, Vol. 19, No. 4, July-Aug. 1996, pp. 893-898.
2. Curtain R. F. and Zwart H. J., *An Introduction to Infinite Dimensional Linear Systems Theory*, Springer-Verlag, Newyork, 1995, Chapter 6.
3. Han D. and Balakrishnan S. N., *Adaptive-Critic Based Neural Networks for Agile Missile Control*, AIAA-98-4495.
4. Padhi R. and Balakrishnan S. N., *Infinite Time Optimal Neuro Control for Distributed Parameter Systems*, ACC00-AIAA1023.
5. Sage A. P., *Optimum Systems Control*, Prentice Hall, 1968, Chapter 7, pp. 137-163.
6. Werbos P., *Neurocontrol and Supervised Learning: An Overview and Evaluation*, A Chapter in the Handbook of Intelligent Control, Van Nostrand Reinhold, 1992.

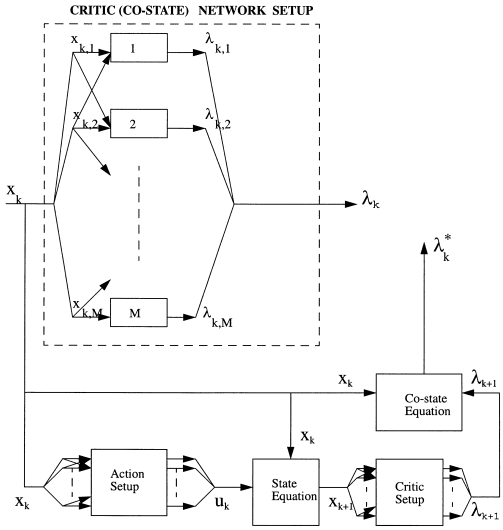

Figure-1: Schematic of Critic Synthesis Pocedure

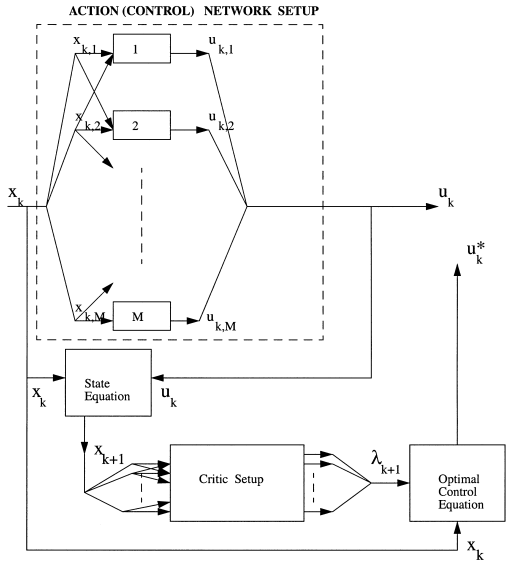

Figure-2: Schematic of Action Synthesis Procedure

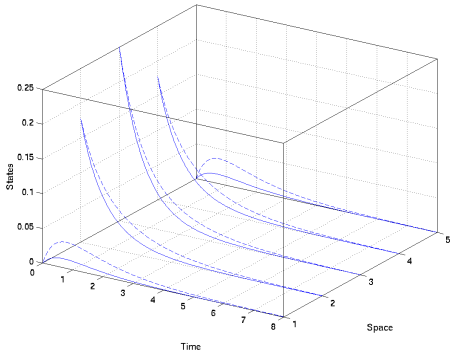

Figure-3: Comparison of state trajectories with LQR solution from a sinusoidal initial profile

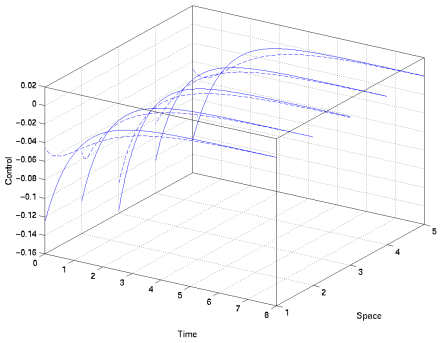

Figure-4: Comparison of control histories with LQR solution from the sinusoidal initial profile

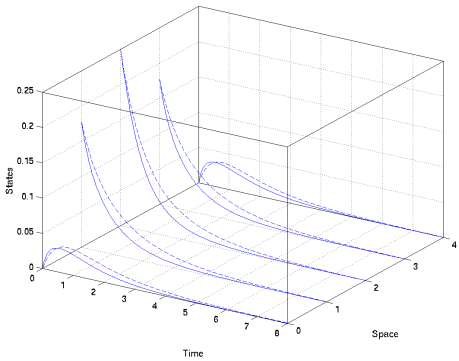

Figure-5: Comparison of state trajectories with closed-form solution from a sinusoidal initial profile

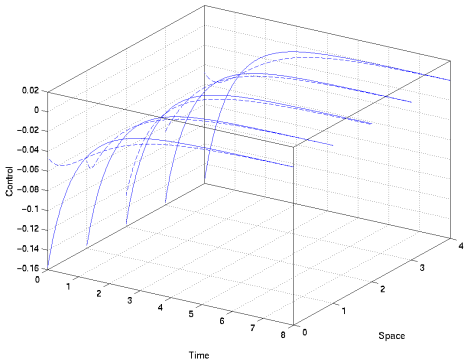

Figure-6: Comparison of control histories with closed-form solution from a sinusoidal initial profile

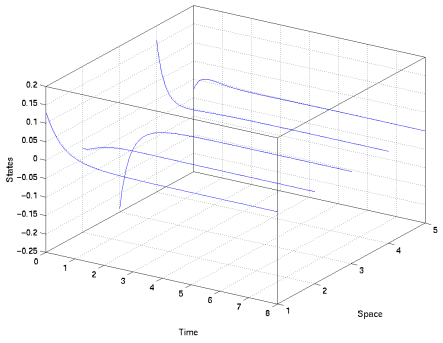

Figure-7: Comparison of state trajectories with LQR solution from a random initial profile

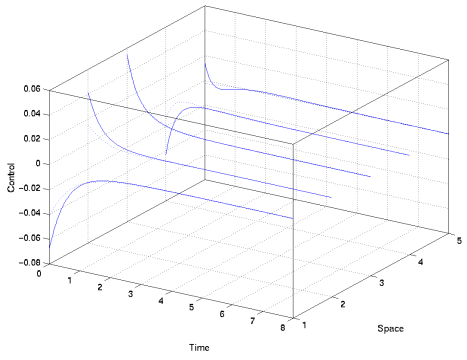

Figure-8: Comparison of control histories with LQR solution from a random initial profile