01 Jan 2002

Experimental Implementation of Adaptive-Critic Based Infinite Time Optimal Neurocontrol for a Heat Diffusion System

Prashant Prabhat

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

Dwight C. Look
Missouri University of Science and Technology

Recommended Citation

P. Prabhat et al., "Experimental Implementation of Adaptive-Critic Based Infinite Time Optimal Neurocontrol for a Heat Diffusion System," *Proceedings of the 2002 American Control Conference (2002, Anchorage, AK)*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2002.
The definitive version is available at https://doi.org/10.1109/ACC.2002.1025190

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

EXPERIMENTAL IMPLEMENTATION OF ADAPTIVE-CRITIC BASED INFINITE TIME OPTIMAL NEUROCONTROL FOR A HEAT DIFFUSION SYSTEM

Prashant Prabhat[1], S.N.Balakrishnan[2] and D.C.Look, Jr.[3]

Dept. of Mechanical and Aerospace Engineering and Engineering Mechanics

University of Missouri - Rolla, MO 65409, USA
pprabhat@umr.edu bala@umr.edu look@umr.edu

Abstract

Recently the synthesis methodology for the *infinite time* optimal neuro-controllers for PDE systems in the framework of *adaptive-critic* design has been developed [1,2]. In this paper, first we model an experimental setup representing one dimensional heat diffusion problems. Then we synthesize and implement an *adaptive-critic* based neuro-controller for online temperature profile control of the experimental setup.

1. Introduction

Many chemical processes involving heat and / or mass transfer or chemical reactions are modeled by partial differential equations of the parabolic type. The dynamic behavior of most of the diffusional process is well understood and the parameter and the boundary conditions are well known [3]. Whereas in our experimental setup it was not only desired to synthesize and implement the optimal neuro-controller, but also to study the behavior of the system and then model its parameters. The first attempts to treat parameter identification in distributed systems involved approximating the system by a lumped model. The method assumed the system to be linear and developed a transfer function approximating the distributed model. Various methods for estimation of diffusivity of a rod have been compared in [4]. In this reference [4], the governing partial differential equation is approximated by a finite difference method and the dynamic (time series) data is used for parameter identification. We follow a similar approach in our study but for a more complicated system where it is not only desired to arrive at a model of thermal diffusivity of the system but also to find a mathematical model of the distributed source (control) term.

The design of controllers for PDE based systems have been studied and implemented in [3], [5] and [6]. The control of a diffusional process by approximating it as a lumped-parameter system, with the technique of finite Fourier transform has been studied and implemented in [3]. In studies [4] and [5], the control is designed by approximating the distributed parameter system into lumped parameter system using an integral transform. In all these studies it was desired to maintain a particular temperature profile in the steady state for the thermal system. In our study, we do not use any of these transforms, instead we follow the methodology of adaptive critic design as studied in [1,2] and use dynamic programming methodology for optimal controller synthesis. The advantages of adaptive critic design include optimal control of the plant maintaining a feedback structure of the controller in real time from any initial state in the domain of interest to the desired final state. In addition, this method can handle linear and nonlinear problems directly, retaining the same structure.

In our current study a linear heat diffusion equation with Neumann boundary conditions has been considered and the optimal neuro-controller has been successfully implemented on the experimental setup for on-line temperature profile control.

2. Experimental Setup and its Modeling

In this section we discuss the development of the experimental setup which would represent the one dimensional heat diffusion problem. The main difficulty encountered, related to the development of the mathematical model of the experimental setup. As opposed to most studies e.g. [3,5,6], neither the behavior nor a model of the experimental setup was available for controller synthesis and implementation. The main objective of this section is to model the experimental setup including the model of each heater.

2.1. Analytical Background

The linear diffusion problem is described by

$$\frac{\partial T(t,y)}{\partial t} = \alpha(y)\frac{\partial^2 T(t,y)}{\partial y^2} + \beta(y)u(t,y) \tag{1}$$

The Neumann boundary conditions are

$$\left.\frac{\partial T(t,y)}{\partial y}\right|_{y=y_0} = 0 \tag{2}$$

$$\left.\frac{\partial T(t,y)}{\partial y}\right|_{y=y_f} = 0 \tag{3}$$

Note that $T(0,y)$ represents any initial profile within the domain of interest $y_0 \le y \le y_f$ and $T(t,y)$ represents the transient or dynamic temperature profile over the entire domain y; $\alpha(y)$ is the thermal diffusivity and $\beta(y)$ is the heat input / heat generated by means other than thermal conduction. In this discussion it will be referred to as the source term distribution over the entire domain. $u(t,y)$ is the control term denoting the current load at which the heater is operating. Its value is 1 for a particular heater if it is ON at 100% of its load and is 0 if it is OFF.

2.2 Experimental Setup

In order to present an overview of the hardware setup to implement the one dimensional diffusion problem, note that the diffusion problem is represented using a series of aluminum slabs and heaters placed one after another as shown in Fig.1. Ten heaters and nine aluminum slabs were assembled. Mica heaters were selected for their small thickness of 0.025"elements. These heaters are 6in. in diameter. Aluminum slabs were used for their high thermal conductivity. They are also 6in. in diameter, and ½in. thick. A hole of 0.125"in diameter was drilled radially into the side of each of the aluminum slabs in order to place a thermocouple within the slab.

---

1. Graduate Student, Mechanical Engineering
2. Professor, Aerospace Engineering (Contact Person)
3. Professor Emeritus, Mechanical Engineering

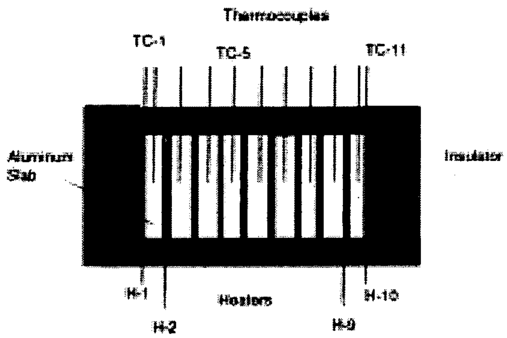

Fig.1 Cross section of Experimental Setup

Since these holes facilitate temperature measurements they were 2" deep. K-type thermocouples were used for the temperature measurement; their tips were placed at the bottom of 2" deep hole.

The software used for input and output of data was LabVIEW installed on a Pentium PC. The temperature readings from the thermocouples were linearized in LabVIEW using the cold junction compensation temperature. The heaters were connected to a 120 volts power supply through Solid State Relays (SSR). The ON and OFF sequence as well as load control of the heaters was achieved using LabVIEW. To control the operating load of the heaters, they were switched ON for a predetermined period of time during each cycle time, Eq.(4). During each cycle, the data was written to a digital I/O for predetermined number of times after an interval of thirty-five milliseconds (limited by the SSR specification), updating the status of each of the heaters to ON or OFF depending on the current desired load for each heater.

Cycle Time(Sec) =0.035 x Number of times data written to digital I/O during each cycle (4)

For development of the model for this system, data was written to the digital I/O 100 times during a cycle time of 3.5 sec.

2.3. Modeling

The methodology and results of modeling the setup are presented in this section by noting that various approaches for identification of parameters for distributed systems have been discussed by Goodson and Polis in [7,8]. Some of the methods discussed are finite differences, modal approximation, Laplace transform and infinite product expansions, analytic solutions and characteristic methods. A finite difference scheme was used in this study to discretize the system and the Tridiagonal Matrix Algorithm (TDMA) [9,10] was used for numerical simulation. By matching the simulation results with experimental results, the mathematical model of the system was developed.

Some of the primary equations involved in TDMA development are recapitulated below. Using backward finite difference method, Eq.(1) can be written as

$$\frac{T_i^k - T_i^{k-1}}{\Delta t} = \alpha_i \left[\frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta y^2} \right] + \beta_i u_i^k \tag{5}$$

where, k represents the time (t) increment and i represents the space (y) increment. Eq.(5) can be rearranged into the standard form of the descritzed equation for TDMA applications as follows[10]:

$$-A_i T_{i+1}^k + B_i T_i^k - C_i T_{i-1}^k = D_i \qquad (6),$$

$$A_i = \frac{\alpha_i}{\Delta y^2} \tag{7}$$

$$B_i = \left[\frac{2\alpha_i}{\Delta y^2} + \frac{1}{\Delta t} \right], \tag{8}$$

$$C_i = \frac{\alpha_i}{\Delta y^2}, \text{ and} \tag{9}$$

$$D_i = \frac{T_i^{k-1}}{\Delta t} + \beta_i u_i^k \tag{10}$$

Similar development can be followed for homogeneous part of Eq.(1).

2.3.1 Model of $\alpha(y)$, $\beta(y)$

In this heat diffusion problem represented by Eqs. (1)-(3) it is desired to find terms for the coefficients $\alpha(y)$ and $\beta(y)$. This was achieved in two steps. Initially, experiments and simulations were conducted to arrive at a form of the $\alpha(y)$. To do so the control term, $\beta(y)u(t,y)$, was neglected while arriving at a model of $\alpha(y)$. This was possible because the system is linear in nature and it was desired to find a solution of the homogeneous equation part of Eq. (1) first. And, in the second phase of modeling, the control term was included based on the superposition principal.

2.4 Experimentation and Numerical Simulations:

Results of some experiments that were conducted for modeling the setup are presented in this section based on the selection of the parameters of the numerical simulation using TDMA such that the experimental results were in agreement with the simulation results.

For the numerical simulation results to be acceptable, the truncation error must be small and the finite difference representation of the marching method needs to meet the conditions of consistency and stability [9]. To keep the size of round off errors down, it is sufficient [11] that

$$A_i > 0 \ , B_i > 0 \, , C_i > 0 \text{ and } B_i > A_i + C_i \tag{11}$$

These conditions are satisfied by the numerical scheme chosen here. Also, since the coefficients A_i, B_i and C_i are always positive, the numerical results are stable. The number of grids, which gave numerically consistent results, was determined after choosing various numbers of grids for simulation. It was observed that if number of grid points selected was 217 nodes, the numerical results were consistent with the experimental results. Further, this lead to development of grid independent solution. For the numerical simulation of TDMA, some of the parameters used were $\Delta t = 0.1\text{sec}$ and $\Delta y = 5.267 \times 10^{-4}$ meters. The results were simulated at 10 sec interval. To simulate the results after each 10 sec interval, the temperature profile over the entire domain that was obtained at the last iteration was used as the initial condition for the next set of simulation.

The experiments were conducted by heating the setup by the heaters until a desired temperature was reached. Temperature data was collected from all 11 thermocouple readings with respect to time.

2.4.1 Model Of Heater-1

This section describes the development of a model for heater-1 shown as H-1 in Fig.1. In order to accomplish this, only heater 1 was kept ON at 100% of its load capacity. Fig.2 is an illustration of the resulting experimental results of this setup. When the heater-1 was turned OFF, Fig.3 depicts the temperature response for the 11 thermocouples.

As discussed in Sec. 2.3.1, the first step of modeling the system was to arrive at a value of $\alpha(y)$; this essentially means predicting the system behavior by homogeneous part of Eq.(1). The TDMA algorithm was used for simulation with the initial temperature profile generated by linear interpolation between the 11thermocouple readings at the initial time represented in Fig. 3. In our initial attempt of simulation, $\alpha(y)$ was assumed to be constant over the entire domain. The TDMA simulations were made to match the cooling curve plots of the 11 thermocouples in Fig.3 by selecting different values of α. The plots below show the best possible match that could be obtained for the TDMA simulation and the experimental results. It can be noted in Fig.2 that the chosen value of $\alpha = 8.5\times10^{-7} m^2/\text{sec}$, leads to very good match between experimental results with that of TDMA simulations. Also the current choice of α, as constant over the spatial domain, can be justified by our simulation results.

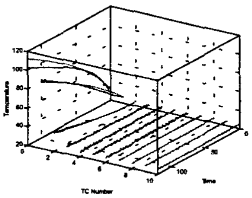

Fig. 2 Experimental results and TDMA simulations for heating curve of heater-1

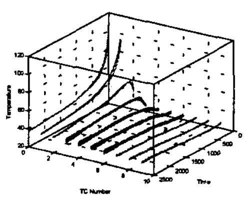

Fig 3. Experimental results and TDMA simulations for cooling curve of heater-1

As discussed above, Eq. (1) represents the system response in the presence of source term. Fig.2 is the corresponding dynamic system response to heater 1 ON. While modeling the heating curve, attempts were made to determine a distributed source term model of $\beta(y)$. Although, the heater is physically located at only one of the nodes of the discretized system for TDMA simulation, a point source term model was not able to represent the system behavior as observed in the dynamic system response in Fig.3. That is why a distributed source term model was developed which could represent the system behavior. TDMA simulation as discussed above was used to arrive at this model of the heater-1 shown in Fig.4. It can be noted that in this TDMA simulation, $u(t,y)$ was assumed to be 1 for heater-1, which was to represent that the heater-1 was switched ON at 100% load capacity. $u(t,y)$ was zero for all other heater models since they were switched OFF. The initial temperature for TDMA simulation was 26^{0}C, room temperature, in Fig.3. By trial and error a distributed source term model was made as shown in Fig.4, which represents $\beta(y)$ for heater-1. Using the $\beta(y)$ distribution shown in Fig.4, the experimental results were forced to be in agreement with the simulation results for all 11 thermocouple results as shown in Fig.2.

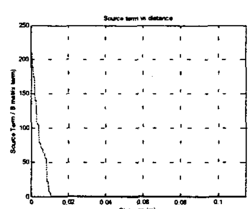

Fig. 4 Source term model for heater-1

The choice of above model for $\beta(y)$ as shown in Fig.4, can be justified in terms of the effect of the source term over the domain of interest. As in Fig. 4, the maximum effect of the heater-1 is on the two sides adjacent to its physical location. And, in general, the effect of the heater decreases as we move farther away from the heater location. By effect, we mean the heat flux generated in the system, which in turn leads to temperature rise. It can be noted that even though the source term, which represents a heater in the experimental setup which is located at node-1 (extreme left) in the descretized system, is being represented by a distributed source term model as in Fig.4. The justification for selection of a distributed model for the heater can be validated by our experimental and simulation results, which are in agreement. This procedure was repeated to find a distributed source term model of each heater.

3. Diffusion/Conduction Optimal Control Problem

After development of the model of the system, it was desired to synthesize the optimal neuro-controller, which we discuss here. In the beginning of this section we recapitulate the methodology for dynamic programming of a distributed parameter system as discussed in [1], and proceed to develop the optimal diffusion problem.

3.1 System Dynamics (State Equation)

We consider a two-dimensional distributed parameter system. *Two dimension* means the two independent variables of time (t) and space (y). The system dynamics we consider evolves in time and is given by

$$x_{k+1,j} = f_k\left(x_{k,1}, x_{k,2}, \ldots x_{k,j}, \ldots, x_{k,M}, u_{k,j}\right) \tag{12}$$

The subscripts k accounts for evolution with time (time step) and j represents the spatial distribution (nodal number). M denotes the final node number in the spatial distribution.

3.2 Cost Function

We consider a general cost function of the following form:

$$J = \sum_{k=1}^{N-1} \sum_{j=1}^{M} \Psi_{k,j}\left(x_{k,j}, u_{k,j}\right) \tag{13}$$

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

$$J_k = \sum_{\tilde{k}=k}^{N-1} \sum_{j=1}^{M} \Psi_{\tilde{k},j}\left(x_{\tilde{k},j}, u_{\tilde{k},j}\right) \tag{14}$$

where $\tilde{k}$ is a dummy variable. Finally, we define the *Co-state* as

$$\Delta y \lambda_{k,j} \equiv \partial J_k / \partial x_{k,j} \tag{15}$$

where Δy is the spatial increment.

3.3 Optimal Control Equation

For optimal control, the necessary condition for optimality is given by

$$\partial J_k / \partial u_{k,j} = 0 \tag{16}$$

After some algebraic manipulation, the optimal control equation obtained is given by

$$\sum_{j=1}^{M}\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \sum_{\bar{j}=1}^{M} \lambda_{k+1,\bar{j}} \left(\frac{\partial x_{k+1,\bar{j}}}{\partial u_{k,j}}\right)\Delta y = 0 \tag{17}$$

3.4 Co-state Dynamics

Substituting for J_k from Eq. (14) and doing further algebraic manipulation, yields

$$\Delta y\lambda_{k,j} = \sum_{\tilde{j}=1}^{M}\left[\left(\frac{\partial\Psi_{k,\tilde{j}}}{\partial x_{k,j}}\right)+\lambda_{k+1,\tilde{j}}\Delta y\left(\frac{\partial x_{k+1,\tilde{j}}}{\partial x_{k,j}}\right)\right] + \sum_{j=1}^{M}\left[\sum_{\tilde{j}=1}^{M}\left\{\left(\frac{\partial\Psi_{k,\tilde{j}}}{\partial u_{k,\tilde{j}}}\right)+\lambda_{k+1,\tilde{j}}\Delta y\left(\frac{\partial x_{k+1,\tilde{j}}}{\partial u_{k,\tilde{j}}}\right)\right\}\right]\left(\frac{\partial u_{k,\tilde{j}}}{\partial x_{k,j}}\right) \tag{18}$$

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. By using Eq.(17), Eq.(18) can be simplified to

$$\Delta y\lambda_{k,j} = \sum_{\tilde{j}=1}^{M}\left[\left(\frac{\partial\Psi_{k,\tilde{j}}}{\partial x_{k,j}}\right)+\lambda_{k+1,\tilde{j}}\Delta y\left(\frac{\partial x_{k+1,\tilde{j}}}{\partial x_{k,j}}\right)\right] \tag{19}$$

3.5 Optimal Diffusion Problem

In this section we reconsider the development of an *infinite time* controller. In order to do this, recall that the problem is described by the PDE and the boundary conditions of Eqs. (1) – (3). For the sake of convenience, we represent Eq.(1)-(3) as follows:

$$\frac{\partial x(t,y)}{\partial t} = \alpha(y)\frac{\partial x^2(t,y)}{\partial y^2} + \beta(y)u(t,y) \tag{20}$$

$$\left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_0} = 0, \qquad \left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_f} = 0 \tag{21}$$

Note that $x(0,y)$ represents any initial temperature profile within the interest domain.

Here, the state variable is $x(t,y)$ and y is the spatial variable. The objective is to find the *optimal control* formulation $u(t,y)$, which minimizes the *quadratic cost function,* defined as

$$J = \frac{1}{2}\int_{t_0}^{\infty}\int_{y_0}^{y_f}\left[Qx^2(t,y)+Ru^2(t,y)\right]\,dy\,dt \tag{22}$$

where, $u(t,y)$ is the control variable at time t and spatial co-ordinate y, Q is the *weighting factor* on the state variable, R is the *weighting factor* on the control variable. Further t_0 and $t_f \to \infty$ are initial and final times where y_0 and y_f are initial and final points on the spatial co-ordinate axis.

3.6 Discrete Formulation

The discretized quadratic cost-function, which is to be minimized, is given by

$$J = \frac{1}{2}\left[\sum_{k=1}^{\infty}\sum_{j=1}^{M}\left(Q_D\, x_{k,j}^2 + R_D\, u_{k,j}^2\right)\right] \tag{23}$$

where

$$Q_D = \Delta t\Delta y Q \tag{24}$$

$$R_D = \Delta t\Delta y R \tag{25}$$

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

$$\Psi_{k,j} = \frac{1}{2}\left[Q_D\, x_{k,j}^2 + R_D\, u_{k,j}^2\right] \tag{26}$$

Then, by applying Eq.(17) and (19), we arrive at the following set of equations as the necessary conditions for optimality. The following equations are the *State, Co-state* and *Optimal Control* equations respectively

$$x_{k+1,j} = x_{k,j} + \Delta t\left[\alpha_j\left(x_{k,j+1} - 2x_{k,j} + x_{k,j-1}\right)/\Delta y^2 + \beta_j u_{k,j}\right] \tag{27}$$

$$\lambda_{k,j} = \lambda_{k+1,j} + \Delta t\left[\alpha_j\left(\lambda_{k+1,j+1} - 2\lambda_{k+1,j} + \lambda_{k+1,j-1}\right)/\Delta y^2 + Q_D\, x_{k,j}\right] \tag{28}$$

$$u_{k,j}^{*} = -R_D^{-1}\,\beta_j\lambda_{k+1,j} \tag{29}$$

where Δt and Δy are the step sizes of discretization in time and spatial variables respectively and u^* is optimal control. Together with the necessary conditions of optimality, we have to satisfy the following initial, transversality and boundary conditions. If $x_{0,k}$ can be *any point* in the domain of interest, then

$$\lambda_{N,j} = 0, \quad as\ N \to \infty,$$
$$x_{k,0} = x_{k,1}, \qquad x_{k,M+1} = x_{k,M}, \text{ and} \tag{30}$$
$$\lambda_{k,0} = \lambda_{k,1}, \qquad \lambda_{k,M+1} = \lambda_{k,M}$$

4.0 Adaptive-Critic Controller Synthesis

The adaptive-critic synthesis procedure is discussed, in fair detail, in [1]. However the core of the technique, which is the iterative training between Critic and Action neural networks, is recapitulated here in brief. The philosophical justification for adaptive-critic structures has been discussed by Werbos [12].

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. Assume $x_{k,j}$ as some random "smooth" profile. Get $u_{k,j}$ from the trained action networks. Then get $x_{k+1,j}$ from the *State Equation* Eq.(27). 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* Eq.(28). 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. Assume random $x_{k,j}$, within the relevant range, and input it to the action networks, to get $u_{k,j}$. Use *State Equation* Eq.(27) and the boundary condition [Eq.(30)] 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.(29). 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 relationship.

In our controller synthesis, we set the numerical values as $Q_D = 1$, $R_D = 1$, $t_0 = 0$, $y_0 = 0$ and $y_f = 1.143 \times 10^{-1} meters$, which is the length of the experimental setup in Fig.1. For discretization of the system, we used $\Delta t = 0.01 \sec$ and $\Delta y = 1.27 \times 10^{-2} meters$, which is the distance between two adjacent heater locations placed equidistant from each other. As our experimental setup has ten heaters, we select ten node points for neural network synthesis. Thus there are ten action and critic networks, one for each of the nodes. The convergence criteria were achieved when no noticeable change was observed in the output of the networks in successive training.

It can be noted that the state and co-state equations in Eq.(27) and Eq.(28) respectively need α_j and β_j values from the experimental model of the system. Since, in our modeling α_j is a constant value over the entire domain, a value of $\alpha_j = 8.5 \times 10^{-7} m^2/\sec$ was used in the equations. However in our TDMA model, the β_j value is not constant. Its value changes with every heater model. In fact, every heater model has a distributed source term model as shown in Fig.4 for heater 1. It can be recapitulated that TDMA used 217 nodes to model the system, whereas our current neuro controller synthesis required an equivalent 10 nodes representation of the source terms. This was achieved by finding the area under the curve of source term (as in Fig.4 for heater-1) for each of the heaters and to use this numerical value as β_j for that node of neural network. Since, the β term in the TDMA represents the rate of heat generation per unit volume, for the 1-D heat conduction, integration of this term over the length, gives an equivalent representation of rate of heat generation term to be incorporated in the neural network by a point source.

In our current implementation, the network structure is retained similar to that in [1]. For the interior node points, we have used a multi-layer feed forward network of the form $\pi_{3,5,5,1}$ for the critic training and similar network for the action training. Here, $\pi_{3,5,5,1}$ denotes a neural network with 3 neurons in the input layer, 5 neurons each in the two other hidden layers and 1 neuron in the output layer. For the boundary node points, however, the network structure is taken to be of the form $\pi_{2,5,5,1}$. This is because we have decided not to input the state values at the fictitious node points as input to the network, since the associated boundary conditions lead to both $x_0 = x_1$ and $x_{M+1} = x_M$. We have taken tangent sigmoid function for all the hidden layers and linear function for the output layer.

To achieve convergence in training the networks the choice of the $x_{k,j}$ value plays an important role. Based on the understanding of the system where the neuro-controller is to be implemented, $x_{k,j}$, the state information should be such that it represents a "feasible" state. So a Fourier series based algorithm was developed to generate "smooth state profiles", for training the networks. Using this algorithm, $x_{k,j}$ profile has smooth gradient over the entire length.

5.0 Online Implementation Of The Neuro Controller

After synthesizing the neuro-controller it was implemented in LabVIEW for online optimal control of desired temperature profile. As we synthesize a regulator problem, the error in temperature at each node location with respect to the desired temperature is feed to the neuro-controller after normalization. Since no sensors were available at exact heater locations, the temperature at the heater location was approximated by linear interpolation of temperature of two adjacent thermocouple readings. This was not true for the heaters placed at two ends (Heater-1 and Heater-10), where the thermocouples were placed on the top of the heaters. The action network, as discussed in Sec 4.0 was implemented in LabVIEW and states were feed to the network to achieve control.

The sequence of activities in neuro-controller implementation are as follows: get the temperature readings from the setup, find error with respect to desired temperature profile, normalize the error to get states. Feed the states to the neuro controller. Get the control from neuro-controller. Implement the control to the heaters. It should be noted that, since our current setup did not have any cooling or sinks additional constraints were implemented in the above algorithm for implementation. Whenever there was an overshoot from the desired temperature profile or whenever the controller demanded cooling at any of the heater locations, the heater was shut off during that particular cycle time.

6.0 Results and discussion

The desired final profile in the current experiment was parabolic with the maximum desired temperature at the center of the setup. Fig.7 shows a plot of the dynamic temperature readings at 10 heater locations. The desired profile is plotted as "*" in the figure. The experiment was started from an arbitrary initial profile, where all the thermocouple readings are close to of 50^0 +/-5^0 C. A plot of the actual control that was directed by the neuro controller to achieve the desired profile is presented in Fig.8.

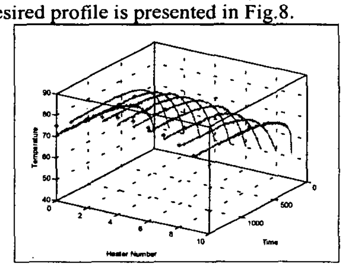

Fig. 7 Experimental results for desired parabolic profile

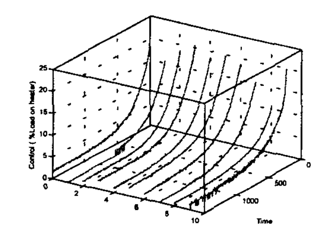

Fig. 8 Control (% Load of heaters)

It can be noted that the neuro-controller achieved the desired profile in the steady state. The results are within the limits of predicted physical system behavior.

Fig.9 and Fig.10 are the temperature and corresponding control plots of the experimentation, when it is was desired to drive the system to a uniform temperature of 75^0 C. The experiment was started with the initial temperature readings between 35^0 C and 40^0 C.

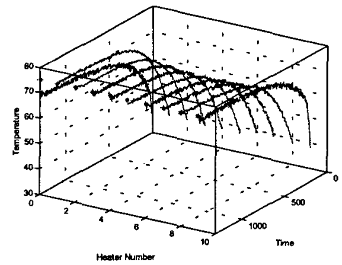

Fig. 9 Experimental results for desired linear profile

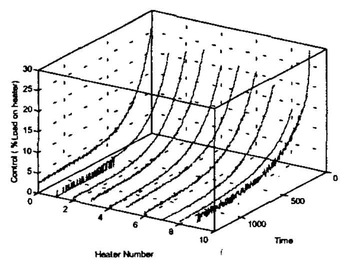

Fig. 10 Control (% Load of heaters)

Fig. 11 represents the experimental results when it was desired to achieve a parabolic temperature profile over the domain, with the difference in reading between two end nodes as 10^0 C. The corresponding control is shown in Fig.12.

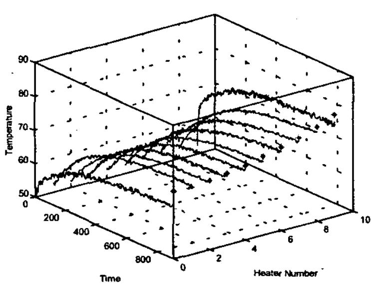

Fig. 11 Experimental results for desired parabolic profile

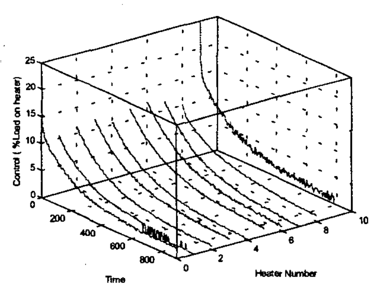

Fig. 12 Control (% Load of heaters)

From the experimental results it can be observed that the optimal neuro-controller can drive the experimental setup from any initial temperature to the desired profile.

7.0 Conclusions

An experimental setup representing the linear heat diffusion equation with Neumann boundary condition was built, its mathematical model was developed and finally the model was incorporated in an adaptive critic neuro-controller to develop an optimal controller. This controller was implemented online and the desired temperature profiles were successfully achieved in the experimental setup with this infinite time optimal regulator.

References

1. Padhi Radhakant, Balakrishnan S.N and Randolph Timothy W., *Adaptive-Critic Based Optimal Neuro Control Synthesis for Distributed Parameter Systems.* Automatica37 (2001) 1223-1234
2. Padhi R. and Balakrishnan S. N., *Infinite Time Optimal Neuro Control for Distributed Parameter Systems*, American Control Conference 2000, Proceedings of 2000, Vol. 6, 3778-3782
3. S. Matsumoto and M. Yoshida, *A method of design for a controller for a parabolic-type, distributed-parameter system. Application to one-dimensional thermal conduction.* International Chemical Engineering, Vol.29, No.1, pp 158-165, January 1989
4. Leden B., Hamza M.H., and Sheirah M.A., *Different Methods for estimation of thermal diffusivity of a heat diffusion process*, Identification and System Parameter Estimation, Edited by: Eykhoff P., Proc. 3rd IFAC symp., The Hague / Delft, The Netherlands, June 12-15, 1973, pp 639-648
5. Masatoshi Yoshida and Shigeru Matsumoto, *Controller design for parabolic distributed parameter system using finite integral transform techniques.* Journal of Process Control, Vol.6, No.6, 359-366,1996
6. Masatoshi Yoshida and Shigeru Matsumoto, *Control of axial temperature in a Packed-bed reactor*, Journal of Chemical Engineering of Japan, Vol 31, No 3, 381-390, 1998
7. Goodson R. Eugene and Polis Michael P., *Identification of Parameters in Distributed Systems, Distribute Parameter Systems Identification, Estimation, and Control*, Control and Systems Theory, Vol.6, Edited by: Ray W.Harmon and Lainiotis Demetrios G. pp 47-122.
8. Goodson R. Eugene and Polis Michael P., *Parameter Identification in Distributed Systems: A synthesis Overview*, Identification of Parameters in Distributed Systems, 1974 joint American Control Conference, Edited by: Goodson R. E. and Polis M. P.
9. Anderson Dale A., Tannehill John C., Pletcher Richard H., *Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corporation*, McGraw-Hill Book Company
10. Patankar Suhas V., *Numerical Heat Transfer and Fluid Flow*, Taylor and Francis Publishers
11. Robert D. Richtmyer and K. W. Morton, *Difference methods for initial-value problems.* New York, Interscience Publishers [1967]
12. Werbos P., *Neurocontrol and Supervised Learning: An Overview and Evaluation*, Handbook of Intelligent Control, Van Nostrand Reinhold, 1992, Editors: White D. A. and Sofge D.