01 Jan 2007

Robust/Optimal Temperature Profile Control of a High-Speed Aerospace Vehicle using Neural Networks

Vivek Yadav

Radhakant Padhi

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

Recommended Citation

V. Yadav et al., "Robust/Optimal Temperature Profile Control of a High-Speed Aerospace Vehicle using Neural Networks," *IEEE Transactions on Neural Networks*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/TNN.2007.899229

Robust/Optimal Temperature Profile Control of a High-Speed Aerospace Vehicle Using Neural Networks

Vivek Yadav, Radhakant Padhi, and S. N. Balakrishnan

***Abstract*—An approximate dynamic programming (ADP)-based suboptimal neurocontroller to obtain desired temperature for a high-speed aerospace vehicle is synthesized in this paper. A 1-D distributed parameter model of a fin is developed from basic thermal physics principles. "Snapshot" solutions of the dynamics are generated with a simple dynamic inversion-based feedback controller. Empirical basis functions are designed using the "proper orthogonal decomposition" (POD) technique and the snapshot solutions. A *low-order* nonlinear lumped parameter system to characterize the infinite dimensional system is obtained by carrying out a Galerkin projection. An ADP-based neurocontroller with a dual heuristic programming (DHP) formulation is obtained with a single-network-adaptive-critic (SNAC) controller for this approximate nonlinear model. Actual control in the original domain is calculated with the same POD basis functions through a reverse mapping. Further contribution of this paper includes development of an online robust neurocontroller to account for unmodeled dynamics and parametric uncertainties inherent in such a complex dynamic system. A neural network (NN) weight update rule that guarantees boundedness of the weights and relaxes the need for persistence of excitation (PE) condition is presented. Simulation studies show that in a fairly extensive but compact domain, any desired temperature profile can be achieved starting from any initial temperature profile. Therefore, the ADP and NN-based controllers appear to have the potential to become controller synthesis tools for nonlinear distributed parameter systems.**

***Index Terms*—Control of distributed parameter systems, neural networks (NNs), proper orthogonal decomposition (POD), temperature control.**

I. INTRODUCTION

In a strictly mathematical sense, almost all real-world engineering problems are distributed in nature and can be described by a set of partial differential equations (PDEs). Even though for many practical problems (e.g., dynamics of car, airplane, etc.) a lumped parameter representation is often adequate, there are wide class of problems (e.g., heat transfer, fluid flow, flexible structures, etc.) for which one must take the spatial distribution into account. These systems are also known as distributed parameter systems (DPS). In this paper, an ADP-based neurocontrol of the temperature profile across the fin of a high-speed aerospace vehicle, modeled as a nonlinear distributed parameter system, is considered.

An interesting historical perspective of the control of distributed parameter systems can be found in [22]. There exist theoretical methods for the control of distributed parameter systems [10] in an infinite dimensional operator theory framework. One engineering approach to control the distributed parameter systems is to develop an approximate model of the system based on finite difference techniques and applying the control design tools directly on that approximated model [32]. Another technique is to have a finite dimensional approximation of the system using a set of orthogonal basis functions via Galerkin projection [17].

Galerkin projection normally results in high-order lumped system representations. For this reason, attention is being increasingly focused in recent literature on the technique of proper orthogonal decomposition (POD) [5], [6], [17], [35], [37]. In this technique, a set of problem-oriented orthogonal functions are designed to approximately span the solution space of the original system of PDEs. This is done through the so-called "snapshot" solutions, which are representative ensemble of the system states at arbitrary instants of time. In the process, a very low-order lumped model is created that is sufficient for practical controller design. For linear systems, it has been proved that such an approach leads to an optimal representation in the sense that it captures the maximum energy of the infinite dimensional system with the least number of basis functions [17], [35]. Even though theorems do not exist, this idea has been successfully used in controller design for nonlinear systems [5], [37].

An important open question in this area is the construction of proper input functions to collect representative snapshots. Quite often, an open-loop controller is used for this purpose. Recently, there have been some attempts at modifications to this technique [3], [36] since a snapshot-solution-based model may not "see" some modes that could be excited in a feedback situation. This drawback is eliminated in our study with the use of a feedback linearized controller [39] in generating snapshots.

The rest of this paper is organized as follows. Neural network (NN)-based controllers are reviewed in Section II. In

Manuscript received December 7, 2005; revised October 1, 2006; accepted February 5, 2007. This work was supported by the National Science Foundation under Grants ECS 0324428 and ECS 060170.

V. Yadav was with the Department of Mechanical and Aerospace Engineering, University of Missouri-Rolla, Rolla, MO 65409-0910 USA. He is now with the Ohio State University, Columbus, OH 43210 USA (e-mail: vyb5b@umr.edu).

R. Padhi is with the Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560012, India (e-mail: padhi@aero.iisc.ernet.in).

S. N. Balakrishnan is with the Department of Mechanical and Aerospace Engineering, University of Missouri-Rolla, Rolla, MO 65409-0910 USA (e-mail: bala@umr.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2007.899229

Section III, a nonlinear model for the fin of a high-speed aerospace vehicle that accounts for all of the three types of heat transfer, namely conduction, convection, and radiation, is developed. In Section IV, the controller design objectives are presented and the related problem formulation is given. In Section V, a POD-based basis function design and its subsequent use in a Galerkin projection scheme are discussed. In Section VI, the dual heuristic programming (DHP)-based NN synthesis procedure to design the optimal controller is described. Furthermore, the single-network-adaptive-critic (SNAC) algorithm is presented. In Section VII, an online NN controller development to provide robustness against uncertainties is provided. In Section VIII, the numerical results are analyzed. Conclusions are drawn in Section IX. Proofs of the hypothesis used are presented in the Appendix.

II. Background

There has also been a lot of interest in the use of NNs for controller design that guarantees desired performance in the presence of uncertainties and unmodeled dynamics. A multistage NN robot controller with guaranteed tracking performance was proposed by [24]. This controller was designed specifically for a serial link robot arm and was developed by using a filtered error/passivity approach. Bounded tracking errors and bounded NN weights were guaranteed. In [31], many architectures of adaptive controllers using recurrent networks were presented. A robust adaptive output feedback controller for single-input–single-output (SISO) systems with bounded disturbance was studied by [2]. In [8], theoretical development and numerical investigation of an adaptive tracking controller using NNs were presented. They provided stable weight adjustment rules for an online NN and simulation results for an F-18 aircraft model. In [23], an online adaptive NN for use in a nonlinear helicopter flight controller was designed. The network helped the system with good tracking capabilities in the face of significant modeling errors. An adaptive output feedback control scheme for uncertain systems using NNs was proposed in [16]. In [24] and [25], the authors discussed an online NN that approximates unknown functions and is used in controlling the plant. A robust adaptive control methodology that uses single hidden-layer feedforward NNs was presented in [27]. In [18], a Lyapunov-equation-based theory for robust stability of systems in the presence of uncertainties was developed. The result is an "extra control" which when added to the basic control effort kept the system stable. This approach was illustrated through a helicopter problem. In this paper, we develop a version of extra control to account for the unmodeled dynamics and parametric uncertainties.

The method of dynamic programming [7], [25] is a powerful tool to solve many real-life problems. It produces a comprehensive solution by generating a family of optimal paths, or what is known as the "*field of extremals*." A major drawback of the dynamic programming approach is that it requires a prohibitive amount of computation and storage, and therefore, is impractical to use.

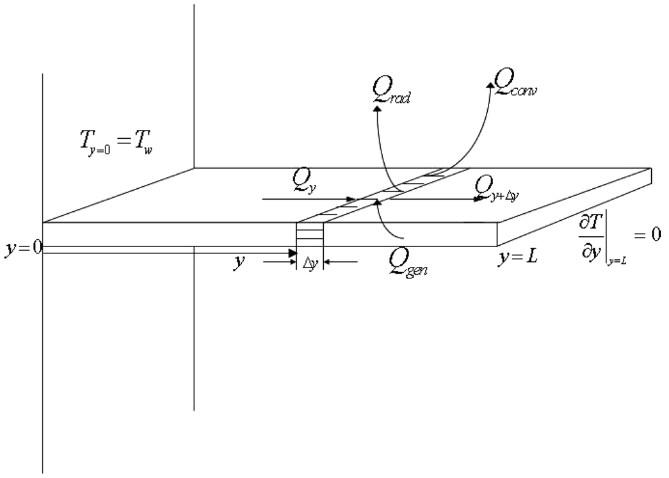

Fig. 1. Pictorial representation of the problem.

However, an *approximate dynamic programming* approach to circumvent the computational load with an *adaptive critic* neurocontroller synthesis has been proposed in [4], [34], [42], and [43]. The adaptive critic methodology approximates and optimizes a control law iteratively during the offline training of "action" and "critic" networks, for an entire envelope of states. The action networks capture the relationship between the state and control variables, whereas the critic network captures the relationship between state and costate variables in the development of optimal control theory. There are many variations of this technique in the literature [34]. Among many successful uses of this method for nonlinear control design are [4], [12], and [41]. Issues of convergence and stability of adaptive critic methods have been addressed in [26].

III. Mathematical Model for the Problem

Mathematical model of heat transfer in a cooling fin of a high-speed aerospace vehicle traversing through the earth's atmosphere is developed in this section using concepts from basic thermal physics [28]. The development is illustrated in Fig. 1.

The law of conservation of energy in the infinitesimal volume at a distance y, having length Δy (as depicted in Fig. 1), for this problem, is described in

$$Q_y + Q_{\text{gen}} = Q_{y+\Delta y} + Q_{\text{conv}} + Q_{\text{rad}} + Q_{\text{chg}} \tag{3.1}$$

where $Q_y = -kA(\partial T/\partial y)$ is the entering rate of heat conduction, $Q_{\text{gen}} = S(T,y)A\Delta y$ is the rate of heat generation, and $S(T,y)$ is the rate of heat generation per unit volume (also a function of both time t and spatial location y) and acts as the control variable for this problem. Note that the controller has been assumed to be continuous in the spatial domain. $Q_{y+\Delta y} \triangleq$ is the exiting rate of heat conduction, $Q_{\text{conv}} = hP\Delta y(T - T_{\infty_1})$ is the rate of heat convection, $Q_{\text{rad}} = \varepsilon\sigma P\Delta y(T^4 - T_{\infty_2}^4)$ is the rate of heat radiation, and $Q_{\text{chg}} = \rho CA\Delta y(\partial T/\partial t)$ is the rate of heat change [28].

A first-order Taylor series expansion is used to approximate the exiting rate of heat conduction as

$$Q_{y+\Delta y} \approx Q_y + \left(\frac{\partial Q_y}{\partial y}\right)\Delta y \tag{3.2}$$

TABLE I
PARAMETER DEFINITIONS AND NUMERICAL VALUES

Parameter	Definition	Numerical value
k	Thermal conductivity	19 $W/(m^oC)$
A	Cross sectional area	2' × 3"
P	Perimeter	4'6"
h	Convective heat transfer coefficient	20 $W/(m°C)$
T_{∞_1}	Temperature of the medium in the immediate surrounding of the surface	100 °C
T_{∞_2}	Temperature at a far away place in the direction normal to the surface	-40 °C
$\in$	Emissivity of the material	0.965
σ	Stefan-Boltzmann constant	5.669×10^{-8} W/m^2K^4
ρ	Density of the material	7865 kg/m^3
C	Specific heat of the material	0.46 $kJ/(kg°C)$

$T(t,y)$ represents the temperature, which varies with both time t and spatial location y. Definitions of the various parameters and the numerical values used in this paper are given in Table I.

Area A and perimeter P are computed assuming the following fin dimensions: $10'\times2'\times3''$. Note that a 1-D approximation for the dynamics is used. In the paper, this means a uniform temperature in the other two dimensions being arrived at instantaneously.

By using the expressions for individual terms and defining $\alpha_1 \triangleq (k/\rho C)$, $\alpha_2 \triangleq -(Ph)/(A\rho C)$, $\alpha_3 \triangleq -(P\varepsilon\sigma)/(A\rho C)$, and $\beta \triangleq 1/(\rho C)$, the PDE representation of conservation of energy in (3.1) becomes

$$\frac{\partial T}{\partial t}=\alpha_1\left(\frac{\partial^2 T}{\partial y^2}\right)+\alpha_2\left(T-T_{\infty_1}\right)+\alpha_3\left(T^4-T_{\infty_2}^4\right)+\beta S(T,y). \tag{3.3}$$

Boundary conditions for (3.3) are

$$\left.\frac{\partial T}{\partial y}\right|_{y=0}=c(t) \quad \left.\frac{\partial T}{\partial y}\right|_{y=L}=0 \tag{3.4}$$

where the value of $c(t)$ is dictated by the temperature profile $T(t,y)$ at $y=0$. An insulated boundary condition at the tip is assumed.

IV. CONTROLLER OBJECTIVES

Design objectives and preliminary steps in a temperature feedback controller development for a high-speed aerospace vehicle are presented in this section.

The main objective of the controller is to make the system reach a desired temperature profile on the fin, $T(y)\rightarrow T_d(y)$ as time $t\rightarrow\infty$, where $T_d(y)$ is the desired temperature distribution along the fin. The controller design is carried out such that $T_d(y)$ acts as a steady-state condition after $T(y)\rightarrow T_d(y)$. Temperature is normalized between 0 and 1 to help with roundoff errors. The normalized temperature X is defined as $X(y)=(T(y)-T_{\infty_2})/(T_{\max}-T_{\infty_2})$, where $T_{\max}$ is the maximum safe operating temperature of the fin (= 1100 K) and the normalized time, τ is defined as $\tau=t/F$, where $F(=1000)$. Let $X_d(y)$ be the nondimensional temperature corresponding to $T_d(y)$. Defining the deviation (error) $x(y)=X(y)-X_d(y)$ and $S_1=\beta FS/\Delta T$ and substituting in (3.3), the error dynamics become

$$\begin{aligned}\frac{\partial x}{\partial\tau}=&A_0\left(\frac{\partial^2 x}{\partial y^2}\right)+\mathrm{A}_1(X_d)x+\mathrm{A}_2(X_d)x^2+\mathrm{A}_3(X_d)x^3\\&+\mathrm{A}_4(X_d)x^4\\&+\left[\alpha_1F\left(\frac{\partial^2 X_d}{\partial y^2}\right)+\frac{\alpha_2F}{\Delta T}\left(\Delta TX_d+T_{\infty2}-T_{\infty_1}\right)\right.\\&\left.+\frac{\alpha_3F}{\Delta T}\left((\Delta TX_d+T_{\infty2})^4-T_{\infty_2}^4\right)\right]+S_1\end{aligned} \tag{4.1}$$

where

$$\begin{aligned}\mathrm{A}_0&=\alpha_1F\\\mathrm{A}_1(X_d)&=F\left(\alpha_2+4\alpha_3\left(X_d\Delta T+T_{\infty_2}\right)^3\right)\\\mathrm{A}_2(X_d)&=6\alpha_3F\Delta T\left(X_d\Delta T+T_{\infty_2}\right)^2\\\mathrm{A}_3(X_d)&=4\alpha_3F\Delta T^2\left(X_d\Delta T+T_{\infty_2}\right)\\\mathrm{A}_4&=\alpha_3F\Delta T^3\\\Delta T&=T_{\max}-T_{\infty_2}.\end{aligned} \tag{4.2}$$

Note that the coefficients of nonlinear terms (4.1) and (4.2) are functions of the desired temperature profile. The controller development now becomes a regulator problem. The steady-state control solution S_1^* is obtained by substituting X_d in place of X in (3.3) and imposing the steady-state condition

$$\begin{aligned}S^*(y)=-\frac{1}{\beta}&\left[\alpha_1\Delta T\left(\frac{\partial^2 X_d}{\partial y^2}\right)+\alpha_2\left(\Delta TX_d+T_{\infty_2}-T_{\infty_1}\right)\right.\\&\left.+\alpha_3\left((\Delta TX_d+T_{\infty_2})^4-T_{\infty_2}^4\right)\right].\end{aligned} \tag{4.3}$$

The S_1^* in (4.3) acts as a feedforward controller for this problem. A feedback controller (see Section IV-A) is added to this control to yield the total control.

A. Feedback Controller

The actual temperature and control variables are written in terms of the desired final (steady)-state temperature profile $X_d(y)$ and the control $S_1^*(y)$

$$\begin{aligned}X(t,y)&=X_d(y)+x(t,y)\\S_1(t,y)&=S_1^*(y)+u(t,y).\end{aligned} \tag{4.4}$$

In (4.4), $x(t,y)$ and $u(t,y)$ are the deviations from their respective steady-state values. Next, the deviation dynamics are developed by substituting (4.4) in (4.1) and simplifying the resulting equation. Consequently, we get

$$\frac{\partial x}{\partial t} = A_0\left(\frac{\partial^2 x}{\partial y^2}\right) + F(x, X_d) + u \tag{4.5}$$

where $\mathrm{F}(x, X_d) \triangleq (\mathrm{A}_1(X_d)x + \mathrm{A}_2(X_d)x^2 + \mathrm{A}_3(X_d)x^3 + \mathrm{A}_4(X_d)x^4)$.

Similarly, the boundary conditions are expressed in terms of the deviations from the steady-state values as

$$\left.\frac{\partial x}{\partial y}\right|_{y=0} = \frac{c(t)}{\Delta T} - \left.\frac{\partial X_d}{\partial y}\right|_{y=0} \qquad \left.\frac{\partial x}{\partial y}\right|_{y=L} = -\left.\frac{\partial X_d}{\partial y}\right|_{y=L}. \tag{4.6}$$

The purpose of the feedback controller $u(t,y)$ is to make the deviations from the steady-state conditions, i.e., $x(t,y) \to 0$ as time $t \to \infty$. As the state deviations tend to zero with time, the associated control effort goes to zero. These goals are achieved by minimizing a quadratic cost function in

$$J = \frac{1}{2}\int_0^\infty \int_0^L (qx^2 + ru^2)\,dy\,dt. \tag{4.7}$$

In (4.7), $q \geq 0$ and $r > 0$ are the weights that express the designer's concern for excessive deviations from the nominal and the control effort, respectively. Equations (4.5)–(4.7) define the complete optimal control problem.

V. Low-Order Lumped Parameter Approximation

This section discusses the development of low-order finite dimensional models for controller synthesis.

A. Procedure to Generate Snapshot Solutions

Snapshot solutions are generated by starting the solution process from different initial conditions that satisfy the boundary conditions. In order to generate snapshot solutions, the simulation is carried out for a fixed amount of time and snapshots are selected at equally spaced instants on this trajectory. This process is repeated for different initial conditions. Rather than using chosen input functions, a feedback-linearization-based control [39] is used to simulate the closed-loop behavior of the system.

B. POD: A Brief Review

The POD is a technique for determining an optimal set of basis functions, with a set of snapshot solutions obtained by the method described in Section V-A.

Let $\{x_i(y) : 1 \leq i \leq N, y \in \Omega\}$ be a set of N *snapshot* solutions (observations) of a physical process over the domain Ω at arbitrary instants of time. Let the dimension of $x_i(y)$ be $n \times 1$. The goal of the POD technique is to find a set of basis functions φ such that I is maximized in

$$I = \frac{1}{N}\sum_{i=1}^{N} \left|\langle x_{i,n\times 1}, \varphi_{n\times 1}\rangle\right|^2 / \langle \varphi_{n\times 1}, \varphi_{n\times 1}\rangle. \tag{5.1}$$

As a standard notation, the L^2 inner product is defined as $\langle \varphi_{n\times 1}, \psi_{n\times 1}\rangle = \int_\Omega \varphi_{n\times 1}, \psi'_{n\times 1} dy$. It has been shown [38] that when the number of degrees of freedom required to describe x_i is larger than the number of snapshots N (always true for infinite dimensional systems), it is sufficient to express the basic functions as linear combinations of the snapshots as

$$\varphi_{n\times 1} = \sum_{i=1}^{N} w_i x_{i,n\times 1}. \tag{5.2}$$

Here, the coefficients w_i are to be determined such that φ maximizes (5.1). Then, this approach consists of the following steps.

- Construct an eigenvalue problem

$$C_{N\times N} W_{N\times n} = \lambda W_{N\times n}$$

where

$$C = [c_{ij}]_{N\times N}, \qquad c_{ij} = \frac{1}{N}\int_\Omega x_i(y)x_j(y)dy. \tag{5.3}$$

- Obtain N eigenvalues and corresponding eigenvectors of the C matrix. Note that matrix C is symmetric, and hence, its eigenvalues are real. Also, it has been shown that all eigenvalues of C are nonnegative [35].
- Sort the eigenvalues of C in a descending order $\chi_1 \geq \chi_2 \geq \cdots \geq \chi_N \geq 0$. Let the corresponding eigenvectors be $W^1 = [w_1^1 \cdots w_N^1]^T, W^2 = [w_1^2 \cdots w_N^2]^T, \cdots, W^N = [w_1^N \cdots w_N^N]^T$. A property of these eigenvectors is that they are mutually orthogonal.
- Normalize the eigenvectors to satisfy

$$\langle W^l, W^l\rangle = (W^l)^T W^l = \frac{1}{N\lambda_l}. \tag{5.4}$$

This will ensure that the POD basis functions are orthonormal.
- Cut off the eigenspectrum judiciously, so that the truncated system with $\tilde{N} \leq N$ eigenvalues will satisfy $\sum_{j=1}^{\tilde{N}} \lambda_j \approx \sum_{j=1}^{N} \lambda_j$. Usually, it turns out that $\tilde{N} \ll N$.
- Finally, construct the $\tilde{N}$ basis functions as

$$\varphi_1(y) = \sum_{i=1}^{N} w_i^1 x_i(y)$$
$$\vdots$$
$$\varphi_{\tilde{N}}(y) = \sum_{i=1}^{N} w_i^{\tilde{N}} x_i(y). \tag{5.5}$$

An interested reader can refer to [17], [35], and [36] for details about this procedure.

C. Lumped Parameter Problem

The reduction of the infinite dimensional PDE-driven problem to a finite set of ordinary differential equations and a related cost function are explained in this section. After obtaining the basis functions $\varphi(y)$, $x(t,y)$ and $u(t,y)$ are expanded as follows:

$$x(t,y) \approx \sum_{j=1}^{\tilde{N}} \hat{x}_j(t)\varphi_j(y)$$
$$u(t,y) \approx \sum_{j=1}^{\tilde{N}} \hat{u}_j(t)\varphi_j(y). \tag{5.6}$$

One can notice that both $x(t, y)$ and $u(t, y)$ are characterized by the same basis functions. This implies that a state feedback controller spans a subspace of the state variables, and hence, the basis functions for the state are assumed to be capable of spanning the controller as well. By substituting these expansions of state and controller variables in (4.5), we get

$$\sum_{j=1}^{\tilde{N}} \dot{\hat{x}}_j \varphi_j = A_0 \sum_{j=1}^{\tilde{N}} \hat{x}_j \varphi_j'' + F\left(\sum_{j=1}^{\tilde{N}} x_j \phi_j\right) + \sum_{j=1}^{\tilde{N}} \hat{u}_j \varphi_j. \quad (5.7)$$

Note that $F(\sum_{j=1}^{\tilde{N}} x_j \phi_j)$ has a linear term and other no-linear terms, hence $F(\sum_{j=1}^{\tilde{N}} x_j \phi_j)$ can be written as $F(\sum_{j=1}^{\tilde{N}} x_j \phi_j, X_d) = A_1(X_d) \sum_{j=1}^{\tilde{N}} x_j \phi_j + f(\sum_{j=1}^{\tilde{N}} x_j \phi_j, X_d)$, where $f(\sum_{j=1}^{\tilde{N}} x_j \phi_j, X_d)$ is defined as $A_2(X_d)(\sum_{j=1}^{\tilde{N}} x_j \phi_j)^2 + A_3(X_d)(\sum_{j=1}^{\tilde{N}} x_j \phi_j)^3 + A_4(X_d) (\sum_{j=1}^{\tilde{N}} x_j \phi_j)^4$. Next, taking the Galerkin projection of (5.7) on the basis function $\varphi_i(.)$ (i.e., taking the inner product with respect to φ_i), and using the fact that the basis functions are orthonormal, yields

$$\dot{\hat{x}}_i = A_0 \sum_{j=1}^{\tilde{N}} \left\langle \varphi_i, \varphi_j'' \right\rangle \hat{x}_j + A_1(X_d)\hat{x}_i + \langle f(x), \varphi_i \rangle + \hat{u}_i. \quad (5.8)$$

Repeating this exercise for $i = 1, \ldots, \tilde{N}$ and arranging the equations in order, leads to a set of ordinary differential equations of the form

$$\dot{\hat{X}} = A\hat{X} + \hat{f}(\hat{X}) + B\hat{U} \quad (5.9)$$

where $\hat{X} \triangleq [\hat{x}_1, \ldots, \hat{x}_{\tilde{N}}]^T$, and $\hat{U} \triangleq [\hat{u}_1, \ldots, \hat{u}_{\tilde{N}}]^T$. Other symbols are defined as follows:

$$\begin{aligned} A &\triangleq A_0[a_{ij}] + A_1(X_d) I_{\tilde{N}} \\ a_{ij} &\triangleq \left\langle \varphi_i, \varphi_j'' \right\rangle = \int_0^L \varphi_i \varphi_j'' dy \\ &= \left[\varphi_i \varphi_j'\right]\Big|_{y=0}^{y=L} - \int_0^L \varphi_i' \varphi_j' dy \\ \hat{f}_i(\hat{X}) &\triangleq \langle f(x), \varphi_i \rangle = \int_0^L f(x) \varphi_i dy \\ B &\triangleq I_{\tilde{N}}. \end{aligned} \quad (5.10)$$

Next, the terms in the cost function (4.7) that contain x and u should be written in terms of $\hat{x}$ and $\hat{u}$

$$\begin{aligned} \int_0^L q x^2 dy &= q\langle x, x \rangle = q \left\langle \left(\sum_{i=1}^{\tilde{N}} \hat{x}_i \varphi_i\right), \left(\sum_{j=1}^{\tilde{N}} \hat{x}_j \varphi_j\right) \right\rangle \\ &= q \sum_{j=1}^{\tilde{N}} \hat{x}_j \hat{x}_j = \hat{X}^T Q \hat{X} \end{aligned} \quad (5.11)$$

where $Q = qI_{\tilde{N}}$ and

$$\begin{aligned} \int_0^L r u^2 dy &= r\langle u, u \rangle = r \left\langle \left(\sum_{i=1}^{\tilde{N}} \hat{u}_i \varphi_i\right), \left(\sum_{j=1}^{\tilde{N}} \hat{u}_j \varphi_j\right) \right\rangle \\ &= r \sum_{j=1}^{\tilde{N}} \hat{u}_j \hat{u}_j = \hat{U}^T R \hat{U} \end{aligned} \quad (5.12)$$

where $R = rI_{\tilde{N}}$. Thus, the cost function in (4.7) in terms of the finite-dimensional states becomes

$$J = \frac{1}{2} \int_0^\infty (\hat{X}^T Q \hat{X} + \hat{U}^T R \hat{U}) dt. \quad (5.13)$$

From (5.9) and (5.13), an analogous optimal control problem in the lumped parameter framework can be defined. This problem is solved next using the NNs in an approximate dynamic programming (ADP) framework.

VI. ADP

In this section, the general discussion on the optimal control of the distributed parameter systems is presented in an ADP framework. Detailed derivations of these conditions can also be found in [4] and [42] and are repeated here for the sake of clarity and completeness. Development in this section will subsequently be used in synthesizing the NNs for optimal cooling control of the fin.

A. Problem Description and Optimality Conditions

Assume a scalar cost function, to be minimized, of the form

$$J = \sum_{k=1}^{N-1} \Psi_k(\hat{X}_k, \hat{U}_k) \quad (6.1)$$

where $\hat{X}_k$ and $\hat{U}_k$ represent the $n \times 1$ state vector and $m \times 1$ control vector, respectively, at time step k. N represents the number of discrete time steps. $\Psi_k(\hat{X}_k, \hat{U}_k)$ is a nonlinear cost function at step k that represents the concerns of the control system designer. Note that when N is large, (6.1) represents the cost function for an infinite horizon problem. Following the aforementioned representation of the cost function, we denote the *cost function from time step* k as

$$J_k = \sum_{\tilde{k}=k}^{N-1} \Psi_{\tilde{k}}(\hat{X}_{\tilde{k}}, \hat{U}_k). \quad (6.2)$$

In a standard dynamic programming form, the cost in (6.2) is written in a recursive form by relating the cost from stage k to end given by J_k as the sum of the cost from stage $(k+1)$ to end given by J_{k+1} and the cost of going from stage k to stage $(k+1)$ denoted by Ψ_k as

$$J_k = \Psi_k + J_{k+1}. \quad (6.3)$$

Ψ_k is known as the "utility function." The $n \times 1$ *costate* vector at time step k is defined as

$$\lambda_k \equiv \frac{\partial J_k}{\partial \hat{X}_k}. \quad (6.4)$$

Then, the necessary condition for optimality is

$$\frac{\partial J_k}{\partial \hat{U}_k} = 0. \tag{6.5}$$

However

$$\frac{\partial J_k}{\partial \hat{U}_k} = \left(\frac{\partial \Psi_k}{\partial \hat{U}_k}\right) + \left(\frac{\partial J_{k+1}}{\partial \hat{U}_k}\right) = \left(\frac{\partial \Psi_k}{\partial \hat{U}_k}\right) + \left(\frac{\partial \hat{X}_{k+1}}{\partial \hat{U}_k}\right)^T \lambda_{k+1}. \tag{6.6}$$

Therefore, the optimality condition reduces to

$$\left(\frac{\partial \Psi_k}{\partial \hat{U}_k}\right) + \left(\frac{\partial \hat{X}_{k+1}}{\partial \hat{U}_k}\right)^T \lambda_{k+1} = 0. \tag{6.7}$$

The costate propagation equation can be derived in the following way:

$$\begin{aligned}\lambda_k = \frac{\partial J_k}{\partial \hat{X}_k} &= \left(\frac{\partial \Psi_k}{\partial \hat{X}_k}\right) + \left(\frac{\partial J_{k+1}}{\partial \hat{X}_k}\right) \\ &= \left[\left(\frac{\partial \Psi_k}{\partial \hat{X}_k}\right) + \left(\frac{\partial X_{k+1}}{\partial \hat{X}_k}\right)^T \lambda_{k+1}\right] \\ &\quad + \left(\frac{\partial \hat{V}_k}{\partial \hat{X}_k}\right)^T \left[\left(\frac{\partial \Psi_k}{\partial \hat{U}_k}\right) + \left(\frac{\partial \hat{X}_{k+1}}{\partial \hat{U}_k}\right)^T \lambda_{k+1}\right].\end{aligned} \tag{6.8}$$

Equation (6.8) evaluated along the optimal path given by (6.7) simplifies to

$$\lambda_k = \left(\frac{\partial \Psi_k}{\partial \hat{X}_k}\right) + \left(\frac{\partial \hat{X}_{k+1}}{\partial \hat{X}_k}\right)^T \lambda_{k+1}. \tag{6.9}$$

B. Optimality Equations for the Fin Problem

Temperature propagation equation across the fin can be written in the following discrete form as

$$\hat{X}_{k+1} = F(\hat{X}_k) + \hat{B}\hat{U}_k \tag{6.10}$$

We notice that a discrete equivalent of the cost function in (5.13) can be written as

$$J \approx \sum_{k=1}^{(N-1)\to\infty} \left(\hat{X}_k^T Q_D \hat{X}_k + \hat{U}_k^T R_D \hat{U}_k\right) \tag{6.11}$$

where $Q_D \equiv Q\Delta t$ and $R_D \equiv R\Delta t$. We also have

$$\Psi_k = \hat{X}_k^T Q_D \hat{X}_k + \hat{U}_k^T R_D \hat{U}_k \tag{6.12}$$

Then, equations for optimal control and costate can be written as

$$\hat{U}_k = -R_D^{-1}\hat{B}^T \lambda_{k+1} \tag{6.13}$$

$$\lambda_k = G(\hat{X}_k, \hat{U}_k, \lambda_{k+1}) \tag{6.14}$$

It should be noted that explicit forms of the functions F and G depend on the type of discretization procedure. With the availability of relationships in (6.10), (6.13), and (6.14), we can proceed to synthesize a neurocontroller as discussed in the following sections.

C. Single Network Adaptive Critic (SNAC)

Typically, ADP-based problems are solved by using two networks in a DHP formulation: one network to capture the relationship between the states and the control at stage k and a second network to capture the relationship between the states and the costates at stage k. In contrast, the SNAC captures the relationship between the states at k and the costates at $(k+1)$.

Even though the relationship between $\hat{X}_k$ and $\hat{\lambda}_{k+1}$ can be captured in a single network, in this paper, the network is split internally into $\tilde{N}$ subnetworks, assuming one network (rather one subnetwork) for each element of the costate vector. The input to each subnetwork, however, is the entire state vector $\hat{X}_k$. Having a separate network for each element of the costates has been found to speed up the training process since cross coupling of weights for different components of the output vector are absent.

Choosing specific architectures for the subnetworks mostly relies on experience and intuition. Use of small networks may not be adequate to capture the nonlinearities whereas large networks may lead to slower training. In this paper, five feedforward $\pi_{5,5,1}$ NNs are used. A $\pi_{5,5,1}$ NN implies that the network has five neurons in the input layer, five neurons in the hidden layer, and one neuron in the output layer. For activation functions, a *tangent sigmoid* function is used for all the hidden layers and a *linear* function is used for the output layer.

D. State Generation for NN Training

Note that the lumped parameter states can be computed from $x(y)$ as shown by

$$\hat{x}_j = \langle x(y), \varphi_j(y) \rangle. \tag{6.15}$$

Let $\hat{X}_{\max}$ and $\hat{X}_{\min}$ denote the vectors of maximum and minimum values of the elements of $\hat{X}_k$, respectively. Note that for values close to zero, the effect of nonlinear term is negligible and systems behave close to linear dynamics. Our offline training process ensures that effects of nonlinear terms come in slowly as the training set is expanded. Implementation is carried out by training in a smaller domain and increasing it to accommodate the entire range. Let $0 \le C_i \le 1$. The initial training set is obtained by setting $C_1 = 0.05$ and generating training points in $S_i^c \triangleq [C_i\hat{X}_{\min}, C_i\hat{X}_{\max}]$. Once the network is trained in this set, C_i is changed as $C_i = C_1 + 0.05(i-1)$ for $i = 2, 3 \ldots$ and the network is trained again. This process is repeated until $C_i = 1$.

E. Training of NNs

The NN training in this paper proceeds along the following steps (Fig. 2).

1) Fix C_i and generate S_i^c (as described in Section VI-C).
2) For each element $\hat{X}_k$ of S_i^c, follow the steps below.
   - Input $\hat{X}_k$ to the networks to get λ_{k+1}. Let us denote it as λ_{k+1}^a.
   - Calculate $\hat{U}_k$, knowing $\hat{X}_k$ and λ_{k+1}, from *optimal control equation* (6.13).
   - Get $\hat{X}_{k+1}$ from the *state equation* (6.10), using $\hat{X}_k$ and $\hat{U}_k$.
   - Input $\hat{X}_{k+1}$ to the networks to get λ_{k+2}.

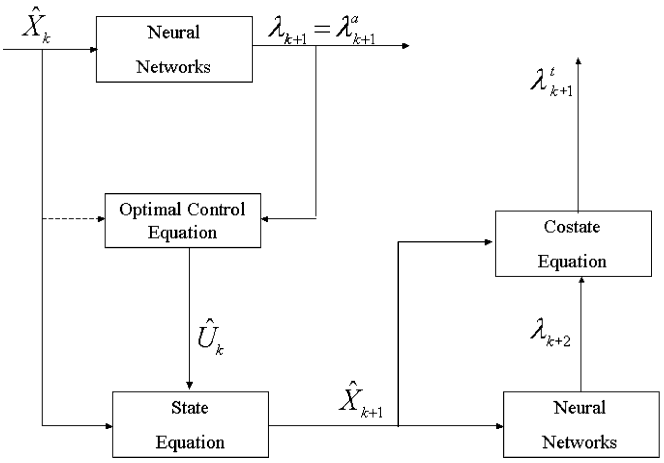

Fig. 2. Schematic of NN synthesis.

- Calculate λ_{k+1} and form the *costate equation* (6.9). Let us denote this as λ^t_{k+1}.

3) Train the networks, with all $\hat{X}_k$ as input and all corresponding λ^t_{k+1} as output.
4) If proper convergence is achieved, stop and revert to step 1), with S^c_{i+1}. If not, go to step 1) and retrain the networks with a new S^c_i.

For faster convergence, a convex combination $[\theta\lambda^t_{k+1} + (1 - \theta)\lambda^a_{k,j+1}]$, $0 < \theta < 1$ as the target output for training, is used. Moreover, to minimize the chance of getting trapped in a local minimum, the *batch training* philosophy is followed, where the network is trained for all of the elements of S^c_i, simultaneously. The Levenberg–Marquardt method [14] is used for training. For each S^c_i, 2000 input–output data points are chosen. After training the networks with 2000 data points for 25 epochs, the networks are checked for convergence (see Section VI-F) with another 2000 different data points. If the convergence condition is met, the networks are trained again with a different set of 2000 data points in S^c_{i+1} and so on. Otherwise, the training process is repeated by generating another set of random data in S^c_i.

F. Convergence Check

Before changing C_i to C_{i+1} and generating new profiles for further training, it should be assured that proper convergence is achieved for C_i. For this purpose, a training set is generated in S^c_i and is used as described in the following.

1) Fix a tolerance value (in this paper, $tol = 0.1$).
2) By using the profiles from S^c_i, generate the target outputs, as described in Section 6-C. Let the outputs be $\lambda^{t_i}_1, \lambda^{t_i}_2, \ldots, \lambda^{t_i}_{\tilde{N}}$.
3) Generate the actual output from the networks by simulating the *trained* networks with the profiles from S^c_k. The values of the outputs are $\lambda^{a_i}_1, \lambda^{a_i}_2, \ldots, \lambda^{a_i}_{\tilde{N}}$.

Check whether simultaneously $\|\lambda^{t_i}_j - \lambda^{a_i}_j\|_2 / \|\lambda^{t_i}_j\|_2 < tol$ $\forall j = 1, 2, \ldots, \tilde{N}$. If yes, it can be said that the networks have converged.

G. Implementation of the Control Solution

After the network controller is synthesized offline, it is implemented as a feedback controller, as shown in Fig. 3.

Note that the SNAC controller generates the control that will take the current fin state from any temperature profile (within the compact set used) to the desired temperature profile X_d.

VII. Robust Control Design Using Online NNs

Accurate thermal modeling of a high-speed aerospace vehicle is very difficult. It is imperative that any thermal controller is robust to uncertainties due to modeling errors or parameter variations. In this section, an extra control scheme to compensate for unmodeled dynamics is developed.

A. Problem Formulation and Uncertainty Description

Let the true model be given by

$$\frac{\partial x(t,y)}{\partial t} = f\left(x(t,y), \frac{\partial x(t,y)}{\partial y}, \frac{\partial^2 x(t,y)}{\partial^2 y}, \cdots\right) + \beta u(t,y) + D\left(x(t,y), y\right) \quad (7.1)$$

where $D(x(t,y),y)$ represents the bounded uncertainty not captured by the nominal model.

The goal is to find an extra control that can offset the effects of this uncertainty and help perform close to nominal system behavior.

Note that the uncertainty can be expanded as explained in Section V-C as follows:

$$D\left(x(t,y), y\right) = \sum_{i=1}^{\tilde{N}} \hat{d}_i(t)\varphi_i(y). \quad (7.2)$$

By using (7.2) and taking inner product of (7.1) with the basis functions, a reduced-order model for the true plant is obtained as

$$\dot{\hat{X}} = \hat{F}(\hat{X}) + B\hat{U} + \hat{D}(\hat{X}) \quad (7.3)$$

where

$$\hat{F}_i(\hat{X}) \triangleq \left\langle f\left(x(t,y), \frac{\partial x(t,y)}{\partial y}, \frac{\partial^2 x(t,y)}{\partial^2 y}, \cdots\right), \varphi_i \right\rangle$$
$$B \triangleq I_{\tilde{N}}$$
$$\hat{D}_i(\hat{X}) = \left\langle D\left(x(t,y), y\right), \varphi_i \right\rangle = \hat{d}_i(t).$$

B. Uncertainty Modeling With Online Network and Weight Updates

This section describes how the system uncertainty is modeled through an NN. The key idea is to capture the unmodeled dynamics using an NN, the weights of which are updated online. In order to ensure convergence, the weight update scheme is so chosen as to yield bounded weight estimates. Note that the output of an NN can be written as $W'\phi(\hat{X})$ where W is a matrix of weights and $\phi(\hat{X})$ is a vector of basis functions. The basis functions are chosen in such a way that $1 \leq |\phi(\hat{X})| \leq M$. It is known that within a compact set of state $\hat{X}$ there exists a matrix of weights and a vector of basis functions that can approximate the uncertainty $(\hat{D}(\hat{X}))$ into any desired accuracy, i.e.,

$$\hat{D}(\hat{X}) = W'_{\text{ideal}}\phi(\hat{X}) + \varepsilon \quad (7.4)$$

where W_{ideal} is the matrix of ideal weights and ε is the approximation error of the NN and, for any positive number $\varepsilon_{\tilde{N}}$, there exists an NN such that $|\varepsilon| \leq \varepsilon_{\tilde{N}}$.

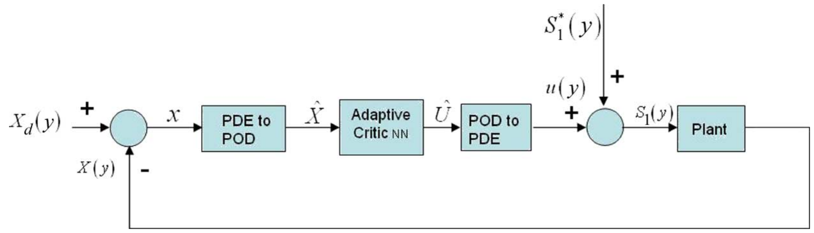

Fig. 3. Implementation of control solution.

The chosen basis functions that form the online network are $[1\ \sin(\hat{X}_i)\ \sin(2\hat{X}_i), \ldots, \sin(20\hat{X}_i)]$ for all i. Fourier series is used because of its boundedness, orthogonality, and good nonlinear function approximation capabilities. The 20 terms of the Fourier series are found to be sufficient for this application. For this NN structure, the weight update rule is presented as follows.

Let $\hat{U}_{\text{opt}}$ denote the control generated by the adaptive critic network controller for the nominal system

$$\dot{\hat{X}} = \hat{F}(\hat{X}) + B\hat{U}. \tag{7.5}$$

Let $\hat{U}_{\text{ex}}$ denote the extra control being applied to compensate for uncertainty in the model. The total control applied in is $\hat{U} = \hat{U}_{\text{opt}} + \hat{U}_{\text{ex}}$.

By substituting the previous expression for $\hat{U}$ in the actual plant model in (7.3) leads to

$$\dot{\hat{X}} = \hat{F}(\hat{X}) + B(\hat{U}_{\text{opt}} + \hat{U}_{\text{ex}}) + \hat{D}(\hat{X}). \tag{7.6}$$

By choosing $\hat{U}_{\text{ex}}$ as $-B^{-1}W'\phi(\hat{X})$, the uncertainty is shown to be compensated for. By substituting for $\hat{U}_{\text{ex}}$ and $\hat{D}(\hat{X})$ from (7.4), (7.6) becomes

$$\dot{\hat{X}} = \hat{F}(\hat{X}) + B\hat{U}_{\text{opt}} - W'\phi(\hat{X}) + W'_{\text{ideal}}\phi(\hat{X}) + \varepsilon. \tag{7.7}$$

Now, an "approximate" system mimicking the nominal system is defined as follows:

$$\dot{\hat{X}}_a = \hat{F}(\hat{X}) + B\hat{U}_{\text{opt}} + K(\hat{X} - \hat{X}_a) \tag{7.8}$$

where K is of the form $K = kI$ (k is scalar and I is the identity matrix).

This (observer-type) system (7.8) is introduced to get an approximation of the error between the states and (7.5) and (7.7). The extra term $K(\hat{X} - \hat{X}_a)$ is introduced to make (reduce) the error between its states and (7.6) and when $\hat{X} \rightarrow \hat{X}_a$, (7.5) and (7.8) become identical. We prove this by choosing the online NN weight update rule as

$$\dot{W} = \Gamma^{-1}\left(\phi(\hat{X})\hat{e}' - k\delta|\hat{e}|W\right) \tag{7.9}$$

where $\hat{e} \triangleq \hat{X} - \hat{X}_a$. Note that the weights are guaranteed to be bounded and the bound on error between the approximate plant and the actual plant can be made as small as possible by choosing design parameters k and δ (see the Appendix). Γ is the learning rate of the network. The first term in (7.9) is used to realize a good approximation of the uncertainty in the model and the second term helps ensure the boundedness of the weights.

C. Application to High-Speed Aerospace Vehicle Problem

It can be seen that the coefficients in the model are functions of the desired temperature profile. Since the control scheme was presented for the profile given by X_d, for any other desired profile (say X_d^*), the equation for deviation of states X from X_d^* can be written as

$$\frac{\partial x}{\partial t} = A_0\left(\frac{\partial^2 x}{\partial y^2}\right) + F\left(x, X_d^*\right) + u$$

where

$$x = X - X_d^*. \tag{7.10}$$

It can be seen that the optimal control scheme developed cannot be used directly to achieve X_d^*. In order to achieve any desired profile, the SNAC network must be trained again. In order to use the same network trained to yield X_d to help achieve X_d^*, add and subtract $f(x, X_d)$ in (7.10) to get

$$\frac{\partial x}{\partial t} = A_0\left(\frac{\partial^2 x}{\partial y^2}\right) + F(x, X_d) + u + \left(F\left(x, X_d^*\right) - F(x, X_d)\right). \tag{7.11}$$

This manipulation allows us to treat the last term $(F(x, X_d^*) - F(x, X_d^*)$ as *unmodeled dynamics*. Now, we can rewrite (7.11) as

$$\frac{\partial x}{\partial t} = A_0\left(\frac{\partial^2 x}{\partial y^2}\right) + F(x, X_d) + u + D(x)$$

where

$$D(x) = \left(F\left(x, X_d^*\right) - F(x, X_d)\right). \tag{7.12}$$

By using the online NN approach to compensate for this unmodeled dynamics, any desired profile X_d^* can be achieved.

It should be pointed out that yet another type of important uncertainty in physics modeling can be accommodated in this framework. Viscous forces in high temperature are difficult to model accurately. The heat generated due to viscosity can be

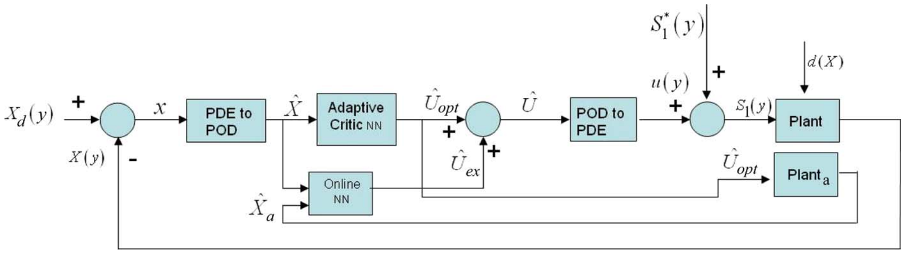

Fig. 4. Implementation of robust control solution.

TABLE II
PARAMETER DEFINITIONS AND VALUES

Parameter	Definition	Numerical value
q	Weight on deviation from desired state	1
r	Weight on control	1
k	Design parameter for online network	5
σ	Design parameter for online network	0.1
Γ	Learning rate of online network	$.75I_{100}$
μ	Coefficient of Viscosity	1.2×10^{-7}~ 1.7×10^{-5}
$V_{y_{1fin}}\dfrac{\partial V_{y_1}}{\partial z_1}$	Velocity gradient term in heat equation	100

considered as an unmodeled uncertainty. The shear stress generated due to viscosity is given by [1]

$$\tau_w = \mu \frac{\partial V_{y_1}}{\partial z_1}\bigg|_{\text{fin}} \tag{7.13}$$

where y_1 is along and z_1 is perpendicular to the motion of the vehicle (also fin surface), V_{y_1} is the velocity of the air along y_1, and μ is viscosity of air and varies linearly with temperature. The viscosity term is evaluated at the surface of the fin. The heat transfer equation with the effect of viscosity becomes

$$\frac{\partial x}{\partial t} = A_0\left(\frac{\partial^2 x}{\partial y^2}\right) + F(x, X_d) + \frac{\mu V_{y_{1fin}}}{\rho c_P}\left(\frac{\partial V_{y_1}}{\partial y_1}\right)\bigg|_{\text{fin}} + u. \tag{7.14}$$

The robust control scheme formulated in this paper can handle this type of uncertainty as well.

D. *Implementation of the Robust/Optimal Control*

Fig. 4 shows the control solution implementation scheme that compensates for model uncertainties and/or change of desired temperature profiles.

VIII. NUMERICAL RESULTS

Numerical simulations were performed in MATLAB. Five basis functions ($\tilde{N} = 5$) were found to be adequate to describe the high-temperature model for an aerospace vehicle. Simulation results are presented in three parts. First part shows the results from the SNAC design where the desired temperature profile was a constant temperature along the spatial direction. Second part presents the results from using the robust control scheme. In this paper, this relates to achieving any desired temperature profile from a set of initial profiles. The third section discusses results where the true plant has a viscosity-related term and a resultant heat addition over the nominal model. Values of the parameters used in the simulations are presented in Table II.

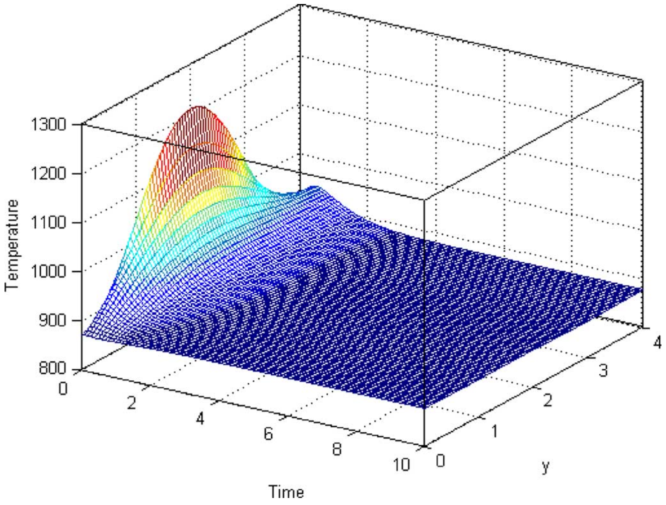

Fig. 5. Temperature profile.

A. *SNAC Controller*

This section presents the results obtained by using the SNAC controller to achieve a constant temperature of 873 K across the fin. It can be seen from the 3-D temperature history presented in Fig. 5 that the desired final temperature profile is reached.

Fig. 6 presents the control effort from the adaptive critic controller on the reduced-order side transformed to the original side. Note that this additional control is used (on the reduced-order

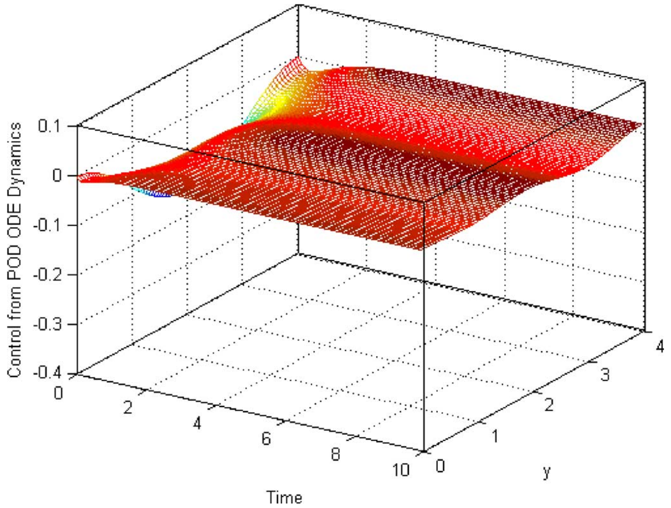

Fig. 6. Control output of the network.

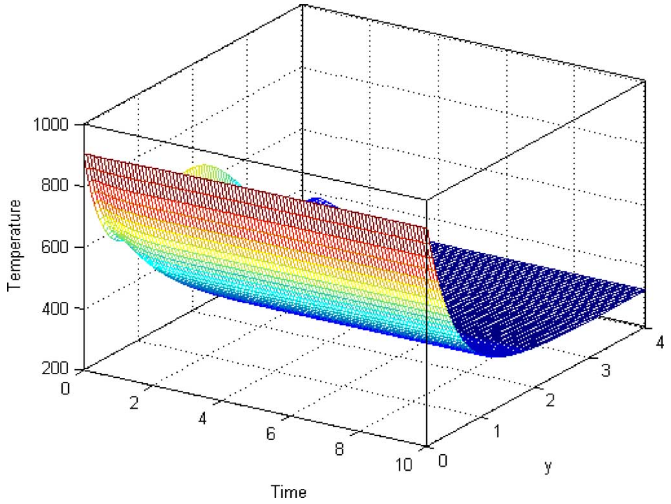

Fig. 7. Temperature profile variation with time.

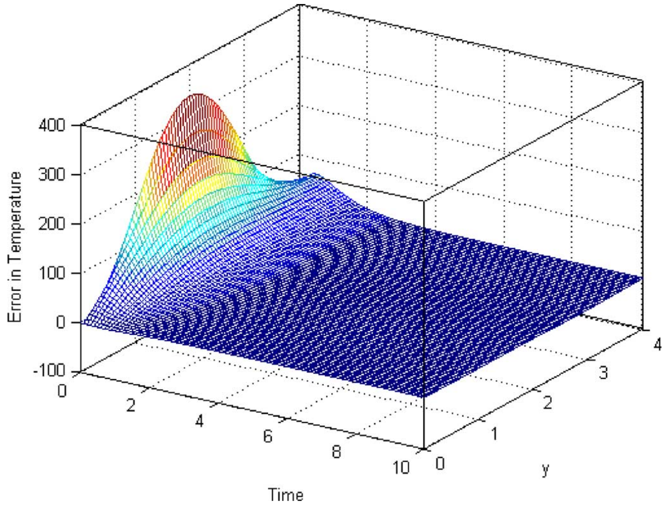

Fig. 8. Error between actual temperature and the desired temperature.

side) to take the deviations (again on the reduced-order side) to zero. The steady-state control is not added to the computed control. The finite dimensional model is used to compute the additional amount of control needed to drive the error between the desired profile and the current profile to zero; then, the basis functions are used to transform this control effort to a true control and applied to the original model. As to be expected, this history decays to zero, as in Fig. 6.

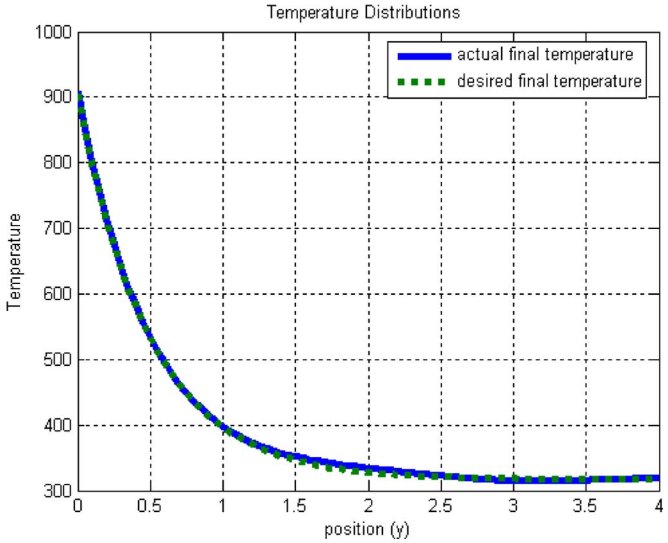

Fig. 9. Temperature distribution.

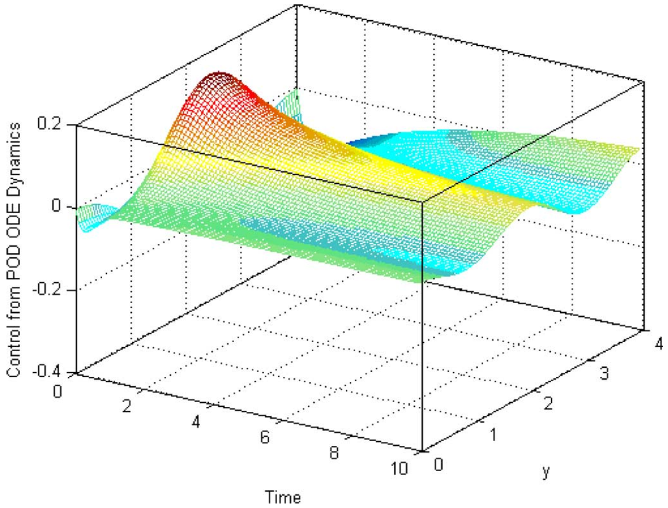

Fig. 10. Control history.

B. Online NN to Reach Any Desired Final Profile

The method proposed in Section VII-C is tested in a case to obtain an exponential profile. If the SNAC controller is used alone to drive the error between the desired state and the final state to zero, then the desired temperature is not achieved because the SNAC was designed to obtain a single specific desired profile although it can achieve it from any starting temperature profile. However, with the use of the online NN for compensation of unmodeled dynamics, any desired profile can be achieved. The parameters used for the simulations of this network are $k = 5$ and $\delta = 0.01$.

Fig. 7 presents the temperature history in the case of the desired exponential profile. Note that the temperature error history (from the desired profile) is presented in Fig. 8 and it goes to zero with time. Fig. 9 shows the desired temperature profile along the fin and the actual temperature at steady state. Note

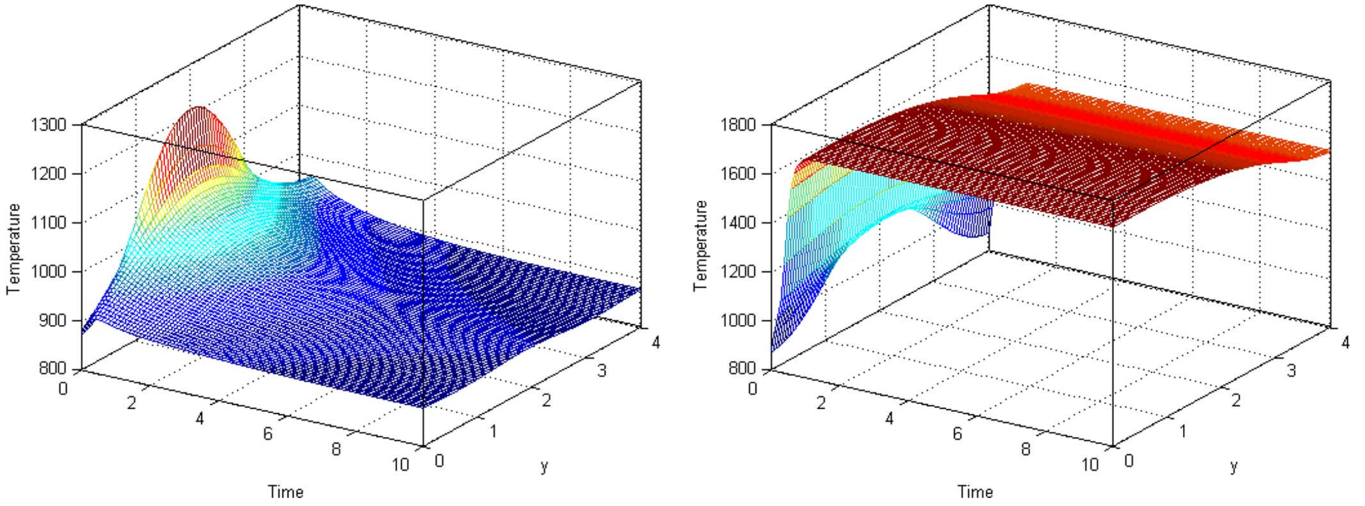

Fig. 11. Temperature variation with time with and without uncertainty compensation.

that they are almost identical. Control history, a sum of SNAC output and extra control, is presented in Fig. 10.

C. Online NN (Robustness to Viscosity Effects)

This section presents the simulation results where the controller design is shown to be robust to unmodeled viscous forces. In the first case, the SNAC is designed as in Section VIII-A to produce a constant temperature across the fin using a model without the viscous terms. The online NN is used to compensate for the viscous effects. It can be seen from Fig. 11 that the desired final temperature is achieved when there is online compensation but it is quite different without it. Fig. 12 shows that the final temperature achieved with the online network and without it. The end results are quite different.

IX. Conclusion

In this paper, ADP-based formulations were used to synthesize suboptimal neurocontrollers for high-speed aerospace vehicles traversing through the atmosphere. An adaptive-critic-based SNAC controller was shown to be able to drive any given initial temperature profile to a desired profile. In order to compensate for the effect of uncertainty and to use the same adaptive critic network for achieving any desired profile, an online NN was used. The weight update rule proposed in this paper ensures boundedness of the weight estimates, and hence, relaxes the persistence of excitation condition. It was shown in simulation studies that the proposed robust control scheme can achieve any desired end profile and can compensate for viscous effects that are difficult to model (unmodeled dynamics). The technique developed in this paper is implementable. Simulation results demonstrate that the proposed technique has excellent promise and could be very useful for a variety of applications since the formulation uses very few assumptions in its development.

Appendix

In this section, it is shown that the weight update scheme (7.9) drives the error between actual and approximate plant to zero and also guarantees that the weights are bounded. First, an expression for the error dynamics is obtained. Next, a Lyapunov-function-based analysis is used to obtain an upper bound on magnitude of norms of error and it shows that the weight matrix of the network is bounded. As the weights are bounded, the need to satisfy the persistence of excitation condition is relaxed. Variables and parameters used in the proof are listed in Table III. All the norms used in the proof are two-norms.

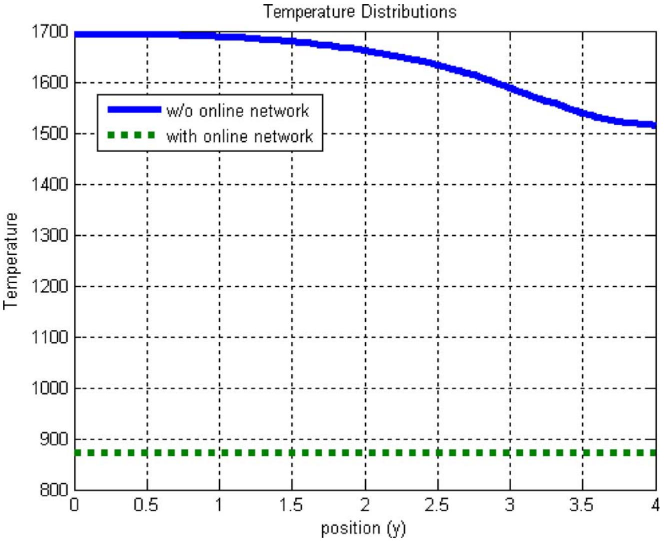

Fig. 12. Final temperatures with and without uncertainty compensation.

By defining $\hat{e} \triangleq \hat{X} - \hat{X}_a$ and $\tilde{W} \triangleq W_{\text{ideal}} - W$, equations for the error and the weight update are

$$\begin{aligned}\dot{\hat{e}} &= -K\hat{e} + \tilde{W}'\phi(\hat{X}) + \varepsilon \\ \dot{W} &= \Gamma^{-1}\left(\phi(\hat{X})\hat{e}' - k\delta|\hat{e}|W\right)\end{aligned} \tag{A.1}$$

where

$$\hat{e} \triangleq \hat{X} - \hat{X}_a. \tag{A.2}$$

By defining a Lyapunov function candidate as

$$V = \frac{1}{2}\hat{e}'\hat{e} + \frac{1}{2}Tr(\tilde{W}'\Gamma\tilde{W}) \tag{A.3}$$

TABLE III
DESCRIPTIONS OF VARIABLES AND PARAMETERS

Variable/Parameter	Description	Dimension
$\hat{e}$	Error between actual plant and approximate plant	$\tilde{N}\times 1$
$K = kI_{\tilde{N}}$	A design parameter	$K = \tilde{N}\times\tilde{N}$ $k = 1\times 1$
ε	Online network's approximation error	$\tilde{N}\times 1$
W_{ideal}	Ideal matrix of weights	$M\times\tilde{N}$
W	Actual matrix of weights	$M\times\tilde{N}$
$\tilde{W}$	Difference between ideal and actual weights	$M\times\tilde{N}$
$\phi(\hat{X})$	Vector of basis functions	$M\times 1$
$D\big(x(t,y),y\big)$	Actual uncertainty in the PDE system	scalar
$\hat{D}(\hat{X})$	Actual uncertainty in the ODE system	$M\times 1$
B_D	Upper bound on $\lvert\hat{D}(\hat{X})\rvert$	scalar
$\varepsilon_{\hat{N}}$	Upper bound on $\lvert\varepsilon\rvert$	scalar
Γ	Learning rate of online network	$M\times M$
δ	A design parameter to control error bound	scalar
V	Lyapunov function candidate	scalar
E	Error between actual uncertainty and network approximation	$\tilde{N}\times 1$

and taking time derivative of (A.3), we get

$$\dot{V} = \hat{e}'\dot{\hat{e}} + Tr(\tilde{W}'\Gamma\dot{\tilde{W}}). \tag{A.4}$$

By substituting error equation in (A.4)

$$\begin{aligned}\dot{V} =& \hat{e}'\left(-K\hat{e} + \tilde{W}'\phi(\hat{X}) + \varepsilon\right) + Tr(\tilde{W}'\Gamma\dot{\tilde{W}})\\ =& -\hat{e}'K\hat{e} + \hat{e}'\tilde{W}'\phi(\hat{X}) + \hat{e}'\varepsilon + Tr(\tilde{W}'\Gamma\dot{\tilde{W}}).\end{aligned} \tag{A.5}$$

Rewrite $\hat{e}'\tilde{W}'\phi(\hat{X})$ in (A.5) in a trace form to get

$$\begin{aligned}\dot{V} =& -\hat{e}'K\hat{e} + Tr\left(\tilde{W}'\phi(\hat{X})\hat{e}'\right) + \hat{e}'\varepsilon + Tr(\tilde{W}'\Gamma\dot{\tilde{W}})\\ =& -\hat{e}'K\hat{e} + \hat{e}'\varepsilon + Tr\left(\tilde{W}'\left(\phi(\hat{X})\hat{e}' + \Gamma\dot{\tilde{W}}\right)\right).\end{aligned} \tag{A.6}$$

Since $\tilde{W} = W_{\rm ideal} - W$ and $\dot{\tilde{W}} = -\dot{W}$, (A.6) can be rewritten as

$$\dot{V} = -\hat{e}'K\hat{e} + \hat{e}'\varepsilon + Tr\left(\tilde{W}'\left(\phi(\hat{X})\hat{e}' - \Gamma\dot{W}\right)\right). \tag{A.7}$$

By using the weight update equation, the trace term in is modified to get

$$\begin{aligned}\dot{V} =& -\hat{e}'K\hat{e} + \hat{e}'\varepsilon + Tr\left(\tilde{W}'k\delta|\hat{e}|W\right)\\ =& -\hat{e}'K\hat{e} + \hat{e}'\varepsilon + Tr\left((W_{\rm ideal} - W)'k\delta|\hat{e}|W\right)\\ =& -\hat{e}'K\hat{e} + \hat{e}'\varepsilon + k\delta|\hat{e}|Tr\left(W'_{\rm ideal}W\right)\\ & -k\delta|\hat{e}|Tr(W'W).\end{aligned} \tag{A.8}$$

Note that $\hat{e}'K\hat{e} = k|\hat{e}|^2$ and $Tr(W'W) = ||W||_F^2$. Therefore, (A.8) becomes

$$\dot{V} = -k|\hat{e}|^2 + \hat{e}'\varepsilon + k\delta|\hat{e}|Tr\left(W'_{\rm ideal}W\right) - k\delta|\hat{e}|||W||_F^2. \tag{A.9}$$

The third term in has a summation representation given by

$$Tr\left(W'_{\rm ideal}W\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{\rm ideal}^{ij}w^{ij}. \tag{A.10}$$

It can be proved that

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j=1}^{m} w_{\rm ideal}^{ij}w^{ij}\\ &\leq \sqrt{\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\left(w_{\rm ideal}^{ij}\right)^2\right)\left(\sum_{i=1}^{n}\sum_{j=1}^{m}(w^{ij})^2\right)}\\ &= ||W_{\rm ideal}||_F||W||_F \leq W_{\max}||W||_F.\end{aligned} \tag{A.11}$$

Hence

$$\dot{V} \leq -k|\hat{e}|^2 + |\hat{e}||\varepsilon| + k\delta|\hat{e}|W_{\max}||W||_F - k\delta|\hat{e}|||W||_F^2. \tag{A.12}$$

A. *Upper Bounds on $|\hat{e}|$ and $||W||_F$*

1) Upper bound on $|\hat{e}|$. In order to establish an upper bound on $|\hat{e}|$, the square on $||W||_F$ needs to be completed in

$$\begin{aligned}\dot{V} \leq& -k|\hat{e}|^2 - k\delta|\hat{e}|\\ &\times\left(\frac{|\varepsilon|}{k\delta} - W_{\max}||W||_F + ||W||_F^2\right)\end{aligned} \tag{A.13}$$

$$\begin{aligned}\dot{V} \leq& -k|\hat{e}|^2 - k\delta|\hat{e}|\\ &\times\left(\frac{|\varepsilon|}{k\delta} + \left(||W||_F - \frac{W_{\max}}{2}\right)^2 - \frac{W_{\max}^2}{4}\right)\end{aligned} \tag{A.14}$$

$$\begin{aligned}\dot{V} \leq& -k|\hat{e}|\left(|\hat{e}| - \frac{|\varepsilon|}{k} - \delta\frac{W_{\max}^2}{4}\right)\\ &-k\delta|\hat{e}|\left(||W||_F - \frac{W_{\max}}{2}\right)^2.\end{aligned} \tag{A.15}$$

Equation (A.15) implies that

$$|\hat{e}| \geq \frac{|\varepsilon|}{k} + \delta\frac{W_{\max}^2}{4}, \qquad \dot{V} \leq 0. \tag{A.16}$$

Therefore, $|\hat{e}|$ is upper bounded although the bound is conservative. However, this bound can be made as small as desired with a proper choice of the design parameters k and δ.

2) Upper bound on $||W||_F$. To establish an upper bound on $||W||_F$, the square term with $|\hat{e}|$ is completed

$$\dot{V} \leq -k|\hat{e}|^2 - k\delta|\hat{e}| \left(||W||_F^2 - W_{\max}||W||_F - \frac{|\varepsilon|}{k\delta} \right). \quad (A.17)$$

The Lyapunov function is decreasing if the expression inside [•] in (A.17) is positive. That is, if

$$||W||_F^2 \geq \frac{W_{\max} + \sqrt{W_{\max}^2 + 4\frac{|\varepsilon|}{k\delta}}}{2}. \quad (A.18)$$

Therefore, if

$$||W||_F^2 \geq \frac{W_{\max} + \sqrt{W_{\max}^2 + 4\frac{|\varepsilon|}{k\delta}}}{2}, \qquad \dot{V} \leq 0 \quad (A.19)$$

then, from (A.16) and (A.18), it can be concluded that $|\hat{e}|$ and $||\tilde{W}||_{L_2}$ are uniformly upper bounded.

Acknowledgment

The authors would like to thank Dr. D. C. Look, Emeritus Professor, University of Missouri-Rolla, Rolla, for the heat-transfer-related discussions. They would also like to thank the anonymous reviewers whose detailed and specific comments have improved this paper.

References

[1] A. D. John, *Introduction to Flight*, 4th ed. New York: McGraw-Hill, 1999.

[2] B. B. Aloliwi and H. K. Khalil, "Adaptive output feedback regulation of a class of nonlinear systems: Convergence and robustness," *IEEE Trans. Autom. Control*, vol. 42, no. 12, pp. 1714–1716, Dec 1997.

[3] A. Annaswamy, J. J. Choi, and D. Sahoo, "Active closed-loop control of supersonic impinging jet flows using POD models," in *Proc. 41st IEEE Conf. Decision Control*, Las Vegas, NV, 2002, pp. 3294–3299.

[4] S. N. Balakrishnan and V. Biega, "Adaptive-critic based neural networks for aircraft optimal control," *J. Guid. Control Dyn.*, vol. 19, no. 4, pp. 893–898, Jul.–Aug. 1996.

[5] J. Burns and B. B. King, "A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems," *J. Vibration Control*, vol. 4, no. 3, pp. 297–323, 1998.

[6] H. T. Banks, R. C. H. Rosario, and R. C. Smith, "Reduced-order model feedback control design: Numerical implementation in a thin shell model," *IEEE Trans. Autom. Control*, vol. 45, no. 7, pp. 1312–1324, Jul. 2000.

[7] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. London, U.K.: Taylor & Francis, 1975.

[8] A. J. Calise and R. T. Rysdyk, "Non-linear flight control using neural networks," *IEEE Control Syst. Mag.*, vol. 18, no. 6, pp. 14–25, Dec. 1998.

[9] A. J. Calise and R. T. Rysdyk, "Adaptive model inversion flight control for tiltrotor aircraft," in *Proc. AIAA Conf. Guid. Navigat. Control*, New Orleans, LA, Aug. 11–13, 1997, AIAA-97-37001-10–63.

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

[11] A. Emami-Neini, J. L. Ebert, D. de Roover, R. L. Kosut, M. Dettori, L. M. L. Porter, and S. Ghosal, "Modelling and control of distributed thermal systems," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 5, pp. 668–683, Sep. 2003.

[12] S. Ferrari and R. F. Stengel, "An adaptive critic global controller," in *Proc. Amer. Control Conf.*, 2002, pp. 2665–2670.

[13] S. K. Gupta, *Numerical Methods for Engineers*. New Delhi, India: Wiley, 1995.

[14] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. Boston, MA: PWS-Kent, 1996.

[15] S. Haykin, *Neural Networks*. New York: Macmillan, 1994.

[16] N. Hovakimyan, F. Nardi, K. Nakwan, and A. J. Calise, "Adaptive output feedback control of uncertain systems using single hidden layer neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1420–1431, Nov. 2001.

[17] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, U.K.: Cambridge Univ. Press, 1996, pp. 87–154.

[18] Z. Huang and S. N. Balakrishnan, "Robust neurocontrollers for systems with model uncertainties: A helicopter application," *J. Guid. Control Dyn.*, vol. 28, no. 3, pp. 516–523, 2005.

[19] K. J. Hunt, "Neural networks for control systems—A survey," *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.

[20] P. Ioannou and P. Kokotovic, *Adaptive System With Reduced Model*. New York: Springer-Verlag, 1983.

[21] J. P LaSalle, "Some extensions of Lyapunov's second method," *IRE Trans. Circuit Theory*, pp. 520–527, Dec 1960.

[22] I. Lasiecka, "Control of systems governed by partial differential equations: A historical perspective," in *Proc. 34th Conf. Decision Control*, 1995, pp. 2792–2796.

[23] J. Leintner, A. Calise, and J. V. R. Prasad, "Analysis of adaptive neural networks for helicopter flight controls," *AIAA J. Guid. Control Dyn.*, vol. 20, no. 5, pp. 972–979, Sep–Oct 1997.

[24] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.

[25] F. Lewis, *Applied Optimal Control and Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[26] X. Liu and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Amer. Control Conf.*, 2000, pp. 1929–1933.

[27] M. B. McFarland, R. T. Rysdyk, and A. J. Calise, "Robust adaptive control using single-hidden- layer feedforward neural networks," in *Proc. Amer. Control Conf.*, Jun. 1999, pp. 4178–4182.

[28] A. F. Miller, *Basic Heat and Mass Transfer*. Homewood, IL: Irwin, 1995.

[29] J. J. Murray, C. J. Cox, C. G. Lendaris, and R. E. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[30] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent adaptation," *IEEE Trans. Autom. Control*, vol. AC-32, no. 2, pp. 134–145, Feb. 1987.

[31] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[32] R. Padhi, S. N. Balakrishnan, and T. W. Randolph, "Adaptive-critic based optimal neuro control synthesis for distributed parameter systems," *Automatica*, vol. 37, pp. 1223–1234, 2001.

[33] R. Padhi, N. Unnikrishnan, and S. N. Balakrishnan, "Optimal control synthesis of a class of nonlinear systems using single network adaptive critics," in *Proc. Amer. Control Conf.*, Jul. 2004, vol. 2, pp. 1592–1597.

[34] D. V. Prokhorov and D. C. Wunsch, II, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[35] S. S. Ravindran, "Proper orthogonal decomposition in optimal control of fluids," NASA, TM-1999-209113.

[36] S. S. Ravindran, "Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition," *SIAM J. Sci. Comput.*, vol. 23, no. 6, pp. 1924–1942, 2002.

[37] S. N. Singh, J. H. Myatt, and G. A. Addington, "Adaptive feedback linearizing control of proper orthogonal decomposition nonlinear flow models," in *Proc. Amer. Control Conf.*, 2001, pp. 1533–1538.

[38] L. Sirovich, "Turbulence and the dynamics of coherent structures: Part I-III," *Quarterly Appl. Math.*, vol. 45, no. 3, pp. 561–590, 1987.

[39] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[40] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford, U.K.: Clarendon, 1985.

[41] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbo generator," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 764–773, May 2002.

[42] P. J. Werbos, "Neuro control and supervised learning: An overview and evaluation," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, D. White and D. Sofge, Eds. New York: Van Nostrand, 1992.

[43] D. White and D. Sofge, Eds., *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. New York: Van Nostrand, 1992.

Vivek Yadav received the B.Tech. degree from the Indian Institute of Technology, Madras, India, in 2003 and the M.S. degree from the University of Missouri—Rolla, Rolla, in 2006, both in mechanical engineering. Currently, he is working towards the Ph.D. degree at the Ohio State University, Columbus.

He has worked on control of distributed parameter systems and use of NNs for obstacle avoidance and communication control.

Radhakant Padhi received the M.E. degree from Indian Institute of Science, Bangalore, India, in 1996 and the Ph.D. degree from the University of Missouri—Rolla (UMR), Rolla, in 2001, both in aerospace engineering.

He has worked as a Postdoctoral Fellow at the UMR for two years after completing. Currently, he is an Assistant Professor at the Department of Aerospace Engineering, Indian Institute of Science. He was also a Scientist at the RCI/DRDO, Hyderabad, India, from 1996 to 1997. He has numerous publications in international journals and conferences, a couple of which has won best paper awards. His research interests are in the broad area of systems theory and applications. He has won a couple of best paper awards.

S. N. Balakrishnan received the M.S. and Ph.D. degrees in aerospace engineering from the University of Texas, Austin, in 1974 and 1983, respectively.

Currently, he is a Professor at the Department of Mechanical and Aerospace Engineering, University of Missouri—Rolla. He has numerous publications in international journals and conferences, and his students have won many best paper awards. He has developed a single network adaptive critic that can be used to solve the two-point boundary value problem using only one NN. His research interests are in the broad area of nonlinear control theory, optimal control, and NN-based controls. Specifically, he is interested in nonlinear control design and analysis, optimization and optimal control, application of NNs in control system design, control and guidance of aerospace vehicles, distributed parameter systems, and application of control theory to biomedical systems.