JOURNAL OF THERMOPHYSICS AND HEAT TRANSFER
Vol. 17, No. 4, October–December 2003

Experimental Model Determination for Neurocontrol of a Thermal Conduction System

Prashant Prabhat,* D. C. Look,† and S. N. Balakrishnan‡
University of Missouri–Rolla, Rolla, Missouri 65409

Recently a synthesis methodology for the infinite time optimal neurocontrollers for partial differential equations systems in the framework of adaptive-critic design has been developed. The adaptive-critic approach is applied to a thermal conduction system. The experimental setup representing a one-dimensional heat conduction problem developed by matching the experimental and simulation results using a tridiagonal-matrix algorithm is presented. The discrete domain forms of the state, the costate and the optimal control equations are derived using the distributed parameter model of the system. The synthesis is introduced of an adaptive-critic-based infinite time optimal neurocontroller for online temperature profile control. The representative states of the system for training the networks was generated using Fourier series-based smooth states profile algorithm. Finally, the action network is implemented for temperature profile control of the experimental setup.

Nomenclature

J	=	cost function
L	=	length
M	=	number of grid points
N	=	number of time steps
Q	=	weight for state variable in the cost function
R	=	weight for control variable in the cost function
S_I	=	domain of interest, set of state profiles
$T(t, y)$	=	temperature within geometry
t	=	time
$u(t, y)$	=	control within geometry
X	=	dynamic function
y	=	spatial position
α	=	thermal diffusivity
β	=	source term distribution
λ	=	costate variable
ψ	=	utility function

Subscripts

c	=	cycle time indicator
D	=	discrete domain indicator
f	=	final spatial (time) indicator
i	=	finite difference spatial indicator
t	=	times of data writing indicator
0	=	beginning spatial (time) indicator

Superscript

k	=	finite difference time step indicator

Introduction

DISTRIBUTED systems are those which require description by partial differential equations (PDE) or other models that account explicitly for spatial variations of the dependent variables with time. Many chemical processes involving heat and/or mass transfer or chemical reactions are modeled by parabolic PDE. The dynamic behavior of most of the diffusion processes is well understood, and the parameters and boundary conditions are well known,[1] whereas, in the experimental setup herein, it was not only desired to synthesize and implement the optimal neurocontroller, but also to study the behavior of the system and then model its parameters. The first step for the experimental validation of the neurocontroller was the development of the experimental setup and parameter identification of its mathematical model, which would represent the heat diffusion (conduction) equation with Neumann boundary conditions. This paper appears to be the first attempt to treat parameter identification in distributed systems involved in approximating the system by a lumped model. The method assumes the system to be linear and develops a transfer function approximating the distributed model. Various methods for estimation of thermal diffusivity of a material have been compared in Ref. 2. In Ref. 2, the governing PDE is approximated by a set of finite difference equations and the dynamic (time series) data are used for parameter identification. The presentation here follows a similar approach, 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. Note that in the study presented, no attempt was made to determine the optimal number of sensor and actuator locations.

The design of controllers for PDE-based systems has been studied and implemented in Refs. 1, 3, and 4. The control of the diffusion process was approximated as a lumped parameter system with the technique of finite Fourier transform. Furthermore, in these studies, the control was designed by approximating the distributed parameter system by a lumped parameter system using an integral transform, the desire being to maintain a particular temperature profile in the steady state for the thermal system. In the study presented here, no use is made of any of these transforms. Instead the methodology of adaptive critic design[5] and dynamic programming methodology for optimal controller synthesis is used. The conventional dynamic programming methodology for the solution of optimal control, despite having many desirable features, is severely restricted by its computational requirements. However, recently, an advanced neurocontrol methodology called the adaptive-critic design has provided a new perspective. The advantages include optimal control of the plant maintaining a feedback structure to 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. This adaptive-critic design has recently been used for optimal control of distributed parameter systems,[6] where the comparison of the numerical results of the linear

Received 7 October 2002; revision received 5 May 2003; accepted for publication 15 May 2003.

*Graduate Student, Department of Mechanical and Aerospace Engineering and Engineering Mechanics.

†Professor Emeritus of Mechanical Engineering, Department of Mechanical and Aerospace Engineering and Engineering Mechanics. Associate Fellow AIAA.

‡Professor of Mechanical Engineering, Department of Mechanical and Aerospace Engineering and Engineering Mechanics. Associate Fellow AIAA.

heat diffusion problem with Neumann boundary conditions and the established operator theory based on a linear quadratic regulator[7] technique shows good results.

In this study, a linear one-dimensional, transient heat conduction equation with Neumann boundary conditions has been considered, and the optimal neurocontrollerhas been successfully implemented on the experimental setup for on-line temperature profile control.

Thus, the goal of this study is the formulation of a procedure for producing a given final temperature profile given an initial temperature profile, that is, develop values and functional forms for $\alpha(y)$ and $\beta(y)u(t, y)$ that will allow the initial temperature distribution to transform to another, predetermined temperature profile. To accomplish this task, the system characteristics must be determined. Therefore, this paper presents the first efforts to carry out this task.

The following discussion lays out the procedure. First, an experimental setup will be discussed. Then procedures used to determine $\alpha(y)$ and $\beta(y)u(t, y)$ for this setup will be developed. The final explantion will be based on control theory, that is, the procedure needed to produce the optimal procedure will be presented.

Experimental Setup and Its Modeling

In this section, the development of the experimental setup that would represent the one-dimensional, transient heat conduction problem is discussed. The main difficulty that was encountered related to the development of the mathematical model of the experimental setup. As opposed to most studies,[1,3,4] 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.

Analytical Background

Consider the one-dimensional, transient conduction problem as indicated by

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

with Neumann boundary conditions of

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

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

The initial condition is that

$$T(0, y) = \text{some given function of } y \tag{4}$$

Figure 1 shows an example initial condition of this type of problem, where it is desired to achieve a known temperature profile over the domain of interest, that is, $T(t \Rightarrow \infty, y)$. Note that $T(0, y)$ represents any initial profile within the domain of interest $y_0 \le y \le y_f$, $T(t, y)$ represents the transient temperature profile, and $T(t \Rightarrow \infty, y)$ is the desired final temperature profile. Here $\alpha(y)$ is the thermal diffusivity and $\beta(y)$ is the heat input (energy transferred into the domain of interest generated by means other than thermal conduction). In this discussion it will be referred to as the source term distribution. The control term is $u(t, y)$, and in this discussion it denotes a current load at which a heater is operating. Its value is 1 for a particular heater if it is full on (100% of its load) and 0 if it is off.

When the heater is partially on, u takes an appropriate value between 0 and 1. For example, if it is 50%, the value of u is 0.5 and so on. Note that, in the experimental setup, it was possible to set the values only to an integer value (in percentage). Accordingly, u can have only discrete values from the set $\{0, 0.01, \ldots, 1.0\}$. Other values were rounded to the nearest hundredth. The functional dependence of u on time t comes out of the solution of the optimal control problem (computed as a function of the state at each time step), and its functional dependence on the spatial variable y is governed by its physical (grid point) location.

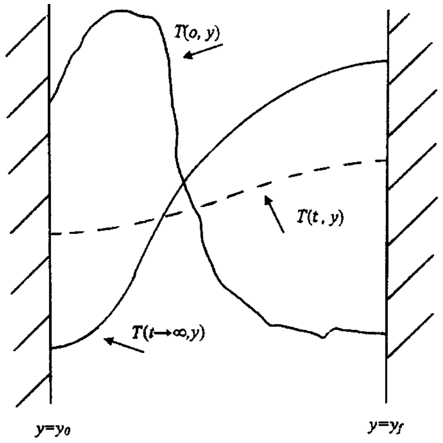

Fig. 1 Experiment schematic.

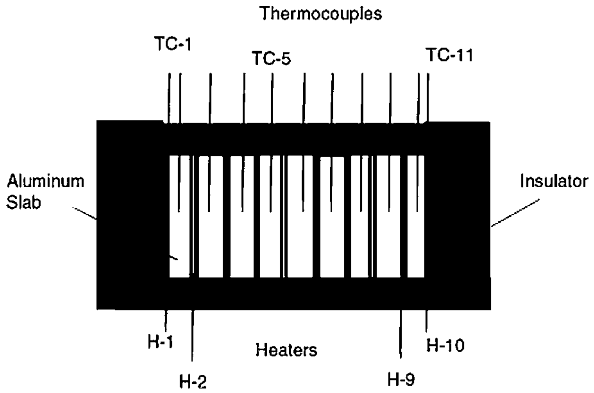

Fig. 2 Cross section (side view) of cylindrical experimental setup.

Experimental Setup

To present an overview of the hardware setup in this one-dimensional, transient conduction problem, note that the experimental setup consists of a series of aluminum slabs and heaters placed one after another, as shown in Fig. 2. There were 10 heaters and 9 aluminum slabs assembled. Mica heaters were selected for their small thickness of 0.000635 m (0.025 in.) per element and high temperature rating of 593°C. The heaters are 0.1524 m (6 in.) in diameter and have a lead bulge of 0.00508 m (0.2 in.) thick on one of their faces where the leading wires are connected to the heater. Aluminum slabs were used because of their high thermal conductivity. They are also 0.1524 m (6 in.) in diameter and 0.0127 m ($\frac{1}{2}$ in.) thick. A notch was cut on one of the faces of the aluminum slabs to accommodate the lead bulge of each heater. A hole of 0.003175 m (0.125 in.) in diameter was drilled radially into the side of each of the aluminum slabs to place a thermocouple into the slab. Because these holes facilitate temperature measurements, they were 0.0508 m (2 in.) deep and placed diametrically opposite to the notch to measure the temperature at a point farthest away from lead bulge of the two heaters assembled on either side of the slab. To minimize the effect of discontinuity in heat conduction on the temperature measurements due to presence of notch, two consecutive slabs were rotated by 90 deg. K-type thermocouples were used for the temperature measurement; their tips were placed at the bottom of a 0.0508-m-deep hole.

The electronic hardware schematic is given in Fig. 3. An external multiplexer device expands the number of analog input signals a plug-in data acquisition (DAQ) device can measure. To measure temperature, the two ends of the thermocouples were connected to the input channels of the multiplexer in the differential measurement mode. Each device channel of the DAQ scans four channels of the external multiplexer. The software used for input and output of data

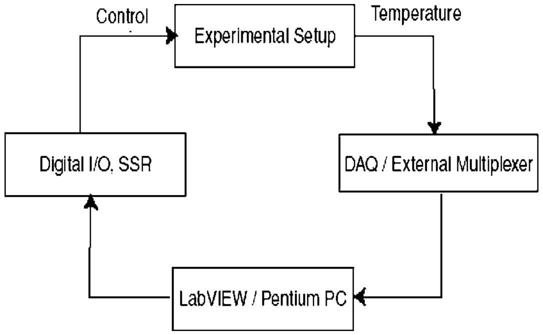

Fig. 3 Hardware setup.

was LabVIEW, installed on a Pentium personal computer. Channel 0 of the external multiplexer is used to get the cold junction compensation temperature reading. The temperature readings from the thermocouples were linearized in LabVIEW using the cold junction compensation temperature. The heaters were connected to an 120-V power supply through solid-state relays (SSR). These relays were switched on and off using a digital I/O. 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. (5)]:

$$N_C = 0.035N_t \tag{5}$$

where N_C is the cycle time in seconds and N_t is the number of times the data are written to the digital I/O during each cycle. For this model setup, N_C was 4.5 s.

During each cycle, the data were written to a digital I/O for a predetermined number of times after an interval 35 ms (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.

Modeling: Finite Difference and Tridiagonal Matrix Algorithm

A methodology for modeling this one-dimensional, transient heat conduction problem for a distributed parameter system was developed and will be presented in this section. While modeling this problem, the system parameters as well as an understanding of the system behavior is needed. Parameters are defined as functions or constants, other than the dependent variable, appearing explicitly in the mathematical model. Various approaches for identification of parameters for distributed systems have been discussed.[8,9] A finite difference scheme was used in this study. The tridiagonal matrix algorithm (TDMA)[10,11] was used for the numerical simulation. When the simulation results were matched with the experimental results, the mathematical model of the system was developed.

Model of $\alpha(y)$

Refer to Eq. (1). It is desired to determine the coefficients $\alpha(y)$ and $\beta(y)$ for this specific setup. This was achieved in two steps. Initially, experiments and simulations were conducted to arrive at a form of $\alpha(y)$. The control term $\beta(y)u(t, y)$ was neglected while arriving at a model of $\alpha(y)$. Thus, the solution to the homogeneous equation

$$\frac{\partial T(t, y)}{\partial t} = \alpha(y)\frac{\partial^2 T(t, y)}{\partial y^2} \tag{6}$$

must be determined first, where the initial and boundary conditions are the same as Eqs. (2–4).

The corresponding finite-difference mesh, using a backward finite difference approximation of Eq. (6), is

$$\frac{T_i^k - T_i^{k-1}}{\Delta t} = \alpha_i\left[\frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta y^2}\right] + \beta_i u_i^k \tag{7}$$

where k is the time t increment and i is the space y increment. Rearranging Eq. (7) yields

$$-\left[\alpha_i/\Delta y^2\right]T_{i+1}^k + \left[\left(2\alpha_i/\Delta y^2\right) + (1/\Delta t)\right]T_i^k - \left[\alpha_i/\Delta y^2\right]T_{i-1}^k = T_i^{k-1}/\Delta t \tag{8}$$

The solution to this set of n equations for this one-dimensional, transient situation can be obtained by the standard Gauss-elimination method. Because of the simple form of the equations, the elimination process turns into an algorithm called TDMA or Thomas algorithm (see Ref. 11). The standard form of Eq. (8) for TDMA application is

$$-A_i T_{i+1}^k + B_i T_i^k - C_i T_{i-1}^k = D_i \tag{9}$$

where

$$A_i = \alpha_i/\Delta y^2 \tag{10}$$

$$B_i = \left[\left(2\alpha_i/\Delta y^2\right) + (1/\Delta t)\right] \tag{11}$$

$$C_i = \alpha_i/\Delta y^2 \tag{12}$$

$$D_i = T_i^{k-1}/\Delta t \tag{13}$$

At this point, assume a relationship in the backward-substitution process in the following form is desired:

$$T_i^k = E_i T_{i+1}^k + F_i \tag{14}$$

In the case of $i > 0$, Eq. (14) can be reindexed to

$$T_{i-1}^k = E_{i-1}T_i^k + F_{i-1} \tag{15}$$

Substituting Eq. (15) in Eq. (9) leads to the suggested general form with the following coefficients:

$$E_i = \frac{A_i}{B_i - C_i E_{i-1}} \tag{16}$$

$$F_i = \frac{[D_i + C_i F_{i-1}]}{[B_i - C_i E_{i-1}]} \tag{17}$$

The insulated boundary conditions at the $y = y_0$ boundary [Eq. (2)] yields

$$T_{i+1}^k = T_i^k \tag{18}$$

Comparison of coefficients of Eq. (18) with that of Eq. (14) yields

$$E_1 = 1, \qquad F_1 = 0 \tag{19}$$

The insulated boundary condition at $y = y_f$ produces

$$T_{i+1}^k = T_i^k \tag{20}$$

Using Eq. (20) in Eq. (14), we get

$$T_{i_{\max}}^k = \left[F_{i_{\max}-1}/\left(1 - E_{i_{\max}-1}\right)\right] \tag{21}$$

At this point, all of the information is available to use the TDMA algotithm. However, note that the time step plays an important role in the convergence of the numerical solution. That is, for a very large value, numerical divergence might occur, whereas too small a value would lead to inconveniently slow convergence. For numerical stability of the finite difference schemes for a parabolic linear PDE of the form $(\partial T/\partial t) = \alpha(\partial^2 T/y^2)$, a well-known stability condition is $\Delta t/(\alpha\Delta y^2) < \frac{1}{2}$. Our selection of Δt satisfies this condition. Even though there is no lower bound from a stability requirement, if it is too small, it will lead toward problems in the neural network training (because the training is based on state and costate propagation). In other words, the corrections in the training will be very slow. We have not carried out any optimization process to fix a sufficiently higher value for Δt.

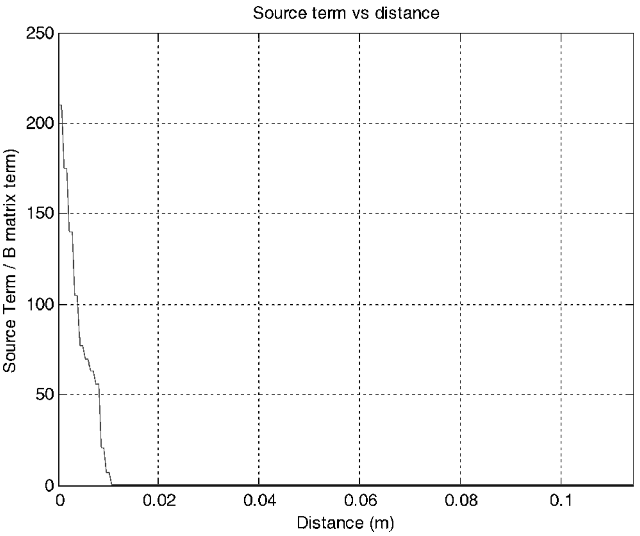

Fig. 4 Source term model for heater 1.

Model of $\beta(y)$

As mentioned earlier, the second step in modeling of this setup was to model the heaters represented by the source terms in Eq. (1). Thus, for modeling the control term/source term, Eq. (1) was used. Recall that $\beta(y)$ is a model of the heat generation term. As explained earlier, $\beta(y)$ is the source term distribution. It is a function of y (Fig. 4). The inclusion of this term requires changes in the finite difference scheme just presented, that is, the backward finite difference scheme yields

$$\frac{T_i^k - T_i^{k-1}}{\Delta t} = \alpha_i \left[\frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta y^2}\right] + \beta_i u_i^k \tag{22}$$

where k and i have the same interpretation. Rearranging Eq. (22) yields

$$-\left[\frac{\alpha_i}{\Delta y^2}\right]T_{i+1}^k + \left[\frac{2\alpha_i}{\Delta y^2} + \frac{1}{\Delta t}\right]T_i^k - \left[\frac{\alpha_i}{\Delta y^2}\right]T_{i-1}^k = \frac{T_i^{k-1}}{\Delta t} + \beta_i u_i^k \tag{23}$$

With use of the same procedure as before, the following coefficients are produced:

$$A_i = \alpha_i / \Delta y^2 \tag{24}$$

$$B_i = \left[\left(2\alpha_i / \Delta y^2\right) + (1/\Delta t)\right] \tag{25}$$

$$C_i = \alpha_i / \Delta y^2 \tag{26}$$

$$D_i = \left(T_i^{k-1} / \Delta t\right) + \beta_i u_i^k \tag{27}$$

Note that, the source/generation term is included in D_i. The rest of the TDMA is the same as presented earlier.

Experimentation and Numerical Simulations

Results of only one of the experiments that were conducted to model this setup is presented in this section. It is based on the selection of the parameters for 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.[10] To keep the size of roundoff errors down, it is sufficient[12] that

$$A_i > 0, \qquad B_i > 0, \qquad C_i > 0, \qquad B_i > A_i + C_i \tag{28}$$

These conditions are satisfied by the numerical scheme presented here. Also, because the coefficients A_i, B_i and C_i are always positive, the numerical results are stable. The number of grids that gave numerically consistent results was determined. It was observed that if the number of grid points selected was greater than or equal to 217 nodes, the numerical results were consistent with the experimental results. (This is then a grid-independent solution.)

For the selected temperature profile presented here, the parameters used were $\Delta t = 0.1$ s, $\Delta y = 5.267 \times 10^{-4}$ m, $i_{\min} = 1$, $i_{\max} = 217$, and $k_{\max} = 100$. Note that, with this selection of parameters, the TDMA was able to simulate the final results at the end of 10 s. To simulate the results after each 10-s 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 the simulation. Therefore, the TDMA is repeated many times to arrive at the simulated results for desired time.

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

This section describes the development of a model for heater 1 shown as H-1 in Fig. 2. To accomplish this, all of the other heaters were off; only heater 1 was kept on at 100% of its capacity. Thus, only its effect on the system behavior was recorded. Figure 5 shows the temperature response with this experimental setup for the 11 thermocouples under these conditions. Note that this profile requires the source term as per Eq. (1). With heater 1 was turned off, Fig. 6 shows the temperature response of the 11 thermocouples. Similarly, Fig. 6 results do not have a source term, that is, Eq. (5) is the governing equation.

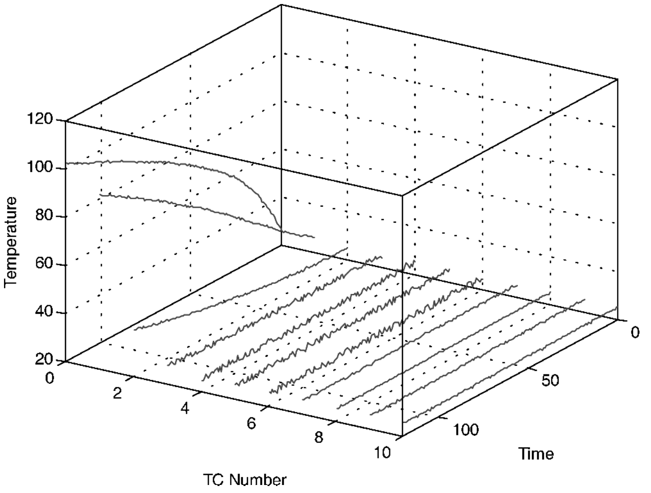

Fig. 5 Heating curve for heater 1.

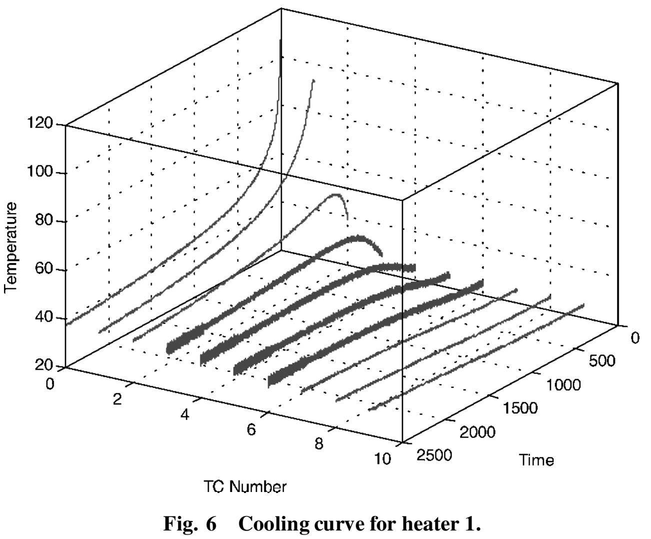

Fig. 6 Cooling curve for heater 1.

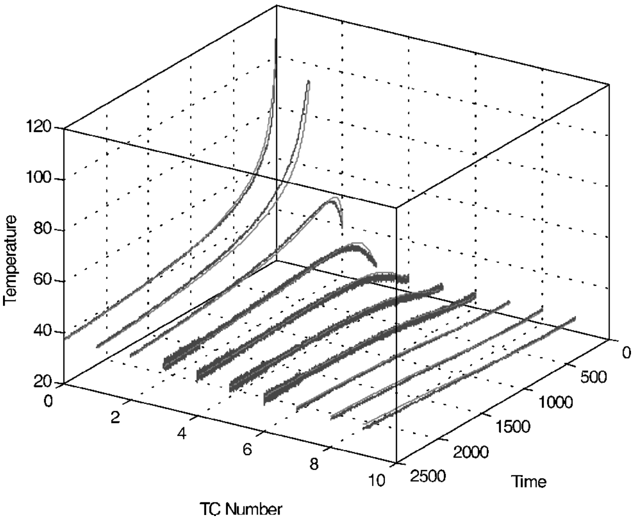

Fig. 7 Experimental results and TDMA simulations for cooling curve of heater 1.

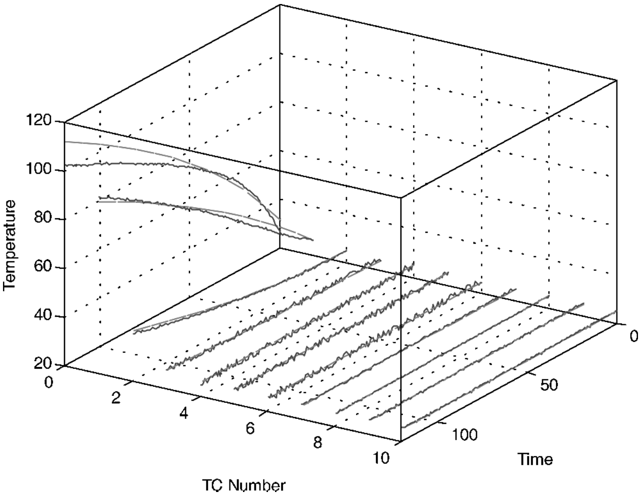

Fig. 8 Experimental results and TDMA simulations for heating curve of heater 1.

The task of determining a value for $\alpha(y)$ was based on the finite difference form of Eq. (6). The initial temperature profile was generated by linear interpolation between the 11 thermocouple readings at the initial time represented in Fig. 6. Thus, various constant values of $\alpha(y)$ were assumed, and the TDMA simulations were made to match the cooling curve of the 11 thermocouples in Fig. 6. Note in Fig. 7 that the chosen value of $\alpha = 8.5 \times 10^{-7}$ m^2/s leads to very good match between experimental results with that of TDMA simulations. Also, the assumption that α is constant over the spatial domain appears to be justified by these results.

Recall that the governing equation whose solution shows the system response in the presence of a source term is Eq. (1). Figure 5 shows such a system. Thus, modeling this heating curve should help to determine a source term model of $\beta(y)$. Recall that $u(t, y)$ must be 1 for heater 1: This means that heater 1 was switched on at 100% load capacity; $u(t, y)$ was zero for all of the other heaters. By trial and error, a distributed source model for $\beta(y)$ was developed for heater 1, which is shown as Fig. 4. For this choice of $\beta(y)$, the experimental results were in agreement with the simulation results for all 11 thermocouple results, as shown in Fig. 8.

The choice of the model for $\beta(y)$ shown in Fig. 4, can be justified in terms of the effect of the source term over the domain of interest. As in Fig. 5, the maximum effect of heater 1 is on the two sides adjacent to its physical location. In general, the effect of the heater decreases as we move farther away from the heater location. The meaning of effect here is the heat flux generated in the system, which leads to temperature rise. Note that even though the source term, which is a heater in the experimental setup, is located at node 1 (extreme left) in the system, it is represented by a distributed source term model as in Fig. 4. The justification for selection of a distributed model for the heater appears to be validated by the experimental and simulation results agreement. This procedure was repeated to find a distributed source term model of each heater.

Diffusion Conduction Optimal Control Problem

After development of the model of each component of the system, the next step was to synthesize the optimal neurocontroller. The first part of this section is a discussion of the methodology for determining the distributed parameter system as discussed in Ref. 6 and then proceed to develop the optimal diffusion problem controller. The vocabulary and terminology of the control specialist will be integrated into the discussion.

System Dynamics (State Equation)

The problem presented here is a one-dimensional, transient distributed parameter system; T is a function of the two independent variables time t and space y. The system dynamics (the temperture) evolves in time and is given by

$$x_{k+1,j} = f_k(T_{k,1}, T_{k,2}, \ldots, T_{k,M}, u_{k,j}) \tag{29}$$

The subscript k accounts for marching of time and j the spatial distribution. M denotes the final node number in that spatial distribution.

Cost Function

The following equation is referred to as a general cost function:

$$J = \sum_{k=1}^{N-1} \sum_{j=1}^{M} \Psi_{k,j}(T_{k,j}, u_{k,j}) \tag{30}$$

where N is the number of discrete time steps and Ψ is any utility function. In agreement with the preceding definition of the cost function, the cost function from time step k is

$$J_k = \sum_{\tilde{k}=k}^{N-1} \sum_{j=1}^{M} \Psi_{\tilde{k},j}(T_{\tilde{k},j}, u_{\tilde{k},j}) \tag{31}$$

Finally, the costate is defined as

$$\Delta y \lambda_{k,j} \equiv \frac{\partial J_k}{\partial T_{k,j}} \tag{32}$$

Optimal Control Equation

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

$$\frac{\partial J_k}{\partial u_{k,j}} = 0 \tag{33}$$

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

$$\sum_{\tilde{j}=1}^{M} \left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial u_{k,j}} \right) + \sum_{\tilde{j}=1}^{M} \lambda_{k+1,\tilde{j}} \left(\frac{\partial T_{k+1,\tilde{j}}}{\partial u_{k,j}} \right) \Delta y = 0 \tag{34}$$

Costate Dynamics

Substituting for J_k from Eq. (31) into Eq. (33) and some algebraic manipulation yields

$$\Delta y \lambda_{k,j} = \sum_{\tilde{j}=1}^{M} \left[\left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial x_{k,j}} \right) + \lambda_{k+1,\tilde{j}} \Delta y \left(\frac{\partial T_{k+1,\tilde{j}}}{\partial T_{k,j}} \right) \right] + \sum_{\hat{j}=1}^{M} \left[\sum_{\tilde{j}=1}^{M} \left\{ \left(\frac{\partial \Psi_{k,\tilde{j}}}{\partial u_{k,\hat{j}}} \right) + \lambda_{k+1,\tilde{j}} \Delta y \left(\frac{\partial T_{k+1,\tilde{j}}}{\partial u_{k,\hat{j}}} \right) \right\} \right] \left(\frac{\partial u_{k,\hat{j}}}{\partial T_{k,j}} \right) \tag{35}$$

The state equation, costate equation, and optimal control equation must be solved simultaneously to obtain the required optimal control. By the use of Eq. (34), Eq. (35) can be simplified to

$$\Delta y\lambda_{k,j} = \sum_{\bar{j}=1}^{M}\left[\left(\frac{\partial\Psi_{k,\bar{j}}}{\partial T_{k,j}}\right) + \lambda_{k+1,\bar{j}}\Delta y\left(\frac{\partial T_{k+1,\bar{j}}}{\partial T_{k,j}}\right)\right] \tag{36}$$

Note that this operation is valid along the optimal trajectory where the partial derivative of the cost function with respect to the control/source is zero.

Optimal Diffusion Problem

To develop a infinite time optimal controller, recall that the problem is described by Eqs. (1–4), where the state variable is $T(t, y)$ and $T(0, y)$ is any initial temperature profile within the interest domain. The objective is to find the optimal control $u(t, y)$ that minimizes the quadratic cost function

$$J = \frac{1}{2}\int_{t_0}^{\infty}\int_{y_0}^{y_f}[QT^2(t, y) + Ru^2(t, y)]\,\mathrm{d}y\,\mathrm{d}t \tag{37}$$

where $T(t, y)$ and $u(t, y)$ are the state and control variables as a function of time and spatial coordinate y, Q is the weighting factor on the state variable, and R is the weighting factor on the control variable. Furthermore, t_0 and $t_f \rightarrow \infty$ are initial and final times, where y_0 and y_f are initial and final points on the spatial coordinate axis.

Discrete Formulation

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

$$J = \frac{1}{2}\left[\sum_{k=1}^{\infty}\sum_{j=1}^{M}\left(Q_D T_{k,j}^2 + R_D u_{k,j}^2\right)\right] \tag{38}$$

where

$$Q_D = \Delta t\Delta y Q \tag{39}$$

$$R_D = \Delta t\Delta y R \tag{40}$$

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, the utility function is

$$\Psi_{k,j} = \tfrac{1}{2}\left[Q_D T_{k,j}^2 + R_D u_{k,j}^2\right] \tag{41}$$

The following equations are the resulting state, costate, and optimal control equations, respectively:

$$T_{k+1,j} = T_{k,j} + \Delta t\left[\alpha_j(T_{k,j+1} - 2T_{k,j} + T_{k,j-1})\big/\Delta y^2 + \beta_j u_{k,j}\right] \tag{42}$$

$$\lambda_{k,j} = \lambda_{k+1,j} + \Delta t\left[\alpha_j(\lambda_{k+1,j+1} - 2\lambda_{k+1,j} + \lambda_{k+1,j-1})\big/\Delta y^2 + Q_D T_{k,j}\right] \tag{43}$$

$$u_{k,j}^* = -R_D^{-1}\beta_j\lambda_{k+1,j} \tag{44}$$

where Δt and Δy are the step sizes of the time and spatial variables, respectively. These equations must be solved simultaneously for optimal control. Note that, together with the necessary conditions of optimality, we have to satisfy the following initial and boundary conditions.

If $T_{0,k}$ can be any point in the domain of interest, then $\lambda_{N,j} = 0$, as $N \rightarrow \infty$,

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

$$\lambda_{k,0} = \lambda_{k,1}, \qquad \lambda_{k,M+1} = \lambda_{k,M} \tag{45}$$

Adaptive-Critic Controller Synthesis

The adaptive-critic synthesis procedure has been presented.[6] However, the core of the technique, which is the iterative training between critic and action neural networks is presented briefly in Fig. 9. The philosophical justification for adaptive-critic structures has been discussed.[13] For this study, the solution processes are shown in Figs. 9–11.

The synthesized set of M critic networks, for $k = N - 1$, with input $T_{N-1,j}$ and output $\lambda_{N-1,j}$ is produced as per the following steps (Fig. 9):

1) Assume $T_{k,j}$ as some random smooth state profile. (The particulars will be discussed later.)

2) Obtain $u_{k,j}$ from the trained action networks.

3) Obtain $T_{k+1,j}$ from the state equation (42).

4) Input $T_{k+1,j}$ to the trained set of critic networks at $(k + 1)$th time step, to obtain $\lambda_{k+1,j}$.

5) Now, with the availability of $T_{k,j}$ and $\lambda_{k+1,j}$, calculate $\lambda_{k,j}$ from the costate equation (43),

6) Train the set of critic networks with input $T_{k,j-1}$, $T_{k,j}$, $T_{k,j+1}$ and output $\lambda_{k,j}$ for all of the networks related to the internal node points. For those intended for the boundary node points, we consider either $T_{k,1}$, $T_{k,2}$ or $T_{k,M-1}$, $T_{k,M}$ as the input.

After that, the focus is on action network synthesis. The training process is carried out in the following steps (Fig. 10).

1) Assume random $T_{k,j}$, within the relevant range, using smooth state profile algorithm and input it to the action networks to obtain $u_{k,j}$.

2) Use state equation (42) and the boundary condition (45) to obtain $T_{k+1,j}$ uniquely.

3) Input $T_{k+1,j}$ to the trained set of critic networks to obtain $\lambda_{k+1,j}$.

4) Get the optimal control $u_{k,j}^*$ from Eq. (44).

5) Train the networks at the kth time step with input $T_{k,j-1}$, $T_{k,j}$, $T_{k,j+1}$ and output $u_{k,j}^*$ for all of the networks related to the internal node points. For those intended for the boundary node points, we consider either $T_{k,1}$, $T_{k,2}$ or $T_{k,M-1}$, $T_{k,M}$ as the input.

Once this process of action synthesis is over, we revert to critic synthesis again. The alternate critic and action network training

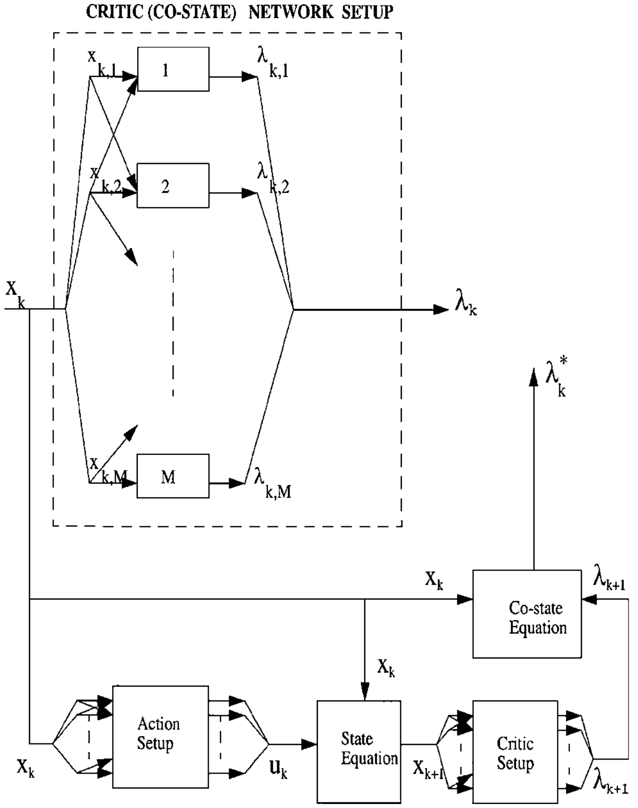

Fig. 9 Schematic of critic synthesis procedure.

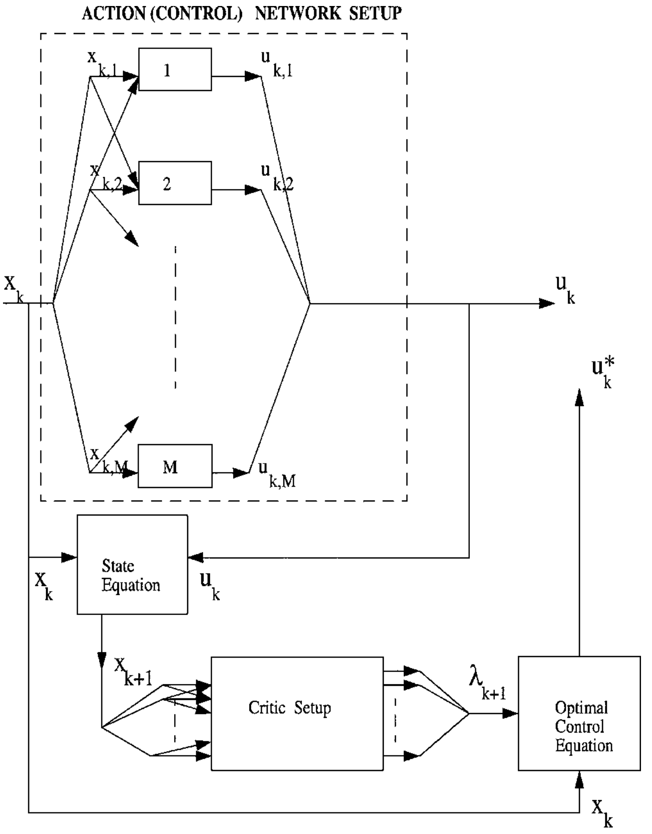

Fig. 10 Schematic of action synthesis procedure.

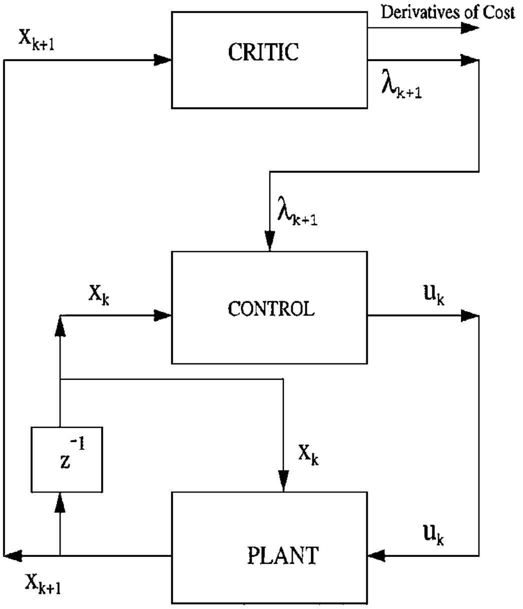

Fig. 11 Successive training of critic and action networks.

process is continued until no noticeable change in the output is observed in the outputs in the successive trainings (Fig. 11). Then the networks converge to give the true optimal relationships.

For the controller synthesis presented here, $Q_D = 1$, $R_D = 1$, $t_0 = 0$, $y_0 = 0$, and $y_f = 1.143 \times 10^{-1}$ m, which is the length of the experimental setup in Fig. 2. For the finite difference procedure for this system, we used $\Delta t = 0.01$ s and $\Delta y = 1.27 \times 10^{-2}$ m, which is the distance between two adjacent heater locations placed equidistant from each other. With this choice of parameters, stable training results were achieved. Because our experimental setup has 10 heaters, we select 10 node points for neural network synthesis. Thus, there are 10 action and critic networks, 1 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.

Note that the state and costate equations (42) and (43) need α_j and β_j values, respectively, from the experimental model of the system. Because 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/s 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. Recall that the TDMA used 217 nodes to model the system, whereas our current neurocontroller synthesis required an equivalent 10-node 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 using this numerical value as β_j for that node of neural network. Because the β term in the TDMA represents the rate of heat generation per unit volume, for the one-dimensional heat conduction, integration of this term over the length gives an equivalent representation of rate of the heat generation term to be incorporated in the neural network as a point source.

In the current implementation, the network structure is retained similar to that in Ref. 6. For the interior node points, we have used a multilayer feedforward network of the form $\pi_{3,5,5,1}$ for critic training and a similar network for action training. Here, $\pi_{3,5,5,1}$ is a neural network with three neurons in the input layer, five neurons each in the two other hidden layers, and one 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 the state values were not input at fictitious node points as input to the network. This was done because the associated boundary conditions lead to both $x_0 = x_1$ and $T_{M+1} = T_M$. This essentially means no additional information goes to the networks by providing the state values at the fictitious node points. The tangent sigmoid function was taken for all of the hidden layers and a linear function for the output layer.

Some of the important parameters to be considered in neural network training are stability and convergence. For the numerical results to be stable, the grid size for both time and space must be chosen properly. With the preceding choice of these parameters, the neural network training was achievable without any numerical instability. 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 neurocontroller is to be implemented, $T_{k,j}$, the state information, should be such that it represents a feasible state. If the choice of $T_{k,j}$ was made as random values selected over the domain, then the convergence could not be achieved for a 10-node architecture. The reason for this is that there is too much waviness in the training profile. Thus, an algorithm was developed to generate smooth state profiles for training the networks. When this algorithm is used, the $T_{k,j}$ profile has a smooth gradient over the entire length. From a temperature gradient perspective also, it was unlikely to encounter a steep temperature gradient in our current setup, which justified the implementation of smooth state profile for training the networks. Fourier series was used to generate the smooth state profiles. The equations involved in the generation of this have been discussed[14] and is repeated briefly here.

We assume an envelope profile to be representative of the smooth state profile as

$$f_{\text{env}}(y) = a + A\cos[-\pi + (2\pi y/L)] \tag{46}$$

where L is the overall length of the spatial domain y. A and a are the coefficients to be chosen. Then we define

$$S_I \equiv \left\{ \begin{matrix} T(t, y) : \|T(t, y)\| \le \|f_{\text{env}}(y)\|, \|T''(t, y)\| \le \|f''_{\text{env}}(y)\| \\ T'(t, 0) = T'(t, 0) = 0 \end{matrix} \right\} \tag{47}$$

as the domain of interest, where $T(t, y)$ is the state of the system to be generated for training the networks. After fixing $0 \le C_i \le 1$, we assume

$$\|x\|^2_{\max} = C_i \|f_{\text{env}}\|^2, \qquad \|x''\|^2_{\max} = \|f''_{\text{env}}\|^2 \tag{48}$$

To satisfy the boundary conditions, we assume a Fourier cosine series expansion as an approximation in the form of

$$T(t, y) = a_0 + \sum_{n=1}^{N_f} a_n \cos\left(\frac{n\pi y}{L}\right) \tag{49}$$

where N_f is the total number of terms considered to be necessary to generate the series and a_0 and a_n are the coefficients of the series.

After some algebra, we can write

$$\frac{L}{2}\left(2a_0^2 + \sum_{n=1}^{N_f} a_n^2\right) \leq C_i\left(a^2 + \frac{A^2}{2}\right)L \tag{50}$$

$$\frac{L}{2}\left(\sum_{n=1}^{N_f} n^4 a_n^2\right)\left(\frac{\pi}{L}\right)^4 \leq A^2\pi^4\left(\frac{2}{L}\right)^3 \tag{51}$$

To satisfy both the inequalities (50) and (51), random coefficients a_n, $n = 0, 1, \ldots, N_f$, are selected to generate a state profile using Eq. (49). Such a profile is guaranteed to lie within our definition of the domain of interest, that is, Eq. (47). The choice of the smooth states profile algorithm leads to proper training of the neural network structure, and the results were stable, which converged in the adaptive-critic training. Note that the controller synthesis was done entirely in off-line training by possibly using all representative states of the real system in the training algorithm. The on-line implementation of the neurocontroller was achieved by simply implementing the weights of the off-line-trained action networks, and no on-line training was required.

On-Line Implementation of the Neurocontroller

After the neurocontroller was synthesized, it was implemented in LabVIEW for on-line optimal control of the desired temperature profile, as shown in Fig. 12. To synthesize a regulator problem, the error in temperature at each node location with respect to the desired temperature is fed to the neurocontroller after normalization. The controller returns the desired control from the heaters. Because no sensors were available at exact heater locations, the temperature at the heater location was approximated by linear interpolation of the temperature of two adjacent thermocouple readings. This was not true for the heaters placed at two ends (heaters 1 and 10), where the thermocouples were placed on the outer face of the heaters. The action network was implemented in LabVIEW, and the states were fed to the network to achieve control.

The sequences of activities in the neurocontroller implementation are as follows:

1) Obtain the temperature readings from the setup.

2) Find the error with respect to desired temperature profile.

3) Normalize the error to obtain states.

4) Feed the states to the neurocontroller. Obtain the control from the neurocontroller.

5) Implement the control to the heaters by regulating their load capacity during the cycle time.

6) Repeat the procedure until the desired temperature profile is achieved.

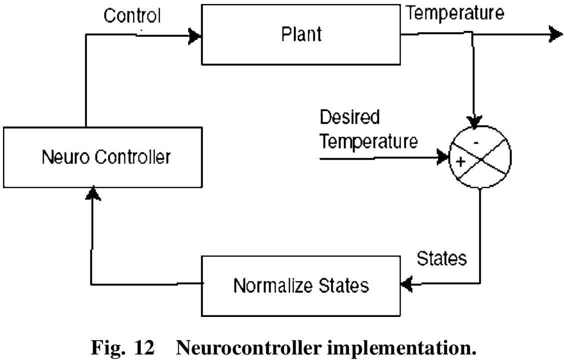

Fig. 12 Neurocontroller implementation.

Note that, because the experimental setup did not have cooling (energy sinks), additional constraints were implemented in the preceding algorithm. Whenever the controller demanded cooling at any of the heater locations, the heater was shut off during that particular cycle time. The temperature was sampled at 1000 samples/s from the first and second channel of the digital I/O. After sampling 80 readings from these thermocouples, they were averaged. This was done to minimize the error due to surrounding noise on the thermocouple readings. All of the temperature readings thus obtained were set using the calibration factor of each thermocouple. The calibration factor for each thermocouple was obtained by sampling temperatures from each thermocouple at room temperature and calibrating their mean temperature reading against a thermometer reading taken at the same time.

Results and Discussion

The experimental results of the neurocontroller implementation are discussed in this section. The desired final temperature profile in the current experiment was a parabolic profile with maximum desired temperature at the center of the setup. Figure 13 shows the dynamic temperature readings at 10 heater locations. The desired profile is plotted as asterisks in Fig. 13. The experiment was started from an arbitrary initial profile, where all of the thermocouple readings are close to 25°C, ±5°C. The actual control that was directed by the neurocontroller to achieve the desired profile is presented in Fig. 14.

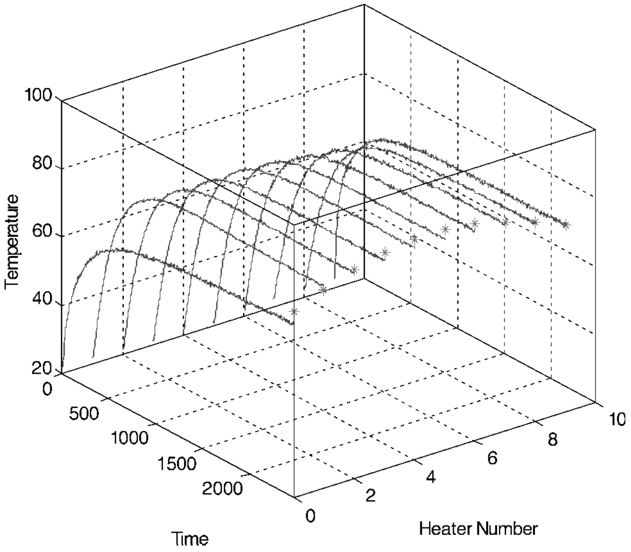

Fig. 13 Experimental results for desired parabolic profile.

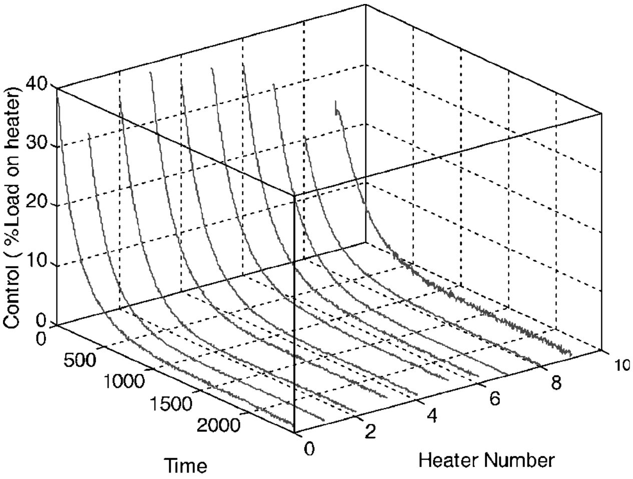

Fig. 14 Control (percent load of heaters) for Fig. 13.

Observe in the results that, by using the neurocontroller, it was possible to achieve the desired profile. Also, note that the neurocontroller achieved the desired profile in steady state (no change with time over a sufficiently long duration). Although some abnormality was observed in the rate of temperature buildup at two of the end nodes, it can be attributed to the physical setup configuration rather than to the neurocontroller. This observation is based on extensive simulations that were carried out for modeling the system, where the observed results are within the limits of the predicted physical system behavior. Some abnormality was also observed at the thermocouple placed on the face of heater 1. As in Fig. 13, even in the steady state, the temperature at node 0 does not reach the desired temperature, whereas all other thermocouples reach the desired temperature. This implies that even though the neurocontroller directs a finite nonzero control to be actuated by heater 0, this does not lead to any temperature rise at node 0.

To check if this abnormality was due to the existing noise in the thermocouple readings, a digital low-pass Bessel filter was implemented in LabVIEW after sampling the temperature readings. The sampling frequency of the Bessel filter was set to the sampling frequency at which data were acquired using the DAQ system. This low-cutoff frequency of the Bessel filter was set to 20 Hz. The low cutoff frequency was chosen by matching the temperature readings against the calibrated thermometer reading as discussed. Similar results as in Figs. 13 and 14 were achieved, even after implementing the Bessel filter. The possible explanation for the abnormality at the node-0 temperature reading in steady state might be attributed to the physical setup. Based on observations of the experimental data, it was observed that, even though the experimental setup is supposed to have perfect insulation on all sides, it is difficult to achieve. This means there is always some heat loss taking place from the system, which has been neglected in the controller synthesis because of its negligible value over a wide range of operating temperature. However, this heat loss from the system is not negligible compared to the heat flux input to the system in steady state when the control asymptotically goes to zero. This heat loss possibly not being dominant near the heater 0 location prevents the temperature rise in steady state. This means any heat flux input by heater 0 compensates for the heat loss due to poor insulation in that region. Similar abnormality in the results were observed in subsequent experiments discussed later.

In any thermal system, one of the factors affecting the temperature response is the boundary condition for the system. Although in the presented formulation the insulated boundary condition at the two ends of the system was taken into account, the heat loss from the longitudinal sides, due to radiative heat loss in the mathematical formulation, was not. However, this radiative heat loss is taken into account in the development of the distributed source term model for each heater. Because we model the system from experimental data, account was also taken for the heat loss from the system in our heater models. This means the heater models are specific to this insulated system. A change in the insulation of the system would require a remodeling of the source terms; hence, the insulation was not changed even after observing the abnormality in the node-0 reading.

Some other desired profiles that were achieved by the neurocontroller are presented in Figs. 15 and 16. Figures 15 and 16 are the temperature and corresponding control plots of experimentation, when it is was desired to drive the system to a uniform temperature of 50°C. The experiment was started with the initial temperature readings between 25 and 30°C.

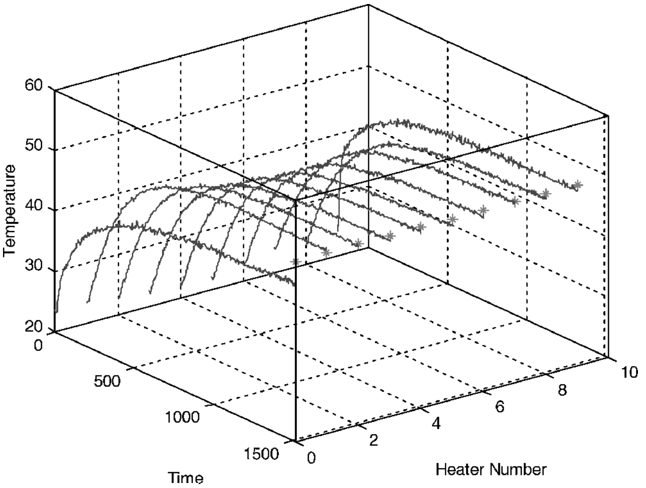

Fig. 15 Experimental results for desired linear profile.

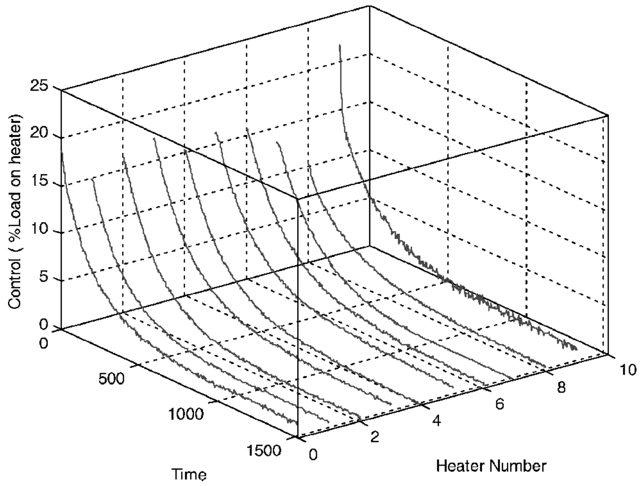

Fig. 16 Control (percent load of heaters) for Fig. 15.

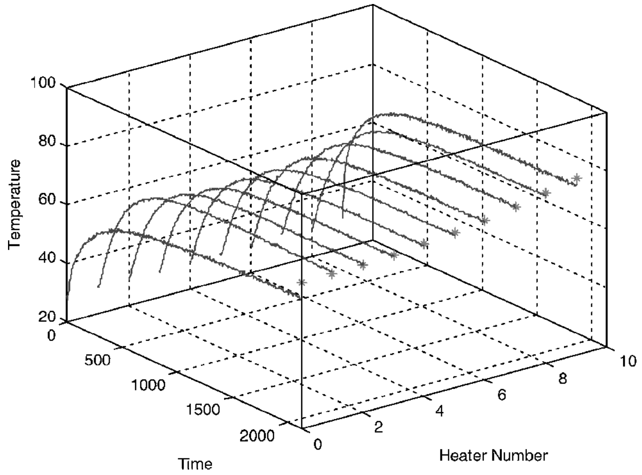

Fig. 17 Experimental results for desired parabolic profile.

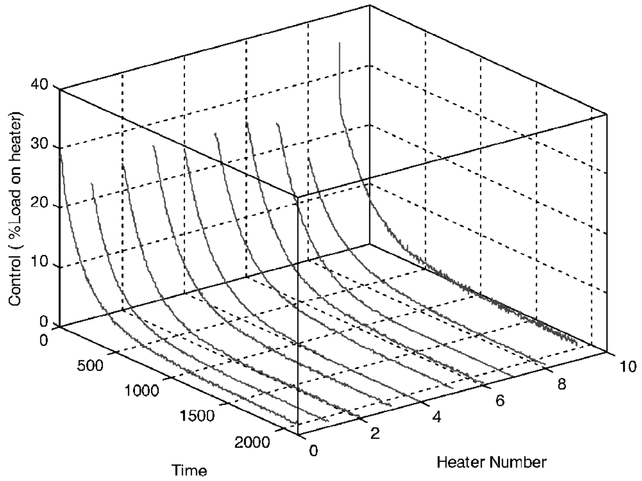

Fig. 18 Control (percent load of heaters) for Fig. 17.

The last set of experimental results is presented in Figs. 17 and 18. In this case it was desired to achieve a parabolic temperature profile, where the difference in reading the between two end nodes was 10°C. As observed in the experimental results discussed earlier, the neurocontroller is able to drive the system from any initial state to the desired temperature profile within the limits of expected behavior of the experimental setup.

The control of a distributed parameter system for a thermal system has wide application in chemical processing units,[3,4] where it is desired to maintain different zones of the distributed system at different temperatures. This presentation explores the possibility of the application of an optimal controller for such a distributed parameter system without using any model reduction techniques for the lumped parameter formulation.

Conclusions

An experimental setup representing the heat conduction equation with a Neumann boundary condition was built, its mathematical model was developed, and the model was incorporated into an adaptive-critic neurocontroller to develop an optimal controller. This controller was implemented on-line, and the desired transformation temperature profiles were successfully achieved in the experimental setup with this infinite time optimal regulator.

Acknowledgments

This research was supported by National Science Foundation Grant ECS 9976588, which the authors acknowledge with gratitude. The authors sincerely thank Radhakant Padhi, University of Missouri–Rolla, for his useful suggestions on neurocontrol formulation.

References

[1] Matsumoto, S., and Yoshida, M., "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, 1989, pp. 158–165.

[2] 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," *Proceedings of the 3rd IFAC Symposium*, edited by P. Eykhoff, Pergamon, New York, 1973, pp. 639–648.

[3] Yoshida, M., and Matsumoto, S., "Controller Design for Parabolic Distributed Parameter System Using Finite Integral Transform Techniques," *Journal of Process Control*, Vol. 6, No. 6, 1996, pp. 359–366.

[4] Yoshida, M., and Matsumoto, S., "Control of Axial Temperature in a Packed-Bed Reactor," *Journal of Chemical Engineering of Japan*, Vol. 31, No. 3, 1998, pp. 381–390.

[5] Padhi, R., and Balakrishnan, S. N., "Infinite Time Optimal Neuro Control for Distributed Parameter Systems," American Control Conf., Paper ACC00, 2000.

[6] Padhi, R., Balakrishnan, S. N., and Randolph, T. W, "Adaptive-Critic Based Optimal Neuro Control Synthesis for Distributed Parameter Systems," *Automatica*, Vol. 37, No. 7, 2001, pp. 1223–1234.

[7] Bryson, A. E., and Ho, Y. C., *Applies Optimal Control*, Blaisdell, Waltham, MA, 1969, Chap. 2.

[8] Goodson, R. E., and Polis, M. P., "Parameter Identification in Distributed Systems: A Synthesis Overview, Identification of Parameters in Distributed Systems," *American Control Conference: Identification of Parameters in Distributed Systems*, edited by R. E. Goodson and M. P. Polis, ASME Monograph, 1974, pp. 1–30.

[9] Goodson, R. E., and Polis, M. P., "Identification of Parameters in Distributed Systems, Distribute Parameter Systems Identification, Estimation, and Control," *Control and Systems Theory*, edited by W. H. Ray and D. G. Lainiotis, Vol. 6, Marcel Dekker, New York, 1974, pp. 47–122.

[10] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Pt. 1, Hemisphere, New York, 1984, Chaps. 3, 5, and 8.

[11] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980, Chaps. 1–5.

[12] Richtmyer, R. D., and Morton, K. W., *Difference Methods for Initial-Value Problems*, Interscience, New York, 1967, Chaps. 3 and 5.

[13] Werbos, P., *Neurocontrol and Supervised Learning: An Overview and Evaluation*, Handbook of Intelligent Control, edited by D. A. White and D. Sofge, Van Nostrand Reinhold, New York, 1992, Chaps. 3 and 13.

[14] Padhi, R., and Balakrishnan, S. N., "Proper Orthogonal Decomposition Based Optimal Control Design of Heat Equation with Discrete Actuators Using Neural Networks," American Control Conf., IEEE 2002-1545, 2002.