JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 1, April 2004

Application of Scattered Data Approximation to a Rotorcraft Heath Monitoring Problem

Andrew J. Meade, Jr.*
William Marsh Rice University, Houston, Texas 77251-1892

An adaptive and matrix–free scheme has been developed for interpolating and approximating sparse multi–dimensional scattered data and has been applied to a time-series problem in rotorcraft. The Sequential Function Approximation (SFA) method is based on a sequential Galerkin approach to artificial neural networks and requires neither ad–hoc parameters for the user to tune, nor rescaling of the inputs. It is linear in storage with respect to the number of samples. The SFA method has been used to model and extrapolate the time series data from a critical temperature sensor in a 1/4 scale model of the V-22 Osprey. The SFA regression model, constructed with radial basis functions, has also been used satisfactorily to evaluate the sensitivity to 74 system health and safety–of–flight parameters during a series of wind–tunnel tests. An upper bound on the error convergence rate that is exponential and does not explicitly depend on the dimensionality of the approximation was derived and confirmed for the time series data.

Nomenclature

a	=	positive constant > 1
C	=	positive constant
c_n	=	n–th linear coefficient in the approximation
d	=	input dimension
f	=	arbitrary function
$\boldsymbol{f}$	=	vector of f evaluated at sample points
G	=	mapping function of Eq. (12)
$g_n = r_{(n\text{-}1)}\left(\boldsymbol{\xi}^*_{\mathbf{n}}\right)$	=	value of component in vector $\boldsymbol{r}_{(n\text{-}1)}$ with maximum magnitude
$\mathrm{H}^v(\mathrm{R}^d)$	=	Sobolev space
j^*	=	component index of $\boldsymbol{r}_{(n\text{-}1)}$ with the maximum magnitude
m	=	number of basis parameters
n	=	number of bases
R	=	real coordinate space
$r_n(\boldsymbol{\xi})$	=	n–th stage of the function residual
$\boldsymbol{r}_n$	=	n–th stage function residual vector
s	=	number of samples
t^i	=	i–th time level
$u(\boldsymbol{\xi})$	=	target function
$u_n^a(\boldsymbol{\xi})$	=	n–th stage of the target function approximation
v	=	arbitrary function
$\langle f, v\rangle = \int_\Omega fv\, d\boldsymbol{\xi}$	=	inner product of f and v
$\langle f, v\rangle_D = \Sigma_i^s(f_i v_i)$	=	discrete inner product of f and v
$\lvert f\rvert$	=	absolute value of f

Received 22 April 2003; revision received 5 February 2004; accepted for publication 9 February 2004.

*Associate Professor, Department of Mechanical Engineering and Materials Science, MS 321, P. O. Box 1892; meade@rice.edu. Associate Fellow AIAA Member.

$\|f\|_2 = \left(\int_\Omega f^2 d\boldsymbol{\xi}\right)^{1/2}$ = L_2 norm of f

$\|f\|_{2,D} = \left(\Sigma_i^s (f_i)^2\right)^{1/2}$ = discrete L_2 norm of f

Greek Symbols

α_n	=	n–th linear coefficient of Eq. (3)
$\boldsymbol{\beta}$	=	set of nonlinear optimization parameters
η	=	basis function argument
$\boldsymbol{\xi}$	=	d–dimensional input of the target function
$\boldsymbol{\xi}_i$	=	i–th sample input of the target function
$\boldsymbol{\xi}_n^*$	=	sample input with component index j^* at the n–th stage
$\xi_i^{(k)}$	=	k–th component of the sample input vector $\boldsymbol{\xi}_i$
σ	=	radial basis function parameter
ϕ	=	basis function
τ	=	user specified tolerance
Ω	=	domain of interest

Subscripts

i	=	dummy index
j^*	=	component index of $\boldsymbol{r}_{(n-1)}$ with the maximum magnitude
n	=	associated with the number of bases
s	=	associated with the number of samples

Superscripts

d	=	associated with the input dimension
m	=	associated with the number of basis parameters

1. Introduction

WITH the costs of testing facilities continuing to rise, rotorcraft experimentalists are increasingly faced with conflicting requirements of efficiency and safety. In an ideal situation, the experimentalist has a priori knowledge of test conditions that may cause safety problems such as high structural loads or overheating of critical parts. Realistically, however, this knowledge is quite limited and therefore the test engineer will approach suspect or new test conditions in a methodical matrix-like manner. This approach, while safe, can be inefficient. For example, dozens of system health and safety-of-flight parameters must be monitored during testing of rotorcraft scale models. Many of the drive train temperatures operate near redline limits and react differently to various test conditions. If it were possible to model the behavior of critical system health values as a function of the test parameters, the test engineer could map an approach to a particular test condition prior to the start of the data run safely and efficiently.

To develop a health monitoring system (HMS) with this capability, a high-fidelity and easy to use tool is needed to assemble and analyze databases of sparse and scattered data so important trends can be easily and quickly identified. In this paper we introduce an adaptive and matrix-free scheme developed for interpolating and approximating sparse multi-dimensional scattered data. The scheme requires neither ad-hoc parameters for the user to tune, nor rescaling of the inputs. This has been applied to a time series regression problem in a HMS. The Sequential Function Approximation (SFA) method[1] is based on a sequential Galerkin approach to artificial neural networks and is linear in storage with respect to the number of samples, s. The method is simple to program and does not require selecting ad-hoc parameter values or rescaling of the inputs. The SFA approach and its implementation will be discussed in Secs. 2.2 and 2.3. In Sec. 3 the SFA scheme will be used to model HMS time histories from the temperature sensor of a critical component in a scale model of the V-22 Osprey, known as the Full Span Tilt Rotor Aeroacoustic Model (FS-TRAM)[2] illustrated in Fig. 1. The constructed SFA model will then be evaluated to determine the sensitivity of the sensor readings to the 74 parameters that include rotor blade conditions and temperatures, rotor speed, and fluid pressures throughout the FS-TRAM. Error bounds for the application are derived and confirmed. Conclusions and future work are given in Sec. 4.

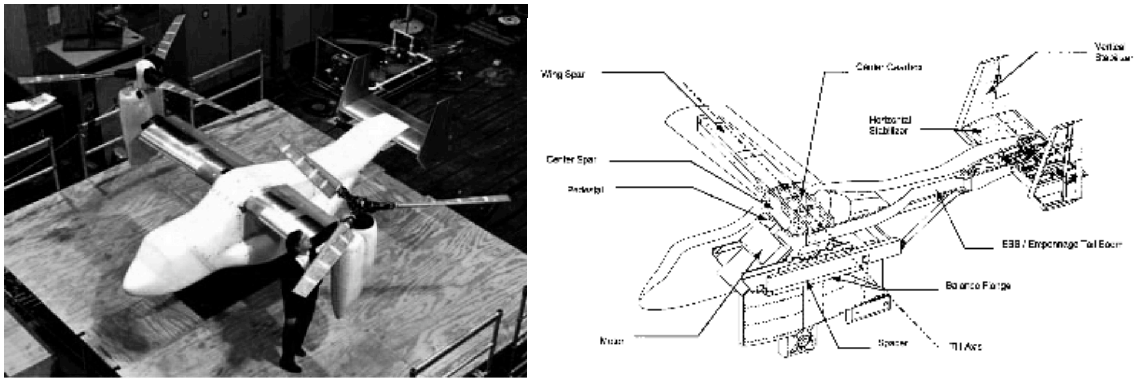

Fig. 1 Full span tilt rotor aeroacoustic model: (a) configuration, (b) schematic.

2. Approach

In this section, the proposed method for matrix-free sequential function interpolation and approximation is described and implemented for scattered multidimensional and real-valued data. Approximation by artificial neural networks is discussed as well as optimization and basis functions.

2.1 Function Approximation in Artificial Neural Network Application

Any approximation algorithm that uses a combination of basis functions mapped into a graph-directed representation can be called an artificial neural network. Function approximation in the framework of neural computations has been based primarily on the results of Cybenko[3] and Hornik et al.,[4] who showed that a continuous *d*-dimensional function can be arbitrarily well-approximated by a linear combination of one-dimensional functions ϕ.

$$u \approx u_n^a(\boldsymbol{\xi}) = c_0 + \sum_{i=1}^{n} c_i \phi\left(\eta\left(\boldsymbol{\xi}, \boldsymbol{\beta}_i\right)\right) \tag{1}$$

where $\eta \in R, \boldsymbol{\xi} \in R^d, \boldsymbol{\beta}_i \in R^m$, and $c_i \in$ R represent the function argument, independent variables, function parameters, and linear coefficients, respectively.

The appropriate linear and nonlinear network parameters in Eq. (1) are traditionally selected by solving a non-linear optimization problem with the objective function given by the L_2 norm of the error over some domain Ω

$$\varepsilon^2 \equiv \int_{\Omega} \left(u - u_n^a\right)^2 d\boldsymbol{\xi} = \left\| u - u_n^a \right\|_2^2 \tag{2}$$

In the neural network literature, the numerical minimization of Eq. (2) by a gradient descent procedure is known as the backpropagation algorithm.[5] More sophisticated optimization methods, including the conjugate gradient and the Levenberg-Marquardt methods, have also been used for neural network training. However, it has been found that even these advanced optimization methods are prone to poor convergence.[6] Clearly, the training algorithms must address a multidimensional optimization problem with non-linear dependence on the network parameters β_i.

As an alternative, Jones[7,8] and Barron[9] proposed the following iterative algorithm for sequential approximation:

$$u_n^a(\boldsymbol{\xi}) = \alpha_n u_{n-1}^a(\boldsymbol{\xi}) + c_n \phi\left(\eta\left(\boldsymbol{\xi}, \boldsymbol{\beta}_n\right)\right) \tag{3}$$

where $\boldsymbol{\beta}_n, c_n$, and α_n are selected optimally at each iteration of the algorithm. As a result, the high-dimensional optimization problem associated with neural network training is reduced to a series of simpler low-dimensional problems. A general principle of statistics was utilized to show that the upper bound of the error ε is of the order $C/\sqrt{n}$, where C is a positive constant. Orr[10] then introduced a forward selection method of sequential network training, in effect a method of incremental function approximation. At each iteration of the algorithm an additional

basis function producing the largest reduction of error in the previous iteration is chosen from a given set of functions and added to the approximation. This forward selection training method can be inefficient as it may require significant computational resources when the set of trial functions is large. A similar principle is utilized in Platt's resource allocating networks (RAN).[11] Whenever an unusual pattern is presented to the network in on- or off-line network training a new computational "unit" is allocated. Note that these computational units respond to local regions of the input space.

The concept of sequential approximation is one of the major features of the method proposed in this paper for the solution of scattered data approximation problems.

2.2 The Proposed Method

The basic principles of the proposed computational method for solving scattered data approximation are presented in this section. The development of this method was motivated by the similarities between iterative optimization procedures reviewed in Sec. 2.1 and the Method of Weighted Residuals (MWR), specifically the Galerkin method.[12]

To begin, we can write the function residual using the n-th stage of Eq. (1) as

$$r\left(c_n,\boldsymbol{\xi},\boldsymbol{\beta}_n\right)=r_n=u\left(\boldsymbol{\xi}\right)-u_n^a\left(\boldsymbol{\xi}\right)=u_{(n-1)}^a\left(\boldsymbol{\xi}\right)-c_n\phi\left(\eta\left(\boldsymbol{\xi},\boldsymbol{\beta}_n\right)\right)=$$
$$r_{(n-1)}-c_n\phi\left(\eta\left(\boldsymbol{\xi},\boldsymbol{\beta}_n\right)\right)=r_{(n-1)}-c_n\phi_n$$

Utilizing the Petrov-Galerkin approach, we select a c_n that will force the function residual to be orthogonal to the basis function,

$$\left\langle r_n,\phi_n\right\rangle=-\left\langle r_n,\frac{\partial r_n}{\partial c_n}\right\rangle=-\frac{1}{2}\frac{\partial\left\langle r_n,r_n\right\rangle}{\partial c_n}=0 \tag{3}$$

which is equivalent to selecting a value of c_n that will minimize $\left\langle r_n,r_n\right\rangle$ or

$$c_n=\frac{\left\langle\phi_n,r_{(n-1)}\right\rangle}{\left\langle\phi_n,\phi_n\right\rangle} \tag{4}$$

The values of our remaining variables $\left(\boldsymbol{\beta}_n\right)$ must minimize the same objective

$$\left\langle r_n,r_n\right\rangle=\left\langle r_{(n-1)},r_{(n-1)}\right\rangle-2c_n\left\langle\phi_n,r_{(n-1)}\right\rangle+c_n^2\left\langle\phi_n,\phi_n\right\rangle \tag{5}$$

that can be rewritten with the substitution of Eq. (4) as

$$\left\langle r_n,r_n\right\rangle=\left\langle r_{(n-1)},r_{(n-1)}\right\rangle\left(1-\frac{\left\langle\phi_n,r_{(n-1)}\right\rangle^2}{\left\langle\phi_n,\phi_n\right\rangle\left\langle r_{(n-1)},r_{(n-1)}\right\rangle}\right) \tag{6}$$

Recalling the definition of the cosine using arbitrary functions f and v,

$$\cos\left(\theta\right)=\frac{\left\langle f,v\right\rangle}{\left\langle f,f\right\rangle^{1/2}\left\langle v,v\right\rangle^{1/2}}$$

and the equivalence between the inner product and the square of the L_2 norm, Eq. (6) can be written as

$$\left\|r_n\right\|_2=\left\|r_{(n-1)}\right\|_2\sin\left(\theta_n\right) \tag{7}$$

where θ_n is the angle between ϕ_n and $r_{(n-1)}$. With Eq. (7) $\|r_n\|_2 < \|r_{(n-1)}\|_2$ as long as ϕ_n is not orthogonal to the previous equation residual $r_{(n-1)}$ that is, $\theta_n \neq \pi/2$. By inspection the minimum of Eq. (7) is

$$c_n\phi_n = c_n\phi\left(\eta\left(\boldsymbol{\xi},\boldsymbol{\beta}_n\right)\right) = r_{(n-1)}$$

Therefore, to force $\|r_n\|_2 \to 0$ a low-dimensional function approximation problem must be solved at each stage n. This involves an unconstrained nonlinear optimization in the determination $\boldsymbol{\beta}_n$. The dimensionality of the nonlinear optimization problem is kept low since we are solving for only one basis at a time. There are no theoretical restrictions on the type and distribution of local basis functions. Both $B1$-splines[13,14] and Gaussian tensors[15] have been used with Eq. (7) using equation residuals in the solution of ordinary and partial differential equations. These popular bases should be available for use in scattered data approximation since Ref. 16 demonstrated that approximating functions through the solution of differential equations was a more stringent test of bases than straight-forward function approximation. There are neither ad-hoc parameters for the user to tune, nor rescaling of the inputs in the method. The algorithm can be initialized with either an empty set ($r_0 = u$) or an arbitrary number of predetermined functions and coefficients. The second option enables the use of solutions from previous numerical analyses.

2.3 Implementation of the Algorithm

In the scattered data approximation we have only discrete samplings of the observed function $u(\xi)$. Given a finite number of samples s, we can write $\boldsymbol{r}_{(n-1)} = \left\{r_{(n-1)}(\boldsymbol{\xi}_1),...,r_{(n-1)}(\boldsymbol{\xi}_s)\right\}$ and $\boldsymbol{\phi}_n = \left\{\phi_n(\boldsymbol{\xi}_1),...,\phi_n(\boldsymbol{\xi}_s)\right\}$. The equations derived in Sec. 2.2 are directly applicable if we use the discrete inner product.

The choice of bases plays a major role in the efficiency and utility of the scattered data approximation algorithm as a whole. We chose normalized bell-shaped bases. Radial basis functions (RBF) were used in the modeling of the time series data from a critical temperature sensor. Defining $\boldsymbol{\xi}_n^*$ such that $\left|r_{(n-1)}\left(\boldsymbol{\xi}_n^*\right)\right| = \max\left|\boldsymbol{r}_{(n-1)}\right|$, then $\boldsymbol{\beta}_n = \left\{\boldsymbol{\xi}_n^*, \sigma_n\right\}$ where

$$\phi\left(\eta\left(\boldsymbol{\xi},\boldsymbol{\beta}_n\right)\right) = \phi_n(\boldsymbol{\xi}) = \exp\left(-\sigma_n^2\left(\boldsymbol{\xi}-\boldsymbol{\xi}_n^*\right)\cdot\left(\boldsymbol{\xi}-\boldsymbol{\xi}_n^*\right)\right)$$

Though the functional form of the SFA scheme allows the basis center to be located anywhere in R^d, the application to scattered data approximation constrains the centers to the set of sample points $\{\boldsymbol{\xi}_1,...,\boldsymbol{\xi}_s\}$, The remaining optimization variable σ_n is continuous and unconstrained.

Numerical experiments show that the discrete inner product formulation of Eq. (6) has a number of local minima with respect to σ_n. This results in an unsatisfactory convergence rate when using common gradient-based optimization methods. These experiments resulted in the two-step formulation used for this paper which is a combination of Eqs. (4) and (5).

$$\min_{\sigma_n}\left(\left\langle r_{(n-1)}, r_{(n-1)}\right\rangle_D - 2g_n\left\langle\phi_n, r_{(n-1)}\right\rangle_D + g_n^2\langle\phi_n,\phi_n\rangle_D\right) \tag{8}$$

$$c_n = \frac{\left\langle\phi_n, r_{(n-1)}\right\rangle_D}{\langle\phi_n,\phi_n\rangle_D} \tag{9}$$

where $g_n = r_{(n-1)}\left(\boldsymbol{\xi}_n^*\right)$. This is the form of the SFA algorithm used in the modeling of the time series from a critical temperature sensor. Note that by using the scalar g_n with a normalized bell-shaped basis function centered at $\boldsymbol{\xi}_n^*$ that $\max\left|\boldsymbol{r}_n\right| \leq \max\left|\boldsymbol{r}_{(n-1)}\right|$ while at the same time the minimization of Eq. (8) and satisfaction of Eq. (9) ensures that

$\langle r_n, r_n \rangle_D < \langle r_{(n-1)}, r_{(n-1)} \rangle_D$. When using RBFs in this implementation of the SFA method the basis parameter σ_n should vary as $Ca^{n/v}$. Then, as per the results of Ref. 16, the upper bound of the residual should be

$$\langle r_n, r_n \rangle_D = \|r_n\|_{2,D} \leq Ca^{-n} \tag{10}$$

where $u(\boldsymbol{\xi})$ is from the space $H^v(R^d)$. In this implementation, Eq. (10) indicates that we should observe exponential convergence that does not depend explicitly on the dimensionality of the approximation. It sidesteps the "curse of dimensionality".[8] Together Eqs. (8) and (9) make up the SFA algorithm used in the HMS time series regression problem.

The two more popular scattered data approximation methods using RBFs in the literature are RBF neural networks and Support Vector Machines (SVM). RBF neural networks usually consist of a single hidden layer whose number of transfer functions, n, are chosen through trial and error by the user. The nonlinear parameters of the hidden layer are determined simultaneously through optimization and the output parameters are solved by the inversion of an n x n matrix, where $n \leq s$. RBF neural networks, besides requiring on the order of n^2 memory and time resources, are also strongly susceptible to problems of dimensionality.[17] Support Vector Machines on the other hand are very attractive for high-dimensional regression since the complexity of the approximation depends only on the number of support vectors and is independent of the dimensions of the input d (Ref. 18). However, SVMs require user tuned ad-hoc parameters and on the order of s^2 memory and time resources.[19]

The scattered data approximation scheme presented in this paper is free of matrix construction and evaluations and is only of order s in storage. Computational cost is reduced while efficiency is enhanced by the low-dimensional unconstrained optimization. Theoretically, the method should be applicable to a number of popular local interpolation functions and should, with the use of RBFs, sidestep the limitations due to dimensionality. The scheme does not require user interaction other than setting the tolerance for termination; the computer does the work. To implement the algorithm the user takes the following steps:

1) Initiate the algorithm with $\boldsymbol{r}_0 = \{u(\boldsymbol{\xi}_1), ..., u(\boldsymbol{\xi}_s)\}$.

2) Search the components of $\boldsymbol{r}_{(n-1)}$ for the maximum magnitude. Record the component index j^*.

3) $\boldsymbol{\xi}_n^* = \boldsymbol{\xi}_{j^*}$.

4) Satisfy Eq. (8) with ϕ_n centered at $\boldsymbol{\xi}_{j^*}$ and initialize the optimization parameter with $\sigma_n = 0$.

5) Calculate the coefficient c_n from Eq. (9).

6) Update the residual vector $\boldsymbol{r}_n = \boldsymbol{r}_{(n-1)} - c_n \boldsymbol{\phi}_n$. Repeat cycle until termination criteria have been met.

A combination of three criteria can be used to terminate the SFA algorithm: $\|r_n\|_{2,D}$, or $|g_n|$ falls below a user specified tolerance (τ) or the number of bases n exceeds the user specified maximum. In this paper we terminate the calculations when either $|g_n| \leq \tau$ or $n \geq s$.

The method is linear in storage with respect to s since it needs to store only s+1 vectors to compute the residuals: one vector of length s ($\boldsymbol{r}_n$) and s vectors of length d $(\boldsymbol{\xi}_1, ...\boldsymbol{\xi}_s)$. To generate the SFA model requires one vector of length n ($\{c_1, \ldots, c_n\}$) and n vectors of length n+1 ($\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n$).

A drawback found in using Eqs. (8) and (9) in practice is its sensitivity to the initial value of σ_n. Though Eq. (6) is stable, numerical experiments have shown that some initial values of σ_n cause local extrema in the formulations of Eqs. (8) and (9) to be selected such that $c_n \rightarrow 0$ while g_n is finite. Problems are not experienced with the SFA scheme if we initialize with $\sigma_n = 0$.

Lastly, since the SFA scheme works with the L_2 norm, time-dependent processes must be reformulated as a mapping of inputs to desired outputs. The temperature variations measured by the FS-TRAM target sensors is a heat transfer problem and should be governed by a system of unknown nonlinear time-dependent partial differential equations. Assume the inputs are sequential in time, or $\boldsymbol{\xi}_i$ is recorded at t^i where $t^i < t^{i+1}$. Limiting our attention to a

single target sensor, we can use the Runge-Kutta temporal integration technique from the finite difference literature[20] to write

$$u\left(t^{i+1}\right) = u\left(t^{i}\right) + \left(t^{i+1} - t^{i}\right) G\left(\boldsymbol{\xi}_i, u\left(t^{i}\right), t^{i}\right) \tag{11}$$

where G is unknown. If the input vector $\boldsymbol{\xi}_i$ includes $u(t_i)$ as well as t^i, and if t^{i+1}-t^i is roughly constant for all i, then Eq. (11) can be rewritten as

$$u\left(t^{i+1}\right) = G\left(\boldsymbol{\xi}_i\right) \tag{12}$$

since G is still an unknown function. Therefore, with the sets $\left\{u\left(t^1\right),\cdots,u\left(t^s\right)\right\}$ and $\{\boldsymbol{\xi}_1,\cdots,\boldsymbol{\xi}_s\}$ we can use the SFA scheme to construct the mapping function G sequentially.

3. Results

Results are presented, compared, and discussed to demonstrate the utility of the SFA scattered data approximation approach in health monitoring. We applied the method to data acquired by the FS-TRAM HMS during eight test runs in NASA Ames Research Center's 80 x 120 ft wind-tunnel. The FS-TRAM HMS is a real-time monitoring and data acquisition system that tracks various fluid pressures and temperatures from numerous model components including the conversion axis, swashplate, static mast bearing, centerline gearbox, and nacelle transmissions. During a test run, if any component reaches thermal limits or exhibits unusual gradients, necessary modifications are made to the test plan that may include the premature cancellation of the run.

The results of this paper were chosen assuming the safety and test engineers have laid out a detailed experimental test plan beforehand and want to determine how to delay or eliminate redline conditions during the run. To develop a HMS with this capability the SFA method that augments it should

1) interpolate data scattered over the range of system health and safety-of-flight parameters accurately using a minimum number of basis functions and with as little user interaction as possible,

2) test the scattered data approximation for input parameter sensitivity so that necessary modifications can be made to the test plan to avoid the thermal limits,

3) use the sensitivity results to eliminate the need to sample certain test parameters in future test runs (i.e. develop a reduced model), and

4) predict the values of critical sensors so as to provide advanced warning on redline conditions.

The mapping of 74 experimental test and system health parameters (Figs. 2-4) at time t^i to the right swashplate bearing J-type temperature sensor (number 4 of Fig. 5) at time t^{i+1} is investigated using the SFA algorithm. We selected the data from eight FS-TRAM test runs (Fig. 6) that vary the fuselage angle of attack (-9°– 11°), rotor shaft angle (0 °– 14° from vertical), the presence of nacelle fairings, and wind-tunnel speeds. The HMS recorded the FS-TRAM system health parameters approximately every 30 seconds from 0 to a maximum of 116 minutes. These tests were conducted over a number of days producing a total of 1607 test samples. As mentioned in Sec. 2.3, a combination of three criteria can be used to terminate the calculation. In this paper we terminated the computer code when either $|g_n| \le \tau$ or $n \ge s$. Since the J-type temperature sensors used in the FS-TRAM experiment were accurate to only $\pm 2\,^{\circ}$F, we set $\tau = 2$ with $s = 1607$. For the purpose of display clarity, the temperature predictions are represented as curves in some of the figures rather than data points. This does not imply a functional relationship between adjoining samples. A measure of the SFA model's sensitivity relative to the input parameters was determined by summing the squares of the partial derivatives of each of the inputs over all the test samples, i.e.,

$$\text{sensitivity to input index } k \quad = \quad \sum_{i=1}^{s} \left(\left. \frac{\partial u_n^a(\boldsymbol{\xi})}{\partial \xi^{(k)}} \right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} \right)^2$$

$$= \quad 4\sum_{i=1}^{s} \left(\sum_{j=1}^{n} c_j \sigma_j^2 \left(\xi_i^{(k)} - \xi_{j*}^{(k)} \right) \phi_j(\boldsymbol{\xi}_i) \right)^2 \tag{13}$$

The SFA algorithm was encoded as a Matlab™ program and all calculations were made on a Dell GX260 personal computer with a 2 GHz processor. We selected the trust-region based nonlinear optimization routine (*fminunc*) to calculate σ_n.

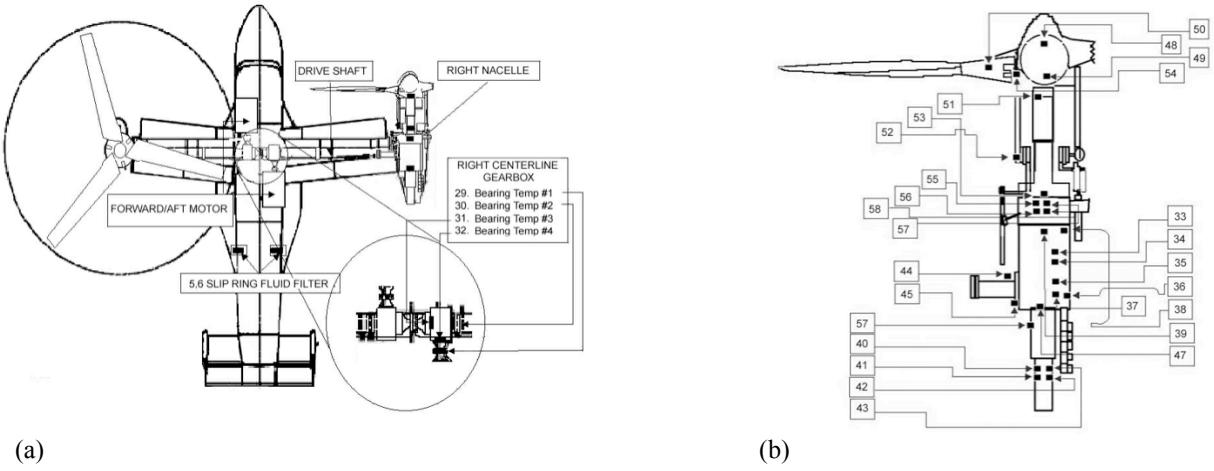

Fig. 2 Location of input sensors: (a) fuselage, (b) right nacelle. Sensors in the left nacelle not shown.

3.1 Time Series Model

Figure 7 illustrates the SFA time series approximation, error, convergence rate, and sensitivity ranking for the temperature sensor of our critical part over the eight test runs. Evaluation of the approximation parameters required 260 seconds in wall clock time. The right swashplate rolling-element bearing is considered the most mechanically critical part of the FS-TRAM model of the five that were monitored during the experiment. The swashplate transfers linear actuator motion to cyclic and collective control of the rotor blades with the rotating rotor shaft running through its center. The swashplate bearing supports the rotating section. As the bearing surface deteriorates in the rotating section, the resulting increase in friction will cause a notable increase in the bearing body temperature which can be monitored by the HMS. If the temperature exceeds the set redline value the swashplate bearing could cause immediate loss of the rotor pitch/collective controls which in turn could result in a catastrophic rotor system failure and the loss of the model.

Input Index	Abbreviation	Definition (Units)
1	TIME	Time Elapsed (Min.).
2	BTCLG1_L	Left Bearing Temperature, Centerline Gearbox #1 (deg F).
3	BTCLG2_L	Left Bearing Temperature, Centerline Gearbox #2 (deg F).
4	BTCLG3_L	Left Bearing Temperature, Centerline Gearbox #3 (deg F).
5	BTCLG4_L	Left Bearing Temperature, Centerline Gearbox #4 (deg F).
6	BTNT2_L	Left Bearing Temperature, Nacelle Transmission #2 (deg F).
7	BTNT3_L	Left Bearing Temperature, Nacelle Transmission #3 (deg F).
8	BTNT4_L	Left Bearing Temperature, Nacelle Transmission #4 (deg F).
9	BTNT5_L	Left Bearing Temperature, Nacelle Transmission #5 (deg F).
10	BTNT7_L	Left Bearing Temperature, Nacelle Transmission #7 (deg F).
11	BTNT8_L	Left Bearing Temperature, Nacelle Transmission #8 (deg F).
12	BTNT9_L	Left Bearing Temperature, Nacelle Transmission #9 (deg F).
13	SRT1_L	Left Slipring Temperature #1 (deg F).
14	SRT2_L	Left Slipring Temperature #2 (deg F).
15	CABT1_L	Left Conversion Axis Bearing Temperature #1 (deg F).
16	CABT2_L	Left Conversion Axis Bearing Temperature #2 (deg F).
17	OILTNT_L	Left Oil Reservoir Temperature, Nacelle Transmission (deg F).
18	HUBT1_L	Left Hub Temperature #1 (deg F).
19	HUBT2_L	Left Hub Temperature #2 (deg F).
20	PCT_L	Left Pitch Case Temperature (deg F).
21	TEMPUB_L	Left Upper Bearing Temperature (deg F).
22	TEMPSPB_L	Left Swashplate Bearing Temperature (deg F).
23	TEMPSMBL_L	Left Static Mast Lower Bearing Temperature (deg F).
24	TQLNKTC_L	Left Torque Link Temperature, Location C (deg F).
25	TEMPBC_L1	Left Balance Flexure Coupling Temperature, Primary (deg F).
26	TMCOUT_L	Left Balance Flexure Temperature, Metric Side (deg F).
27	TGCOUT_L	Left Balance Flexure Temperature, Ground Side (deg F).
28	RPM	Rotor Speed (rpm).
29	BTCLG1_R	Right Bearing Temperature, Centerline Gearbox #1 (deg F).
30	BTCLG2_R	Right Bearing Temperature, Centerline Gearbox #2 (deg F).
31	BTCLG3_R	Right Bearing Temperature, Centerline Gearbox #3 (deg F).
32	BTCLG4_R	Right Bearing Temperature, Centerline Gearbox #4 (deg F).
33	BTNT1_R	Right Bearing Temperature, Nacelle Transmission #1 (deg F).
34	BTNT3_R	Right Bearing Temperature, Nacelle Transmission #3 (deg F).
35	BTNT4_R	Right Bearing Temperature, Nacelle Transmission #4 (deg F).
36	BTNT5_R	Right Bearing Temperature, Nacelle Transmission #5 (deg F).

Fig. 3 Temperature sensor inputs. Input indices 1-36.

The right and left swashplate assemblies are exposed to outside airflow during the experiment which may explain why the temperature of the right bearing reaches only up to only 115°F as shown in Fig. 7a while other FS-TRAM components can easily exceed 230°F. Figure 7a also illustrates the temperature variations caused by the changes in rotor speed and the cyclic and collective settings. The errors shown in Fig. 7b are caused by the plateaus in the temperature curve since sensor readings that vary less than ±2 °F were approximated as a single value.
Because the convergence rate of Fig. 7 and the remaining figures in this paper follow or exceed that given in Eq. (10), the results are considered satisfactory since they show that the SFA time series approximation can be constructed accurately using the optimum number of bases. The sensitivity plot of Fig. 7d indicates that this sensor signal is overwhelmingly sensitive to rotor speed (input index 28) as expected, followed by the right slip-ring temperature (input index 43) and temperatures of nearby bearings and the elapsed time (input index 1). The proximity of other bearings inside the same component may explain the sensitivity of the right swashplate temperature to those sensors though it is unknown why the sensitivity to the left pitch case temperature (input index 20) should be present. It is possible this data channel may have been mislabled.

3.2 Model Reduction

In an effort to show the insensitivity of the SFA approximation to the majority of input indices, the input values for all samples were reset to zero with the exception of 14 of the most sensitive inputs: indices 1, 3, 4, 5, 20, 28, 29, 31, 32, 35, 46, 51, 55, and 56. The SFA algorithm was then used to map this sparse set of parameters to the critical component sensor readings. Figure 8 illustrates the temperature approximation, error, convergence rate, and sensitivity ranking again using a maximum residual of 2.0 with 1607 samples. Evaluation of the approximation parameters required 266 seconds. When comparing Figs. 7d and 8d it is apparent that the sensitivities of the majority of indices are zero as expected, but the relative rankings of the 14 most sensitive parameters are unchanged as are the convergence rates. Figure 8d confirms that the insensitivities of Fig. 7d are not just artifacts of the SFA method and that it is possible to reduce the complexity of the SFA model.

Input Index	Abbreviation	Definition (Units)
37	BTNT7_R	Right Bearing Temperature, Nacelle Transmission #7 (deg F).
38	BTNT8_R	Right Bearing Temperature, Nacelle Transmission #8 (deg F).
39	BTNT9_R	Right Bearing Temperature, Nacelle Transmission #9 (deg F).
40	SRT1_R	Right Slipring Temperature #1 (deg F).
41	SRT2_R	Right Slipring Temperature #2 (deg F).
42	SRT4_R	Right Slipring Temperature #4 (deg F).
43	SRT5_R	Right Slipring Temperature #5 (deg F).
44	CABT1_R	Right Conversion Axis Bearing Temperature #1 (deg F).
45	CABT2_R	Right Conversion Axis Bearing Temperature #2 (deg F).
46	OILRTCLG	Oil Reservoir Temperature, Centerline Gearbox (deg F).
47	OILTNT_R	Right Oil Reservoir Temperature, Nacelle Transmission (deg F).
48	HUBT1_R	Right Hub Temperature #1 (deg F).
49	HUBT2_R	Right Hub Temperature #2 (deg F).
50	PCT_R	Right Pitch Case Temperature (deg F).
51	TEMPUB_R	Right Upper Bearing Temperature (deg F).
52	TEMPSPB_R	Right Swashplate Bearing Temperature (deg F).
53	TEMPSMBL_R	Right Static Mast Lower Bearing Temperature (deg F).
54	TQLNKTC_R	Right Torque Link Temperature, Location C (deg F).
55	TEMPBC_R1	Right Balance Flex Coupling Temperature, Primary (deg F).
56	TEMPBC_R2	Right Balance Flex Coupling Temperature, Backup (deg F).
57	RASTEMP	RAS External Temperature (deg F).
58	TMCOUT_R	Right Balance Flexure Temperature, Metric Side (deg F).
59	TGCOUT_R	Right Balance Flexure Temperature, Ground Side (deg F).
60	SLUCSFT_1	SLUCS Fluid Filter Temperature #1 (deg F).
61	SLUCSFT_2	SLUCS Fluid Filter Temperature #2 (deg F).
62	# OF MOTORS	Number of Motors.
63	A1_R TH EU MEAN	Lateral Cyclic from Actuators (degrees).
64	ASHAFT_R TH EU MEAN	Right Rotor Shaft Angle (degrees).
65	B1_R TH EU MEAN	Longitudinal Cyclic from Actuators (degrees).
66	CP_R1 TH EU MEAN	Rotor Power Coefficient, Right Rotor.
67	CT_R1 TH EU MEAN	Rotor Thrust Coefficient, Right Rotor.
68	CX_R1 TH EU MEAN	Rotor X-Force Coefficient, Right Rotor.
69	MTIP TH EU MEAN	Rotor Tip Mach Number.
70	MU TH EU MEAN	Advance Ratio.
71	THETA_R TH EU MEAN	Collective Pitch, Right Rotor (degrees).
72	ALPHA_F	Fuselage Pitch Angle (degrees).
73	NF	Nacelle Fairings. On (1) or Off (0).
74	NO	Nacelle Fairings Openings. Open (0) or Taped Closed (1).

Fig. 4 Temperature sensor inputs. Input indices 37-74.

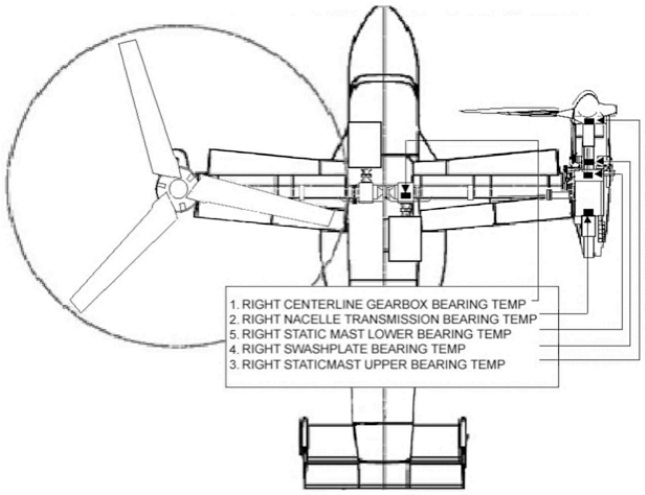

Fig. 5 Location of critical temperature sensors.

Test Run Title	Wind-Tunnel	ALPHA_F	ASHAFT_R TH EU MEAN	NF	NO	# of Samples
Run 143	Not Blowing	Varied	0	1	1	137
Run 144	Not Blowing	Varied	0	0	0	120
Run 189	Not Blowing	-9	-14	0	0	94
Run 191	Not Blowing	-9	-14	0	0	288
Run 198	Blowing	5	0	1	1	220
Run 199	Blowing	11	-5	1	1	199
Run 201	Blowing	11	-5	1	1	324
Run 202	Blowing	11	-5	1	1	225

Fig. 6 FS-TRAM test run summary.

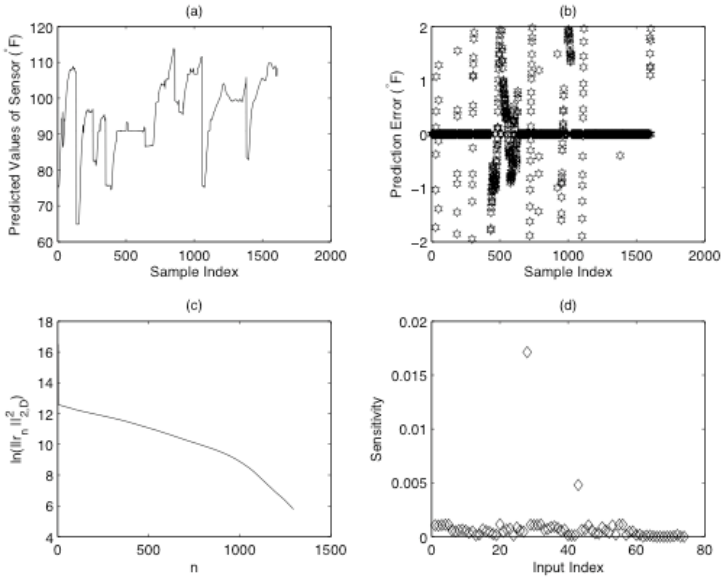

Fig. 7 SFA time series model of the critical sensor using 1296 bases for 74 inputs: (a) predicted temperature, (b) prediction error, (c) convergence rate of the residual, (d) input sensitivity.

3.3 Model Extrapolation

As previously mentioned, we believe one of the greatest needs of the test and safety engineers is a method for predicting the values of critical sensors so advance warning on redline conditions can be given. However, extrapolation is the most challenging test for any scattered data approximation scheme. Though Poggio and Girosi[21] have shown that RBFs can approximate arbitrarily well any continuous function, accurate extrapolation by data alone (a ``black-box mode'') depends on the sufficient quantity and quality of data. If either of these conditions are not satisfied, extrapolation may still be possible with the proper incorporation of expert knowledge and/or combination with physics-based models through regularization.[22]

Figure 9 is an attempt to extrapolate the mapping constructed using the 74 inputs and data from all the test runs with the exception of Run 201. Evaluation of the approximation parameters required 251 seconds. After consultations with the FS-TRAM engineers it was determined an extrapolation with an error less than 5°F would be satisfactory. Unfortunately, as Fig. 9b shows, our extrapolation error for Run 201 immediately exceeds 28°F. This poor performance seems to be primarily caused by the overwhelming dependence of the SFA model on the rotor speed within the training domain. The abrupt change in rotor speed between sample indices 244 and 265 of Run 201 (Fig. 9c) also affected the extrapolation of the sensor model (Fig. 9a). The sensitivity rankings shown in Fig. 9d are identical to those of Fig 7d. The extrapolation results using the 14 most sensitive inputs of Sec. 3.2 are not shown since they are identical to Fig. 9.

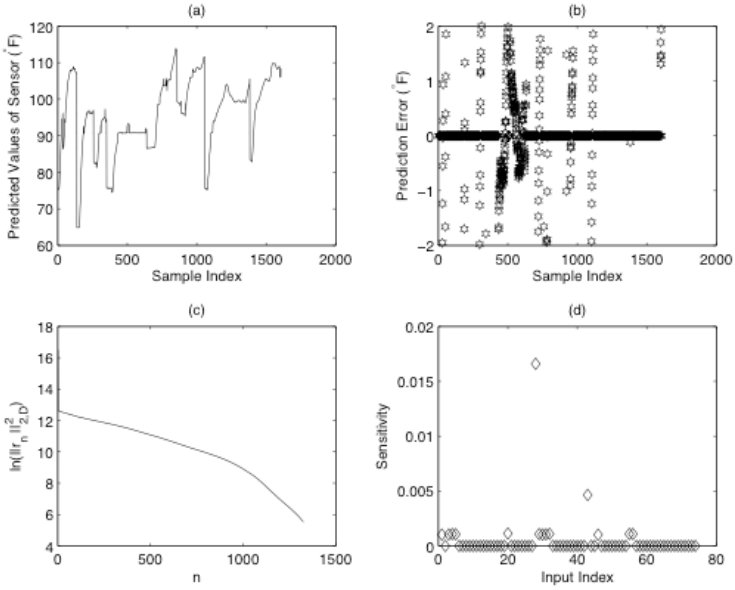

Fig. 8 SFA time series model of the critical sensor using 1326 bases for the 14 most sensitive inputs: (a) predicted temperature, (b) prediction error, (c) convergence rate of the residual, (d) input sensitivity.

It was the opinion of the FS-TRAM engineers that the sensitivities given in Fig. 9d were not necessarily valid for similar experimental test runs performed previously. From their observations the overall results were more dependent on input indices 1, 29, 35, 51-53, 62-67, 69, and 70. The test engineers based their recommended inputs on observations from many more experiments than those supplied to the author. In an attempt to incorporate this expert knowledge, Fig. 10 shows the extrapolation results using the 14 recommended inputs. Evaluation of the approximation parameters required 136 seconds. The performance is qualitatively satisfactory with an error less than $\pm 10\,^{\circ}\mathrm{F}$, though still insufficient for the purpose of health monitoring.

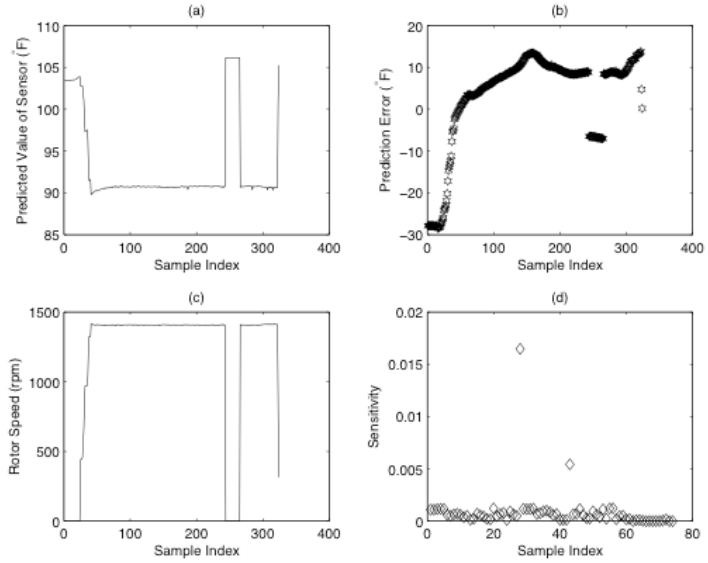

Fig. 9 SFA time series extrapolation of the critical sensor for Run 201; 1015 bases were used for 74 inputs: (a) predicted temperature, (b) prediction error, (c) rotor speed, (d) input sensitivity.

Figures 9 and 10 illustrate the problem central to extrapolation in black-box mode; the mapping can only be constructed with available data. Equation (13) used to measure the relative sensitivity, is a gross measure since it is

summed over all of the available data sets. As more data sets are added it is possible for the sensitivity measures to change. This may explain why the ranking of our sensitivity measures do not match those observed by the FS-TRAM engineers who have access to more data sets than those used in this study. Though our black-box mode of extrapolation using the available data failed, the tool can still be useful in helping the engineer iteratively select physics-based models for further attempts at extrapolation. For example, in an actual FS-TRAM experiment the engineer could take part of the available data and test the extrapolation against the remaining data in black-box mode. With unsatisfactory results, the engineer could then take the rankings of the sensitivity measures and select appropriate physics-based models to incorporate in the next extrapolation attempt. In future work we will incorporate a mathematical model of the heat transferred from the right nacelle transmission to the right swashplate bearing temperature sensor.

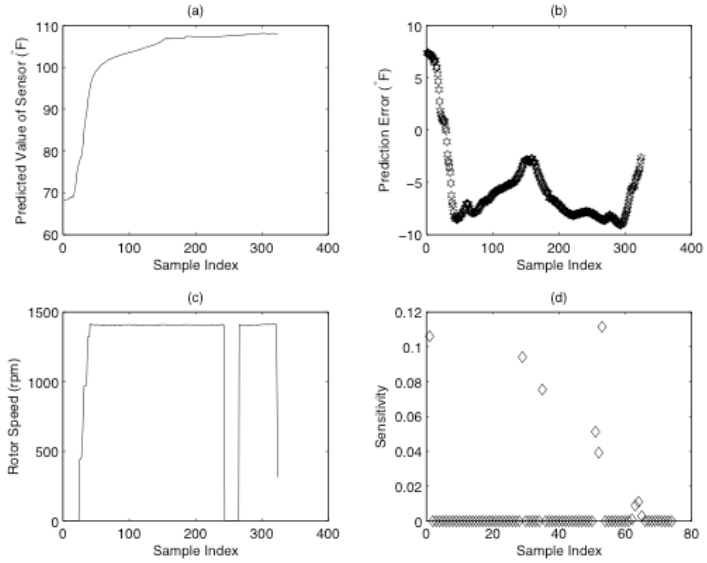

Fig. 10 SFA time series extrapolation of the critical sensor for Run 201; 788 bases were used for 14 recommended inputs: (a) predicted temperatures, (b) prediction error, (c) rotor speed, (d) input sensitivity.

4. Conclusions

An adaptive and matrix-free scheme has been developed for interpolating and approximating sparse multi-dimensional scattered data. The scheme requires neither ad-hoc parameters for the user to tune, nor rescaling of the inputs. This has been applied to a time series problem in a rotorcraft health monitoring system acquiring data from a scale model of the V-22 Osprey. The SFA method is based on a sequential Galerkin approach to artificial neural networks and is linear in storage with respect to the number of samples, s. Using this method an accurate approximation is built by incremental additions of optimal local basis functions. Matrix construction and evaluations are avoided. Computational cost is reduced while efficiency is enhanced by the low-dimensional unconstrained optimization. Theoretically, the method should be applicable to a number of popular local basis functions.

The SFA method has been used to model the time series data from the right swashplate bearing J-type temperature sensor recorded by the FS-TRAM HMS as a function of 74 system health parameters. We applied the method to data acquired by the FS-TRAM HMS from 8 test runs in NASA Ames Research Center's 80 x 120 ft wind-tunnel. The constructed SFA regression model using radial basis functions has been used satisfactorily to evaluate the sensitivity of the temperature time series from the critical sensor to the 74 system health parameters. Exponential convergence that sidesteps the ``curse of dimensionality'' was demonstrated for the temperature time series data used in the analysis.

Using the same data, we also constructed a SFA model using data from 7 test runs and extrapolated the model to approximate the remaining test run. The extrapolation results in black-box mode were unsatisfactory and only partially successful with the use of expert knowledge, which would probably not be available *a priori*. Future work will investigate improving the extrapolation by merging mathematical models with the experimental data through regularization. Future investigations will also include more sophisticated optimization routines, the reformulation of Eqs. (8) and (9) to decrease their sensitivity to initial values of σ_n and the use of various basis functions such as low-order polynomials, B-splines, and Gaussian tensors.

Acknowledgments

Special thanks to the NASA Ames Research Center Code ARA staff: Mr. José Navarrete, Mr. Michael Derby, Mr. Jeff Johnson, and Dr. Gloria Yamauchi for their considerable help. The author would also like to thank the reviewers for their helpful suggestions. Support for this work was provided by the NASA Ames Research Center grant NCC-2-8077 and NASA Cooperative Agreement No. NCC-1-02038.

References

[1]Meade, A.J., Kokkolaras, M., and Zeldin, B.A., "Sequential Function Approximation for The Solution of Differential Equations," *Communications in Numerical Methods in Engineering*, Vol. 13, 1997, pp. 977–986.

[2]Johnson, J. L., and Young, L. A., "Tilt Rotor Aeroacoustic Model Project," Confederation of European Aerospace Societies (CEAS) Forum on Aeroacoustics of Rotorcraft and Propellers, Rome, Italy, June 1999.

[3]Cybenko, G., "Approximations by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems,* Vol. 2, 1989, pp. 303–314.

[4]Hornik, K., Stinchcombe, M., and White, H., "Multi-Layer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[5]Rumelhart, D., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986, pp. 318–362.

[6]Saarinen, S., Bramley, R., and Cybenko, G., "Ill-Conditioning in Neural Network Training Problems," *SIAM Journal on Scientific Computing*, Vol. 14, No. 3, 1993, pp. 693–714.

[7]Jones, L. K., "Constructive Approximations for Neural Networks by Sigmoidal Functions," *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1586–1589.

[8]Jones, L. K., "A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates For Projection Pursuit Regression and Neural Network Training," *The Annals of Statistics*, Vol. 20, No. 1, 1992, pp. 608–613.

[9]Barron, A. R., "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Transactions on Information Theory*, Vol. 39, No. 3, 1993, pp. 930–945.

[10]Orr, M. J., "Regularization in the Selection of Radial Basis Function Centres," *Neural Computation*, Vol. 7, No. 3, 1995, pp. 606–623.

[11]Platt, J., "A Resource-Allocating Network for Function Interpolation," *Neural Computation*, Vol. 3, 1991, pp. 213–225.

[12]Fletcher, C.A.J., *Computational Galerkin Methods*, Springer-Verlag, New York, 1984.

[13]Kokkolaras, M., Meade, A. J., and Zeldin, B., "Concurrent Implementation of the Optimal Incremental Approximation Method for the Adaptive and Meshless Solution of Differential Equations," *Optimization and Engineering*, Vol. 4, No. 4, 2003, pp. 271–289.

[14]Thomson, D. L., Meade, A. J., and Bayazitoglu, Y., "Solution of the Radiative Transfer Equation in Discrete Ordinate Form by Sequential Function Approximation," *International Journal of Thermal Sciences*, Vol. 40, No. 6, 2001, pp. 517–527.

[15]Thomson, D. L., "Sequential Function Approximation of the Radiative Transfer Equation," Ph.D. Dissertation, Rice University, Houston, TX, 1999.

[16]Meade, A. J., and Zeldin, B., "Approximation Properties of Local Bases Assembled From Neural Network Transfer Functions," *Mathematical and Computer Modelling*, Vol. 28, No. 9, 1998, pp. 43–62.

[17]Govindaraju, R. S., and Zhang, B., "Radial-Basis Function Networks," *Artificial Neural Networks in Hydrology*, Kluwer Academic Publishers, Netherlands, 2000, pp. 93–109.

[18]Smola, A. J., and Schölkopf, B., "A Tutorial on Support Vector Regression," Neuro-COLT2 Technical Report Series, NC2-TR-1998-030, Oct. 1998, p. 4.

[19]Collobert, R., and Bengio, S., "SVMTorch: Support Vector Machines for Large-Scale Regression Problems," *Journal of Machine Learning Research*, Vol. 1, 2001, pp. 143–160.

[20]Chapra, S. C., and Canale, R. P., *Numerical Methods for Engineers*, McGraw–Hill, New York, 2002, pp. 701–711.

[21]Poggio, T., and Girosi, F., "Networks for Approximation and Learning," *Proceedings of IEEE*, Vol. 78, No. 9, 1990, pp. 1481–1497.

[22]Zeldin, B., and Meade, A. J., "Integrating Experimental Data and Mathematical Models in the Simulation of Physical Systems," *AIAA Journal*, Vol. 35, No. 11, 1997, pp. 1787–1790.