JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 40, No. 1, January–February 2003

Adaptive State Filtering for Space Shuttle Main Engine Turbine Health Monitoring

Rube B. Williams Jr.*
Los Alamos National Laboratory, Los Alamos, New Mexico 87545
and
Alexander G. Parlos†
Texas A&M University, College Station, Texas 77843

Real-time estimation of system states or parameters that are difficult or expensive to measure directly is often needed for adaptive control or health monitoring purposes. A practical algorithm is proposed for adaptive state filtering in nonlinear dynamic systems when the state equations are unknown or too complex to model analytically. The state equations are constructively approximated by using recurrent neural networks. The proposed algorithm is based on the predictor-update approach of the Kalman filter, but a least-mean-square filter implementation with an adaptive filter gain is used. Furthermore, unlike the Kalman filter and its nonlinear extensions, the proposed algorithm makes minimal assumptions regarding the underlying nonlinear system dynamics and their noise statistics. The filter is used to estimate the high-pressure turbine discharge temperature of the space shuttle main engine, during setpoint changes and turbopump failures. The filter is developed by using simulated engine data, and its performance is tested on both simulated and actual recorded space shuttle main engine transients. When the complexity of the problem studied is considered, the resulting filter accuracy is shown to be quite acceptable. Further use of the adaptive filter gain developed is to enable real-time detection of certain system failures, such as the turbopump failures of the space shuttle main engine. This concept is demonstrated also by using both simulated and experimental failure data.

Nomenclature

$b_{[l,i]}$	=	neural node bias
D	=	closed domain defined by training set
$E\{\}$	=	expectation operator
E_1, E_2	=	objective functions for learning
$\boldsymbol{e}_q(\,)$	=	modeling error of $\boldsymbol{q}(\,)$
$F_{[l,i]}$	=	neural node discriminatory function
$\boldsymbol{f}(\,)$	=	nonlinear system dynamics function
$\boldsymbol{f}_{\mathrm{NN}}(\,)$	=	neural approximation of $\boldsymbol{f}(\,)$
$\boldsymbol{h}(\,)$	=	nonlinear system dynamics function
$\boldsymbol{h}_{\mathrm{NN}}(\,)$	=	neural approximation of $\boldsymbol{h}(\,)$
i, j	=	node indices
$\boldsymbol{J}_q(\,)$	=	gradient cost function
k	=	discrete time index
L	=	number of network layers
$L(\,)$	=	high-pressure turbine (HPT) work coefficient
$\hat{L}(\,)$	=	estimate of HPT work coefficient
l	=	number of estimated states
l, l'	=	layer indices
$M_R(\,)$	=	mixture ratio, ratio of oxidizer to fuel flow
$\dot{M}_F(\,)$	=	measured fuel flow rate, lbm/h
$\dot{\hat{M}}_F(\,)$	=	estimated fuel flow rate, lbm/h
$N(l)$	=	number of neural nodes for layer l
NP	=	number of training examples
n	=	number of outputs in training
n_q	=	dimension of vector $\boldsymbol{q}(\,)$
n_y	=	dimension of vector $\boldsymbol{y}(\,)$
$P_C(\,)$	=	measured main chamber pressure, psia
$\hat{P}_C(\,\vert\,)$	=	estimated main chamber pressure, psia
$\boldsymbol{q}(\,)$	=	system parameter vector
$\boldsymbol{q}_d(\,)$	=	deterministic component of $\boldsymbol{q}(\,)$
$\hat{\boldsymbol{q}}(\,\vert\,)$	=	deterministic estimate of $\boldsymbol{q}(\,)$
$\boldsymbol{R}_q(\,)$	=	adaptive filter gain matrix
$T_F(\,)$	=	high-pressure fuel turbopump (HPFTP) discharge temperature, °R
$\hat{T}_F(\,)$	=	estimated HPFTP discharge temperature, °R
$T_0(\,)$	=	high-pressure oxidizer turbopump (HPOTP) discharge temperature, °R
$\hat{T}_0(\,)$	=	estimated HPOTP discharge temperature, °R
$\boldsymbol{u}(\,)$	=	system input vector, normalized
u_{FPOV}	=	fuel preburner oxidizer valve position, normalized
u_{OPOV}	=	oxygen preburner oxidizer valve position, normalized
$V(\,\vert\,)$	=	filter objective function
$\boldsymbol{v}(\,)$	=	parameter noise vector
$\boldsymbol{w}(\,)$	=	output noise vector
$w_{[l,j][l',i]}$	=	neural node weight
$\boldsymbol{x}(\,)$	=	system state vector
$x_{[l,i]}(\,)$	=	neural node output
$\hat{\boldsymbol{x}}(\,\vert\,)$	=	system state vector estimate
$\hat{\boldsymbol{x}}_{\mathrm{NN}}(\,\vert\,)$	=	neural state vector estimate
$\boldsymbol{y}(\,)$	=	system output vector
$\hat{\boldsymbol{y}}(\,\vert\,)$	=	system output vector estimate
$\hat{\boldsymbol{y}}_{\mathrm{NN}}(\,\vert\,)$	=	neural output vector estimate
$z_{[l,i]}(\,)$	=	neural node state
$\hat{\alpha}(k \mid k)$	=	filtered value of any $\alpha(k)$
$\hat{\alpha}(k+1 \mid k)$	=	predicted value of any $\alpha(k)$
$\hat{\alpha}(k \mid k+1)$	=	smoothed value of any $\alpha(k)$
$\Delta\boldsymbol{q}(\,)$	=	estimation error for parameter $\boldsymbol{q}(\,)$
$\varepsilon(\,\vert\,)$	=	innovations term
τ	=	filter window size, s

Received 1 February 2002; revision received 23 September 2002; accepted for publication 16 October 2002.

*Technical Staff Member, Nuclear Systems Design and Analysis, Mail Stop K575, P.O. 1663; rbwx@lanl.gov. Member AIAA.

†Associate Professor, Department of Mechanical Engineering, 116 Engineering/Physics Building, Mail Stop 3123; a-parlos@tamu.edu. Senior Member AIAA.

Introduction

TYPICALLY, to mitigate performance degradation resulting from subsystem failures in real-world systems, control and health monitoring approaches require the availability of system

state information that is impractical to measure directly.[1,2] Ideally, for such problems, system state information can be extracted from measured outputs by using an adaptive state filter. However, many time-varying problems of aerospace interest involve systems that operate in unstable regimes and near stress limits, where rapid estimator adaptation is critical for safe operations.[3–5] For such problems, particularly where the relevant state equations are poorly understood, or are too complex for analytical expression, suitable adaptive filters are practically nonexistent.

A conventional way of solving a nonlinear state-estimation problem involving a time-varying dynamic system is to forgo an optimal solution and to assume an estimator structure of suitably reduced order that performs satisfactorily for the given problem and solution method. For mildly nonlinear problems, linearly parametrized model structures and recursive least-mean-square (LMS) or recursive-least-square parameter adaptation methods can be used.[6,7] For such solutions, the adaptation law does not seek the "best" model of the target system, but the best of available models that are forthcoming from the assumed model structure. Such state estimators are sometimes referred to as restricted complexity adaptive filters[6,8] and as a class include adaptive implementations of the extended Kalman filter (EKF).[9,10]

To extend this general approach to highly nonlinear dynamic problems, it is somewhat natural to consider neural networks (NNs) as candidate model structures. Such highly flexible nonlinear input–output mappings have in the last decade demonstrated a high level of success in identifying the underlying deterministic behavior of highly nonlinear stochastic dynamic systems[11–16] and also as near-optimal nonlinear state filters,[17–22] albeit for systems that are time invariant or that are nearly so. NNs are considered effective universal approximation tools for nonlinear systems. Nevertheless, the currently available NN learning algorithms are generally too slow to deal effectively with real-time adaptive estimation and control problems encountered in practice requiring rapid convergence. This necessitates pretraining of NNs, with only incremental tuning performed online.

Nevertheless, for many engineered systems, there is generally much information available offline about the system's deterministic behaviors, including those behaviors resulting from subsystem degradation and failure. Such information can be acquired from some combination of prototypes, especially instrumented tests, first-principle models, and other means. Therefore, the potential contribution of NNs in real-time adaptive state-estimation problems may be their ability to capture available deterministic information, of various complexities, into compact model structures.

The objective of this paper is to present a practical closed-loop adaptive state filtering algorithm, suitable for a certain class of highly nonlinear time-varying problems, where relatively rapid adaptation to system failure is required, and the system state equations are unknown.[23] The proposed filter is also demonstrated to provide a certain disturbance detection capability that can directly contribute to system fault detection. The adaptive state filtering algorithm proposed is based on the predictor-update approach of the Kalman[9] filter and requires the prior identification of the deterministic part of a reduced-order system model structure using dynamic NNs. The identified system model structure forms the predictor part of the filter. In the update part of the filter, the identified model structure is incorporated into a recursive stochastic gradient approach, along the lines of conventional LMS filtering. The proposed filtering algorithm is a follow-on research to previous work in Refs. 21 and 22 where Parlos et al. demonstrated that adaptive state filters constructed using recurrent NNs could, in some instances, outperform the EKF. In the present study, an LMS filter formulation is pursued instead of the minimum-variance formulation presented in Ref. 21.

The effectiveness of the proposed adaptive filtering algorithm is demonstrated by using a case study of space shuttle main engine (SSME) transient state estimation. This case study follows, in some sense, the contributions of the authors of Refs. 1, 4, 16, and 24–26, where researchers utilized the SSME as a testbed for developing estimation methods that can potentially contribute to fault-tolerant control operations for reusable rocket engines. In the presented study, the discharge temperatures of the SSME turbines are estimated during normal and abnormal operating conditions using both simulated and actual SSME output measurements. Additionally, the utility of the proposed adaptive filter gain for detecting SSME engine faults is briefly presented utilizing both simulated and actual SSME measurements. Because of space limitations, this presentation is limited to a couple of case studies.

The presentation is organized as follows. The adaptive state filtering problem of interest is defined, and the equations used in the proposed adaptive filtering algorithm are presented. The NN learning algorithms used in the development of the filter predictors are discussed, and the adaptive state filter is applied for estimation of the SSME turbine discharge temperatures during normal operating conditions and during turbine failures. Finally, the results of this application are discussed.

Nonlinear State Filtering

Problem Statement

Consider the following system representation in discrete-time, nonlinear, stochastic state-space form, also known as the noise representation:

$$\boldsymbol{x}(k+1) = \boldsymbol{f}[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{q}(k)], \qquad \boldsymbol{y}(k) = \boldsymbol{h}[\boldsymbol{x}(k)] + \boldsymbol{w}(k)$$
$$\boldsymbol{q}(k) = \boldsymbol{q}_d(k) + \boldsymbol{v}(k) \tag{1}$$

It is assumed that $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ are independent white noise vector processes. It is assumed that all vectors involved in Eq. (1) are of appropriate dimensions.

At this stage of the problem definition, no assumptions are made regarding implicit or explicit knowledge of $\boldsymbol{f}(\,)$ and $\boldsymbol{h}(\,)$, other than that such a representation is an accurate description of the underlying system dynamics. In many traditional state filtering approaches, exact knowledge of Eq. (1) is assumed, and the noise vectors $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ are considered zero-mean, white Gaussian processes. Equation (1) becomes the starting point for the development presented herein. An assumption made regarding the system of Eqs. (1) is that

$$\left| E\left(\frac{\Delta y_i(k)}{\Delta q_j(k)} \right) \right| > 0, \qquad \forall k, \qquad n_q < i \le n_y, \qquad 1 \le j \le n_q \tag{2}$$

Equation (2) requires that the deterministic component of each measured output varies monotonically with each element of $\boldsymbol{q}(k)$.

The objective of the state filtering problem is to obtain the system state estimate $\hat{\boldsymbol{x}}(k)$ for $\boldsymbol{x}(k)$, given $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$. In linear state filtering, the notation used to denote this estimate is important because, depending on the chosen filtering method, different optimal state estimates are obtained. However, in nonlinear state filtering problems, the resulting state estimates are not optimal in any sense. Therefore, the notation $\hat{\boldsymbol{x}}(k \mid k)$ is used herein to mean simply the state estimate at the discrete time k, following the update resulting from the measurement $\boldsymbol{y}(k)$, at time k.

Conventional Method of Solution

The EKF solution to nonlinear state filtering assumes the availability of an exact system model shown by Eq. (1) (Ref. 27). The algorithm consists of the following two steps. The first step is the prediction step, which is performed before observing the $(k+1)$th sample. The assumed model $\boldsymbol{f}(\,)$ and $\boldsymbol{h}(\,)$ and the filtered state estimate $\hat{\boldsymbol{x}}(k \mid k)$ are used to compute the state and output predictions $\hat{\boldsymbol{x}}(k+1 \mid k)$ and $\hat{\boldsymbol{y}}(k+1 \mid k)$. The a priori error covariance matrix is also computed by using the state estimate and the Jacobian of $\boldsymbol{f}(\,)$, evaluated at the state estimate. The assumed covariance matrix of the process noise is used in this step. The second step is referred to herein as the update step. This step is performed following observation of the $(k+1)$th sample. The state prediction $\hat{\boldsymbol{x}}(k+1 \mid k)$ is updated with a linear combination of $\varepsilon(k+1 \mid k)$, resulting in $\hat{\boldsymbol{x}}(k+1 \mid k+1)$. The coefficients used to weigh the innovation terms form the elements of the EKF gain matrix. The EKF gain matrix is updated by using the Jacobian of $\boldsymbol{h}(\,)$, evaluated at the $\hat{\boldsymbol{x}}(k+1 \mid k)$,

the a priori error covariance matrix, and the assumed covariance matrix of the measurement noise. Finally, the a posteriori error covariance matrix is computed by using the updated EKF gain matrix, the a priori error covariance matrix, and the Jacobian of $\boldsymbol{h}(\,)$ evaluated at the $\hat{\boldsymbol{x}}(k+1\,|\,k)$. In this formulation of the EKF, state estimate $\hat{\boldsymbol{x}}(k\,|\,k)$ could be augmented to include a vector such as $\hat{\boldsymbol{q}}(k\,|\,k)$.

Proposed Method of Solution

There are three significant differences between the proposed filter and the conventional EKF solution. First, the nonlinear functions $\boldsymbol{f}(\,)$, and $\boldsymbol{h}(\,)$ are assumed to be unknown, and they are replaced by empirical NN approximations constructed from input–output measurements. Second, an LMS filter is formulated and solved, rather than a minimum variance filter. Third, noise statistics are assumed to be unknown and are not explicitly used in the filter computations. The NN architecture considered in this paper is that of a dynamic recurrent multilayer perceptron (RMLP).[13] The approximation properties of the static feedforward multilayer perceptron NNs are well documented in the literature.[28–31] Similar approximation properties have not been proven in the literature about the recurrent form of this NN, either in static or dynamic configuration because of the complexity of the underlying NN structure. Despite the lack of rigorous results for recurrent NNs, their utility in approximating dynamic systems is explored in this paper.

Adaptive State Filter Formulation

Filter Equations

In this section, no system model is assumed known for the predictor step of the filter. Rather, a finite set of input, output, state, and parameter data is assumed to be available. Therefore, NNs are used to develop approximations of the functional forms entering the filter equations. The adaptive state filter for $\hat{\boldsymbol{x}}(k+1\,|\,k+1)$ is formulated by using the following two steps. The first step, the prediction step, occurs before observing the $(k+1)$th sample. The state and output predictor values are computed using the following equations:

$$\hat{\boldsymbol{x}}_{\mathrm{NN}}(k+1\,|\,k) = \boldsymbol{f}_{\mathrm{NN}}[\hat{\boldsymbol{x}}_{\mathrm{NN}}(k\,|\,k), \boldsymbol{u}(k), \boldsymbol{y}(k), \hat{\boldsymbol{q}}(k\,|\,k)]$$

$$\hat{\boldsymbol{y}}_{\mathrm{NN}}(k+1\,|\,k) = \boldsymbol{h}_{\mathrm{NN}}[\hat{\boldsymbol{x}}_{\mathrm{NN}}(k+1\,|\,k), \hat{\boldsymbol{q}}(k\,|\,k)] \tag{3}$$

The second step is an update step that is conducted following observation of the $(k+1)$th sample. The updated state and parameter estimates are computed using the following equations:

$$\hat{\boldsymbol{q}}(k\,|\,k+1) = \hat{\boldsymbol{q}}(k\,|\,k) + \boldsymbol{R}_q(k+1\,|\,k+1)\boldsymbol{J}_q(k+1\,|\,k+1)$$

$$\hat{\boldsymbol{x}}_{\mathrm{NN}}(k+1\,|\,k+1) = \boldsymbol{f}_{\mathrm{NN}}[\hat{\boldsymbol{x}}_{\mathrm{NN}}(k\,|\,k), \boldsymbol{u}(k), \boldsymbol{y}(k), \hat{\boldsymbol{q}}(k\,|\,k+1)]$$

$$\hat{\boldsymbol{q}}(k+1\,|\,k+1) = \hat{\boldsymbol{q}}(k\,|\,k+1) \tag{4}$$

Here, $\hat{\boldsymbol{q}}(k\,|\,k+1)$ is an updated estimate of the parameter estimate at k that uses the sensor information obtained at time $k+1$ and is effectively a smoothed update of $\hat{\boldsymbol{q}}(k\,|\,k)$. In the second of Eqs. (4), the $\hat{\boldsymbol{q}}(k\,|\,k+1)$ is used in the predictor $\boldsymbol{f}_{\mathrm{NN}}(\,)$, replacing $\hat{\boldsymbol{q}}(k\,|\,k)$ that is used in the predictor step of the filter. This substitution results in the computation of $\hat{\boldsymbol{x}}(k+1\,|\,k+1)$, which contains information from the innovations at $k+1$. Finally, in the third of Eqs. (4), it is assumed that $\hat{\boldsymbol{q}}(k\,|\,k+1)$ is equal to $\hat{\boldsymbol{q}}(k+1\,|\,k+1)$. This assumptions is a good approximation when the trend of the parameter vector $\boldsymbol{q}(\,)$ is slowly varying.

The variables $\boldsymbol{J}_q(\,)$ and $\boldsymbol{R}_q(\,)$ are defined as

$$\varepsilon(k\,|\,k) = \boldsymbol{y}(k) - \hat{\boldsymbol{y}}(k\,|\,k-1), \qquad V(k\,|\,k) = \frac{1}{2}\varepsilon(k\,|\,k)^T\varepsilon(k\,|\,k)$$

$$\boldsymbol{J}_q(k\,|\,k) = -\left[\frac{\partial V(k\,|\,k)}{\partial \hat{q}_1(k-1\,|\,k-1)} \quad \frac{\partial V(k\,|\,k)}{\partial \hat{q}_2(k-1\,|\,k-1)} \cdots \frac{\partial V(k\,|\,k)}{\partial \hat{q}_{n_q}(k-1\,|\,k-1)}\right]^T \tag{5}$$

$$R_{qi}(k\,|\,k) = \frac{\left|\sum_{s=k-\tau}^{k} J_{qi}(s\,|\,s)J_{qi}(s-1\,|\,s-1)\right|}{\sum_{s=k-\tau}^{k} J_{qi}^2(s\,|\,s)}$$

$$\xrightarrow{\tau \text{ large}} \frac{|E\{J_{qi}(k\,|\,k)J_{qi}(k-1\,|\,k-1)\}|}{E\left\{J_{qi}^2(k\,|\,k)\right\}}$$

$$\boldsymbol{R}_q(k\,|\,k) = \mathrm{diagonal}\left(R_{q1}, R_{q2}, \ldots, R_{qn_q}\right) \tag{6}$$

$R_{qi}(k\,|\,k)$ is the noise-to-signal ratio of $J_{qi}(k\,|\,k)$, where τ is based on prior experience and it is selected to be sufficiently large. Calculation of $\boldsymbol{J}_q(k\,|\,k)$ is performed analytically for the selected RMLP NN models, by using the gradients of these networks as defined in the respective backpropagation learning algorithms.[13,15,21] However, because there can be no contributions from network internal recurrent links to the calculation of $\boldsymbol{J}_q(k\,|\,k)$, as defined by Eq. (5), the analytical expressions for this gradient cost function, in all cases, reduce to a backpropagation along the feedforward links.

Comments on the Neural Predictors

The networks $\boldsymbol{f}_{\mathrm{NN}}(\,)$ and $\boldsymbol{h}_{\mathrm{NN}}(\,)$ form a model structure that is parameterized in $\hat{\boldsymbol{q}}(\,)$. However, the terms model and model structure are interchangeably used concerning $\boldsymbol{f}_{\mathrm{NN}}(\,)$ and $\boldsymbol{h}_{\mathrm{NN}}(\,)$. NNs do not extrapolate well, and as a result, the identified models $\boldsymbol{f}_{\mathrm{NN}}(\,)$ and $\mathrm{h}_{\mathrm{NN}}(\,)$ are only defined for $\hat{\boldsymbol{q}}(\,) \in D$.

Also, because of its explicit dependence on $\hat{\boldsymbol{q}}(k\,|\,k)$, $\boldsymbol{h}_{\mathrm{NN}}(\,)$ is not exactly analogous to $\boldsymbol{h}(\,)$, as it is defined in Eq. (1). Because the deterministic component of the parameter $\boldsymbol{q}(\,)$ is typically much more slowly varying than the other dynamic signals used in identifying the NN predictor models, neural forgetting of the implicit dependence of $\boldsymbol{h}(\,)$ on past values of the parameter $\boldsymbol{q}(\,)$ can limit the success of the available training data in some cases. To improve the training of the NN predictors, an explicit dependence on $\hat{\boldsymbol{q}}(k\,|\,k)$ is utilized in $\boldsymbol{h}_{\mathrm{NN}}(\,)$. Furthermore, $\boldsymbol{f}_{\mathrm{NN}}(\,)$ contains measured system outputs as inputs, and it is not analogous to $\boldsymbol{f}(\,)$, as it is defined in Eq. (1). For certain cases, the state-space information available for training the networks is too incomplete for accurate predictor identification, and the use of input–output data as network inputs can improve predictor training. Additionally, certain defined system states may also be sensed outputs in some applications, and these sensed values should be used as inputs to the NN predictors during such online filtering operations. For the special case, when only output predictions are desirable in the first step of the filtering algorithm, the composite functional $\boldsymbol{h}_{\mathrm{NN}}[\boldsymbol{f}_{\mathrm{NN}}(\,)]$ can be identified without impacting the accuracy of the filtering algorithm.

Asymptotic Behavior of the Adaptive Filter Gain

The parameter $\boldsymbol{q}(k)$ can be defined in terms of $\hat{\boldsymbol{q}}(k\,|\,k)$ and $\boldsymbol{e}_q(k)$ as

$$\boldsymbol{q}(k) = \hat{\boldsymbol{q}}(k\,|\,k) + \boldsymbol{e}_q(k) + \boldsymbol{v}(k) \tag{7}$$

Then $\Delta\boldsymbol{q}(k)$ is expressed as

$$\Delta\boldsymbol{q}(k) = \boldsymbol{q}(k) - \hat{\boldsymbol{q}}(k\,|\,k) = \boldsymbol{e}_q(k) + \boldsymbol{v}(k) \tag{8}$$

However, for sufficiently small values of $\varepsilon(k+1)$, the gradient-descent update approximation can be utilized, that is,

$$\Delta\boldsymbol{q}(k) = \boldsymbol{J}_q(k+1\,|\,k+1) \tag{9}$$

which, for this case, directly relates the gradient cost of the prediction error at $k+1$ to the size of the parameter estimation error at k. By substitution of Eq. (8) into Eq. (9), and then into Eq. (6), it follows that for a sufficiently small sampling interval

$$\boldsymbol{R}_q(k+1\,|\,k+1) \to \boldsymbol{I} \quad \text{as} \quad \Delta\boldsymbol{q}(k) \to \boldsymbol{e}_q(k)$$

$$\boldsymbol{R}_q(k+1\,|\,k+1) \to 0 \quad \text{as} \quad \Delta\boldsymbol{q}(k) \to \boldsymbol{v}(k) \tag{10}$$

These asymptotic expressions for the filter gain express the obvious and desirable fact that, as the parameter error estimates approach a white noise process, the filter gain becomes infinitesimally small.

Comments on Filter Convergence

Offline development of the filter predictors, described later, establishes the existence of a slowly varying sequence of parameters, $\{\hat{\boldsymbol{q}}(k) \in D\}$, and a small associated state and output prediction error, for a certain operating range of a given target system. It is inherently assumed in the filter development that the monotonic dependence of the measured outputs on $\hat{\boldsymbol{q}}(k)$ is sufficiently learned by the identified predictor, such that small prediction errors define a global region of attraction for the adaptive filtering algorithm. This region of attraction is valid for the operating conditions defined by the training set used in filter development and by the adaptation problem constraint $n_q < n_y$, that is, the adaptation problem has more output conditions than adaptable parameters. Furthermore, the treatment of the measured outputs and system states is much in the same manner during training, that is, that some state values are available as targets during training guarantees that low output prediction errors associated with low state filtering errors are obtained.

During online filter operation, only the output prediction errors can be calculated because state observations are not available. Because of the requirement $n_q < n_y$, the adaptive filtering algorithm is overspecified. As a result, on an unknown target system a parameter sequence for $\hat{\boldsymbol{q}}(k)$ that is associated with small output prediction errors is not guaranteed to exist. However, it may be possible to establish a small enough prediction error criterion during offline predictor identification, such that online prediction errors satisfying this error criterion will, with good likelihood, imply that the target system is a realization of the system used to train the NNs. In such a case, the state estimation errors can also be expected to be suitably small. This latter filter property, though it cannot be guaranteed, is associated with filter convergence, and it requires that certain observability assumptions be placed on the system under investigation. Such assumptions regarding the system of Eq. (1) are not explicitly stated in this work because it is exceedingly difficult to ascertain observability arguments about a dynamic system from input-state–output measurements.

Inspection and trial-and-error iterations are utilized in the approach presented to design and implement an adaptive filter based on the proposed algorithm. Unfortunately, little can be proven, in general, concerning convergence of the identified model to the target system and convergence of the adaptive filtering algorithm. Such difficulty in convergence analysis is primarily due to the complications arising from the adaptive nature of the filter and the use of nonlinear models as filter predictors. Therefore, substantial subjective responsibility is placed on the filter designer when using the proposed filtering algorithm.

NN Predictor Identification

Identification of a NN predictor is accomplished via a sequence of steps that includes iterative learning.[11,15] The data used for identification, sometimes called the training data set, typically comprise an estimation data set that is concatenated with a validation data set, where both data sets are rich and independently generated. The network parameters, or weights, are typically adjusted by using a gradient-descent algorithm such that prediction errors on the estimation data set are minimized. Prediction errors on the validation data set are also generated but are not used for weight update. Generally, for a number of iterations, both the estimation and validation errors will decrease. Eventually, however, prediction errors will start increasing on the validation data set, indicating overtraining, that is, network learning of information in the training data that is not general to the underlying system behavior. The network weights calculated just before overtraining begins are considered the best weights for the given network. Candidate networks are determined from experience and trial and error, and the network that ultimately produces the lowest prediction errors on the validation data set is considered the best system model. Normally, following predictor identification, one or several independently generated testing data sets are generated and processed by the best NN model to check the effectiveness of the learning on independent data sets and to benchmark the identified NN model against characteristic target system behaviors.

The NN used in this study to approximate the filter predictors is the RMLP network, and it is based on prior work presented in Ref. 11. The equations describing the ith node in the lth layer of the network are as follows:

$$z_{[l,i]}(k+1) = \sum_{j=1}^{N(l)} w_{[l,j][l,i]} x_{[l,i]}(k) + \sum_{j=1}^{N(l-1)} w_{[l-1,j][l,i]} x_{[l-1,j]}(k+1) + b_{[l,i]} \tag{11}$$

$$x_{[l,i]}(k) = F_{[l,i]}\left[z_{[l,i]}(k)\right] \tag{12}$$

where $i = 1, \ldots, N(l)$ and $l = 1, \ldots, L$. For the output layer, $l = L$, the discriminatory function is assumed linear. The first summation of Eq. (11) includes the recurrent network weights, whereas the second summation includes the feedforward weights.

The RMLP network can be simplified to three layers, an input layer that acts as a buffer, a nonlinear hidden layer, and a linear output layer. These simplified RMLP network equations can be expressed in the form of an empirical input–output predictor in state-space form.[13] The input vector of this predictor formulation can be modified for use in the learning mode called teacher forcing (TF), where no past predictor outputs are used in the input layer. This predictor formulation is also referred as a static NN. The same predictor can be used for the learning mode, called global feedback (GF), where a number of tapped delayed predictor outputs are used in the input layer of the network. The resulting network is referred to as a dynamic NN, and it is constructed by appropriately modifying the predictor's input vector.[18] Further details regarding the implications of using TF vs GF on the learning algorithms are presented Ref. 15.

Application to SSME

In this section, a case study of the proposed adaptive filtering method is presented. The proposed algorithm is used to estimate the SSME high-pressure fuel turbine (HPFT) and high-pressure oxidizer turbine (HPOT) discharge temperatures, during normal operations and during high-pressure fuel turbopump (HPFTP) failures. The objective of this case study is to demonstrate the general performance attributes of the adaptive state filter for a practical, complex problem. NN predictors are identified from simulated data, and the designed adaptive filter is tested using both simulated and actual experimental SSME data during failure conditions. An overview of the proposed adaptive filter for the SSME case study is presented in Fig. 1.

SSME Description

The SSME is a liquid-propellant rocket engine. Fuel and oxidizer flows, driven by four turbopumps, combine in dual preburners and in the main combustion chamber to produce high-temperature gas that is expanded through the nozzle to produce thrust. The engine has five control valves and utilizes open-loop control for startup and shutdown and linear closed-loop control for the so-called mainstage operation. During mainstage operation, the engine power level is regulated by two control loops. These control loops manipulate two control valves, the fuel preburner oxidizer valve (FPOV) and the oxygen preburner oxidizer valve (OPOV), which throttle the fuel and oxidizer turbopump power and the corresponding propellant flow rates, respectively. One control loop regulates the main chamber pressure P_C, by positioning both the FPOV and the OPOV to maintain a pressure setpoint. The other control loop trims the FPOV position to cause a calculated mixture ratio M_R to track a setpoint. The M_R is calculated by the control system as a linear regression of the measured fuel flow $\dot{M}_F$ and P_C. Together, P_C and the M_R determine the engine power output. The other control valves are configured during engine startup to positions that correspond to the maximum engine performance requirements for the mission.[32]

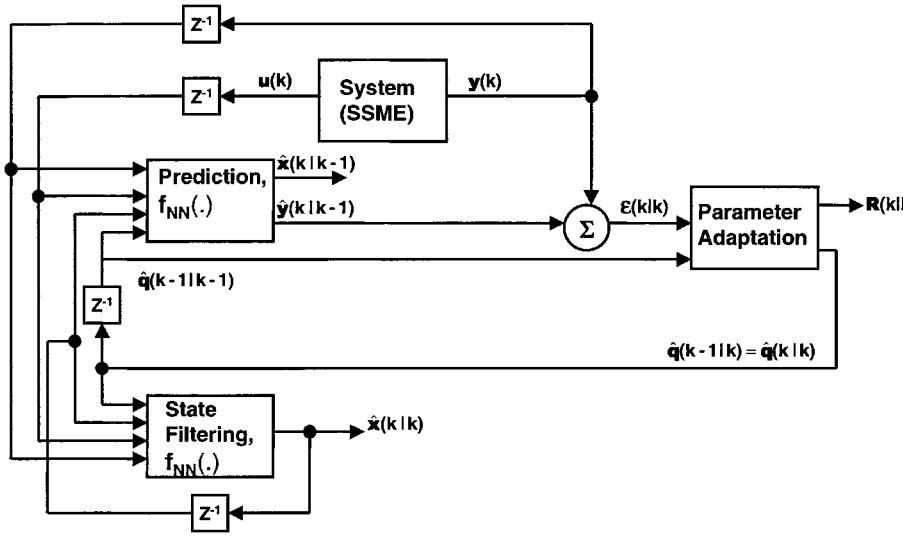

Fig. 1 SSME adaptive state filter block diagram.

SSME Test Data Description

Actual SSME test data are used for validation of the SSME simulator that is used in filter development and for validation of the designed adaptive filter. The test data are from a SSME test firing, that is, test 901-853 of SSME 0523 on stand A1 at the Stennis Space Center (SSC) in January 1996. The test was scheduled to run for 750 s, but test engineers shut down the engine after operating it at, or near, full power for about 553 s. The premature shutdown was due to metallic evidence of turbine failure in the engine exhaust plume.

In this test, an SSME was operated at a constant M_R of 6.01, while the P_C was varied. The engine was initialized at 100% power level at engine startup, throttled from 100% power down to 90% power after 40 s, then back up to 100% power in 25 s, or in 65 s from engine startup. At 90 s from startup, the engine was taken to 104% power, and then up to 109% power at 115 s from startup. At 140 s from startup, it was throttled down to 104% power and held at 104% power level until engine shutdown at 553 s following startup. Posttest analysis conducted by Marshall Space Flight Center and SSC engineers, who primarily evaluated vibration data and conducted visual inspections, revealed that significant degradation events had occurred in the HPFT blades at 130, 276, 404, and 415 s following startup, and also a gradual degradation of turbine condition took place between 520 s and engine shutdown.[2,23] The experimental data were collected at a rate of 25 samples/s, and they included observations from the initiation of mainstage operations. Data for all thermal and fluid variables of interest were collected throughout the test. Further operational details regarding the test data may be found in Refs. 2 and 23.

SSME First-Principles Model and Its Validation

The SSME digital transient model (DTM) is an SSME simulator that was used extensively in this research. This simulator is a detailed first-principles model of the thermal–fluid dynamics and control of the SSME.[31] DTM was developed at Rocketdyne, and it has been used extensively to analyze the performance of the SSME. The model consists of a large number of coupled nonlinear differential equations with appropriate time-varying coefficients. Correlations from a variety of tested engines running under different conditions are used to define the scope and practical limitations of the various equations. Finite difference techniques are used when an SSME subsystem warrants a distributed parameter model. For a complete description of the DTM, see the relevant model documentation.[32]

Nominally, the SSME DTM contains sufficient modeling sophistication to approximate the dynamic behavior of a particular engine to a high degree of accuracy. Tuning of the DTM is accomplished by adjusting some of its coefficients, as required, to minimize the error between the mathematical model response and the test data for a particular engine. In this study, the DTM was validated against the SSC test data by comparing its transient response during the first 175 s of the test data. All simulated SSME variables, with the exception of the high-pressure turbine (HPT) discharge temperatures, compared quite well with the test data, and maximum relative errors of approximately 3% were obtained. The HPT discharge temperatures had maximum relative errors of approximately 10%. Further tuning of the DTM could have eliminated these discrepancies. However, it was desirable to maintain some discrepancies between the DTM response and the SSC test data, allowing verification of the adaptive state filter on a process with some level of dynamic uncertainty. Extensive transient response comparisons of the DTM responses and the SSC test data may be found in Ref. 23.

Adaptive Filter Development

NN predictor models for certain SSME processes are identified by using transient data generated from the SSME DTM. Any number of dynamic states and parameters can be simulated with the DTM. Experience, existing literature,[16,24,32] and trial-and-error iterations were used to determine the NN architectures and model structures that are suitable for the HPT discharge temperature estimation problem being studied. It was determined that HPT discharge temperatures can be predicted by using different combinations of past and current values of various SSME variables. These variables include u_{FPOV}, u_{OPOV}, T_F, T_0, P_C, and $\dot{M}_F$. Because P_C and $\dot{M}_F$ are nominally measured outputs that are used by the SSME control system, these variables were selected as the measured outputs for the filtering application.

Only two measured outputs are available for this problem. As a result, only one adaptable parameter, $q(k) = L(k)$, is considered for consistency with the filtering problem formulation assumptions [see Eq. (2)]. In this case study, it is known a priori that the target system will experience HPFTP failures, although the timing and the severity of the failures are assumed unknown. In the training data set, the $L(k)$ is set equal to a coefficient in the SSME DTM model representative of the HPT work. This coefficient is adjusted to model various off-nominal turbine performance conditions. HPT efficiency degradation of greater than 5% of the nominal value is normally associated with catastrophic failures.[23]

Training Set Selection

The training data set consists of an estimation set used in updating the network weights and biases and a validation data set used in terminating the learning and in selecting the best network architecture. The identification data set is defined such that $L(k)$ takes the discrete values of 1.05, 1.0, 0.95, 0.9, and 0.85, corresponding to SSME DTM HPT work outputs of 105, 100, 95, 90, and 85%, respectively, of the value corresponding to nominal SSME conditions at 100% engine power level. The range of values used for $L(k)$

Table 1 Identified neural network models of SSME processes

Functional description	NN architecture
$\hat{P}_C(k+1) = f_{\text{NN1}}[\hat{P}_C(k), \hat{\dot{M}}_F(k), \hat{T}_F(k), \hat{T}_0(k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), L(k)]$	RMLP, 7–4–1
$\hat{\dot{M}}_F(k+1) = f_{\text{NN2}}[\hat{P}_C(k), \hat{\dot{M}}_F(k), \hat{T}_F(k), \hat{T}_0(k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), L(k)]$	RMLP, 7–2–1
$\hat{T}_F(k+1) = f_{\text{NN3}}[\hat{P}_C(k), \hat{\dot{M}}_F(k), \hat{T}_F(k), \hat{T}_0(k), u_{\text{FPOV}}(k), L(k)]$	RMLP, 6–6–1
$\hat{T}_0(k+1) = f_{\text{NN4}}[\hat{P}_C(k), \hat{\dot{M}}_F(k), \hat{T}_0(k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), L(k)]$	RMLP, 6–8–1

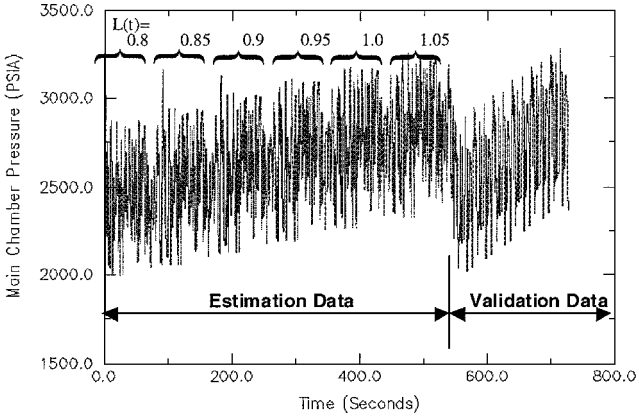

Fig. 2 Training data set for SSME main chamber pressure.

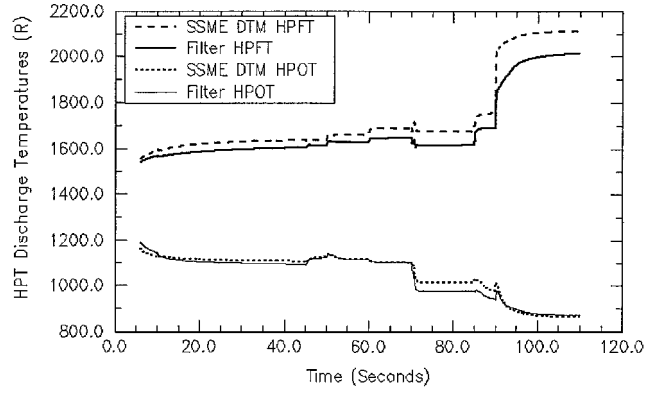

Fig. 3 Comparison of filtered estimates and SSME DTM responses for the HPT discharge temperatures during HPFTP degradation events.

during training was selected to influence the NNs to establish good interpolated accuracy for output and state estimates when $L(k)$ take values on the smaller, embedded range [1.0, 0.9], which provides suitable coverage for potential turbine failures.

To generate the training data set, the SSME DTM is initialized at the 105% power level setpoints for P_C and M_R, under the command of the SSME control system and with the parameter $L(k)$ set to 1.05. Next, the control system is disengaged from the FPOV and OPOV actuators, and for several seconds these actuators are commanded by a pseudorandom pattern of valve positions that correspond to demand settings in the range of 90–110% of the rated power. The behavior of the states of interest and the corresponding inputs are saved, and these steps are repeated for the next lower value of $L(k)$. The resulting data sets are concatenated to create the estimation data set that is used in predictor identification. Analogously, a validation data set is also created. The estimation and validation data sets are concatenated to create the training data set. Figure 2 shows the identification data set behavior for P_C. From inspection of Fig. 2, it is readily apparent that the magnitude of P_C varies monotonically with $L(k)$, as required by Eq. (2). A similar dependency is observed for the measured variable $\dot{M}_F$.

Training of the Filter NNs

In this case study, a state predictor is not explicitly used in the prediction step of the filtering algorithm. Rather, the state and output predictors that are given by Eq. (3) are combined, and a composite output predictor is obtained as follows:

$$\hat{\boldsymbol{y}}(k+1\,|\,k) = \boldsymbol{h}_{\text{NN}}[\boldsymbol{f}_{\text{NN}}(\hat{\boldsymbol{x}}(k\,|\,k), \boldsymbol{u}(k), \boldsymbol{y}(k), \hat{\boldsymbol{q}}(k\,|\,k)), \hat{\boldsymbol{q}}(k\,|\,k)]$$

$$\equiv \boldsymbol{g}_{\text{NN}}[\hat{\boldsymbol{x}}(k\,|\,k), \boldsymbol{u}(k), \boldsymbol{y}(k), \hat{\boldsymbol{q}}(k\,|\,k)] \tag{13}$$

In particular, two predictors are developed for the output vector $\hat{\boldsymbol{y}}(k+1\,|\,k) = [\hat{P}_C(k+1\,|\,k)\hat{\dot{M}}_F(k+1\,|\,k)]^T$. Also, because measured values for $P_C(k)$ and $\dot{M}_F(k)$ are available during the filter operation, the past measured values of these variables are used as inputs to the NN models. In the filter update step, the filter gradients and the adaptation gain are calculated by using Eqs. (5) and (6). The state updates for $\hat{T}_F(k+1\,|\,k+1)$ and $\hat{T}_0(k+1\,|\,k+1)$ are obtained by identifying explicit state predictors, as dictated by the second of Eqs. (4). Finally, the parameter $L(k)$ is updated by using a relation consistent with the last of Eqs. (4).

A summary of the SSME adaptive filter equations is as follows:

$$\hat{P}_C(k+1\,|\,k) = f_{\text{NN1}}[P_C(k), \dot{M}_F(k),$$

$$\hat{T}_F(k\,|\,k), \hat{T}_0(k\,|\,k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), \hat{L}(k\,|\,k)]$$

$$\hat{\dot{M}}_F(k+1\,|\,k) = f_{\text{NN2}}[P_C(k), \dot{M}_F(k),$$

$$\hat{T}_F(k\,|\,k), \hat{T}_0(k\,|\,k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), \hat{L}(k\,|\,k)]$$

$$\hat{L}(k\,|\,k+1) = \hat{L}(k\,|\,k) + R_L(k+1\,|\,k+1)J_L(k+1\,|\,k+1)$$

$$\hat{T}_F(k+1\,|\,k+1) = f_{\text{NN3}}[P_C(k), \dot{M}_F(k),$$

$$\hat{T}_F(k\,|\,k), \hat{T}_0(k\,|\,k), u_{\text{FPOV}}(k), \hat{L}(k\,|\,k+1)]$$

$$\hat{T}_0(k+1\,|\,k+1) = f_{\text{NN4}}[P_C(k), \dot{M}_F(k),$$

$$\hat{T}_0(k\,|\,k), u_{\text{FPOV}}(k), u_{\text{OPOV}}(k), \hat{L}(k\,|\,k+1)]$$

$$\hat{L}(k+1\,|\,k+1) = \hat{L}(k\,|\,k+1) \tag{14}$$

The NNs involved in these predictors are all RMLPs with GF. A summary of the resulting filter NN architectures is presented in Table 1. To obtain the results of Table 1, a number of network architectures were trained for each predictor. After selecting a promising architecture, further training is performed, and the validation data set error is used to stop training. Network training is completed after suitably small prediction errors are obtained on the validation data sets for each identified predictor. Following approximately 6000 iterations on the selected network architecture, the prediction errors on the validation data sets begin to increase and learning is terminated. The relative validation errors range from approximately 0.5 to 2.5%.

SSME Monitoring Filter Testing Results

Filter Testing Using Simulated Data

The SSME DTM is used to generate testing data sets. These data sets are power level transients with and without high-pressure turbopump failures. Both high-pressure oxidizer turbopump (HPOTP) and HPFTP failures are modeled using the SSME DTM. A few representative cases are described here. The presented errors for the output predictions and the filtered state estimates are in terms of the absolute value of the maximum relative error (MRE) defined as

$$\text{MRE} = \max_k \left| \frac{\text{actual}(k) - \text{estimate}(k)}{\text{actual}(k)} \right| \tag{15}$$

Figure 3 shows a comparison of $\hat{T}_F(k\,|\,k)$ and the corresponding SSME DTM responses during a transient with HPFTP degradation events. In this case study, the engine power level is varied by using

Table 2 Simulation testing, error results for up to 5% HPT failure events

	MRE, %			
Description	P_C	$\dot{M}_F$	T_0	T_F
Nominal	0.5	0.5	4.5	3.5
HPFT failure	0.5	2	4.5	3.5
HPOT failure	3.5	1.5	2	2
HPFT and HPOT failure	1.5	1	2	2

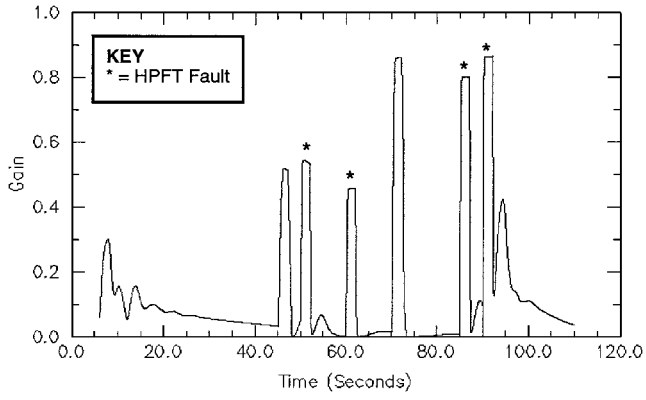

Fig. 4 Adaptive filter gain response to the simulated SSME DTM transient with HPFTP degradation events.

the P_C setpoint in steps, starting from the nominal 100% power level condition at 5 s, followed by a step up to 104% power level at 45 s, and a step down to 90% power at 70 s. HPFTP degradation events are modeled by setting the appropriate SSME DTM turbine coefficient to the values 1.0, 0.99, 0.98, 0.95, and 0.85 at the times 0.0, 50.0, 60.0, 85.0, and 90.0 s, respectively. Figure 3 demonstrates that for up to 5% drop in the HPFTP performance, occurring at 85 s, the HPT filtered states have less than 5% MRE with respect to the simulated testing data. However, for the 15% HPFTP performance degradation event, initiated at 90 s, the HPT filtered temperatures are as far as 150 R from the simulated values, resulting in approximately 8% MRE.

Table 2 presents a summary of the filter simulated testing results when HPT degradation does not exceed 5%. Table 2 demonstrates that the HPT filtered states are within 5% of the simulated test data for all cases, when the HPT performance degradation is no more that 5%. The prediction errors for P_C and $\dot{M}_F$ are consistently low and do not show an obvious relation to the state estimation errors of the HPT discharge temperatures. However, the prediction error for P_C increases to 1.5 and 3.5% (Table 2) when the test data include HPOTP degradation. This increase is an apparent indication that such a condition was not well represented in the training data set and potentially outside the filter scope, notwithstanding that the state-estimation errors may still be quite low. Such behavior, where the prediction error provides some indication of the applicability of the filter for the given system conditions is highly desirable.

Figure 4 shows the behavior of the adaptive filter gain for the same transient case illustrated in Fig. 3. The adaptive filter gain is observed to take large values at times corresponding to setpoint changes 45 and 70 s and also at the times corresponding to the modeled failure occurrences, marked by an asterisk in Fig. 4. At these times, the deterministic component of the gradient cost function, that is, the modeling error, becomes excited. While filter adaptation is rapid, the large gain values persist for the length of the filter window τ [see Eq. (6)], which for this case was set to 50 samples, or equivalently to 2 s. As a result, all of the failure events can be effectively detected by discriminating them from setpoint changes. Discrimination of simultaneous setpoint changes and failure events has not been explored.

For comparison, Figs. 5 and 6 shows the HPT discharge temperature estimates and the adaptive filter gain response, respectively, for an SSME DTM transient exactly analogous to that described earlier and represented by Figs. 3 and 4, but instead of HPFT failures,

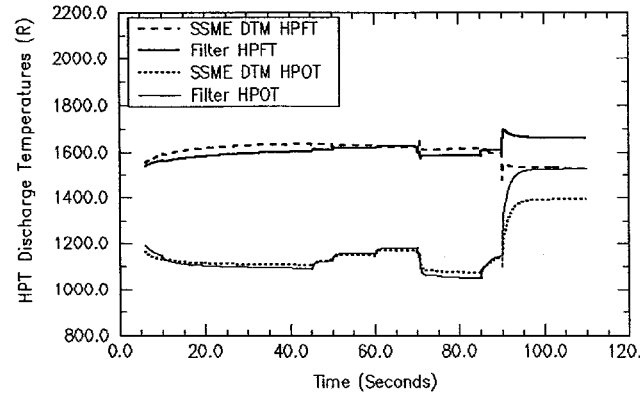

Fig. 5 Comparison of filtered estimates and SSME DTM responses for the HPT discharge temperatures during HPOTP degradation events.

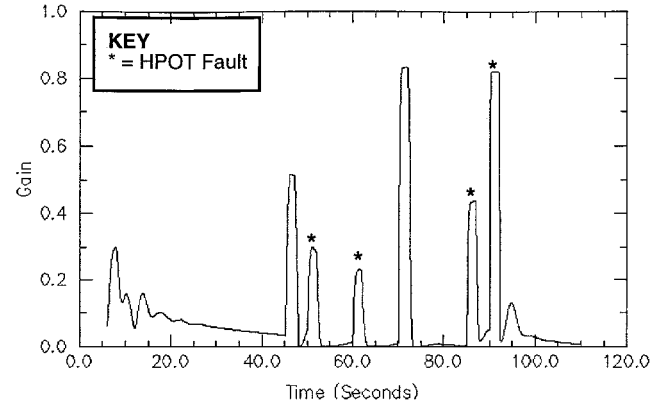

Fig. 6 Adaptive filter gain response to the simulated SSME DTM transient with HPOTP degradation events.

equivalent magnitude HPOT failures were modeled. The MRE for the state estimates, shown in Fig. 5, remains within 2% (Table 2), until the 85% HPOTP degradation event, at 90 s. Figure 6 shows that, except for the 85% turbine degraded performance event, the magnitudes of adaptive gain response to HPOTP failure events (marked by an asterisk in Fig. 6) are significantly lower than that for the analogous HPFTP failure events (Fig. 4). Such preferential sensitivity of the adaptive gain response to HPFTP failure was observed to be consistent in this study, but was not investigated in depth.[23] However, observe that, given such relative sensitivity of the filter gain, its response combined with the filtered state estimates is potentially informative for fault isolation methods.

Filter Testing Using Experimental Data

The adaptive filter developed herein was applied to data from an SSC test firing of an SSME, as already described. Figures 7 and 8 compare the filtered HPT discharge temperatures with the measured signals for T_F and T_0, from the initiation of the mainstage operations to engine shutdown. In Fig. 7, the HPFT discharge temperature estimates track the general behavior of the actual HPFT discharge temperature response with degradation events in the HPFTP at 130, 276, and 404 s. The estimation error grows larger after the engine power level is set to 104% power level at 140 s. Throughout the transient, the MRE for the HPFT discharge temperature estimates is less than 5%. In Fig. 8, the filter estimates for HPOT discharge temperature appear to be largely bounded by the target system response. In some instances, the filter estimates substantially undershoot the actual HPOT discharge temperature behavior. During a 20-s period while the SSME is at 90% power, the MRE for the HPOT discharge temperature estimates reaches 10%. Also, for 25 s after the engine is set to the 104% power level, the MRE reaches 6%, whereas during the remainder of the transient it is well below 5%.

The MRE values comparing the filtered estimates with the SSC test data are summarized in Table 3. As expected, the maximum filter prediction and state-estimation errors are larger than the equivalent

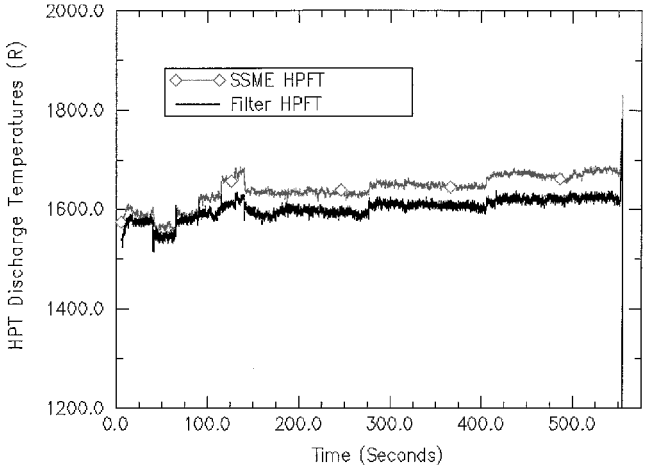

Fig. 7 Comparison of the filtered estimates to measured SSME HPFT discharge temperature for the SSC test data.

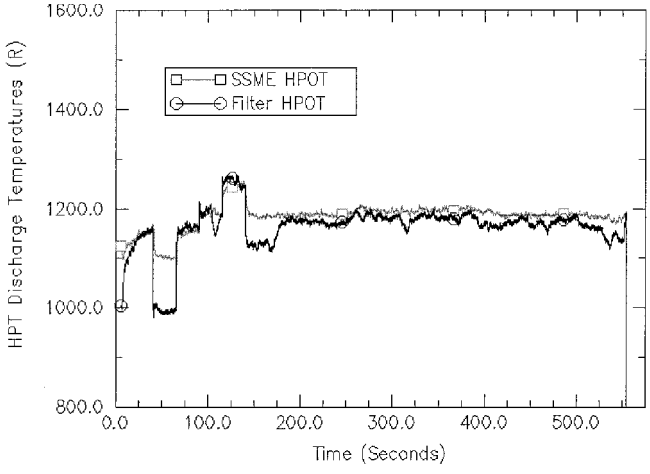

Fig. 8 Comparison of the filtered estimate to measured SSME HPOT discharge temperature for the SSC test data.

errors in the simulated testing results. Based on the simulated testing results, larger prediction errors for P_C in the SSC test data case may be an indication that the NN models do not represent the target system well. As a benchmark, the simulated SSME DTM responses were also directly compared to the SSC data. To enable this comparison, the SSME control schedule was modeled to match that of the SSC data. The MRE values comparing the simulated SSME DTM and SSC responses, before HPFTP failure at 130 s, are also given in Table 3. Table 3 shows that the errors for the SSME DTM responses compared directly to the SSC test data are similarly large as the filter estimation errors for the SSC data, suggesting that the SSME DTM did not accurately represent the actual engine considered. Although the SSME DTM is capable of high accuracy modeling of the actual SSME behavior, the inaccuracies are to be expected because in this case study the SSME DTM coefficients were not tuned using the actual SSC test data as a reference. Such tuning is typically required to match the DTM to the responses of the engine configuration considered. That the estimation errors for both the filter and the SSME DTM compared to the SSC test data are of similar magnitude further suggests that the filter accuracy is limited by the modeling error present in the training data set.

Figure 9 demonstrates that the adaptive filter gain becomes significantly large at times corresponding to pressure setpoint changes and the HPFTP degradation events described earlier. As with the simulated test case described earlier, the adaptive filter gain effectively detects each failure event, which are marked by an asterisk in Fig. 9. Such behavior suggests some potential benefit of the proposed adaptive filter to fault detection methods, where some type of discriminate analysis can be utilized to exclude setpoint changes and

Table 3 Estimation errors for SSC test data comparisons

	MRE, %			
Comparison	P_C	$\dot{M}_F$	T_F	T_0
Filter estimates and SSC	2.5	2	5	11
SSME DTM estimates and SSC	3	2	6	10

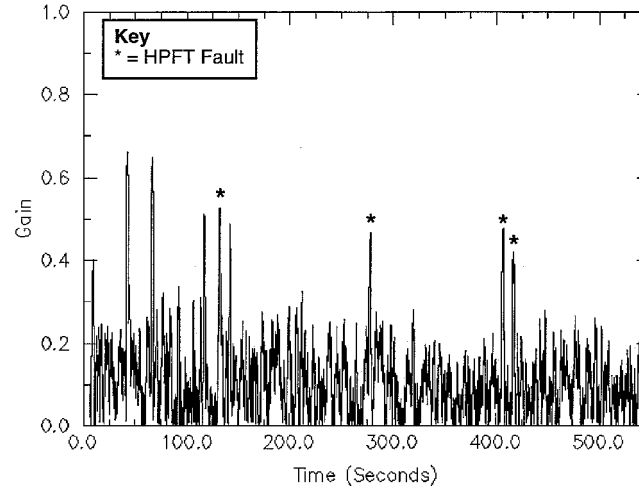

Fig. 9 Adaptive filter gain response for the measured SSME behavior during the SSC test data.

other known disturbance events. Such an approach will permit the real-time detection of certain component degradations at similarly incipient stages as demonstrated herein concerning SSME turbine failure.

Conclusions

Adaptive state filtering solutions, suitable for practical complex dynamic systems with severe failure modes and unknown system state equations are scarce. However, fault-tolerant control and monitoring problems can require highly selective and critical information that is impractical to measure. A novel adaptive state filtering algorithm has been presented for a limited class of such problems. The proposed filter utilizes a predictor-update approach, combining recurrent NN predictors, describing the state-space behavior of a given system, with a novel recursive LMS state-update algorithm. The filter is proposed for application to highly nonlinear systems, where a certain monotonic relationship between time-varying parameters and measured system outputs can be assumed.

The filter is applied for estimation of the turbine discharge temperatures from actual SSME signals, while the engine is also experiencing HPT failures. The adaptive filter demonstrates acceptably good accuracy, limited only by the accuracy of the NN predictors and the associated training set used to develop them. The filtering results demonstrate that NN predictors can be utilized in closed-loop adaptive filtering algorithms, along the lines of conventional restrictive complexity adaptive filters. The ability of the adaptive filter gain to enable detection of faults is also demonstrated. A mechanism for discrimination of such faults from setpoint changes and deterministic disturbances must be established. Substantial subjective responsibility remains with the filter designer because of the trial and error and extensive experience required for ascertaining the suitability of the proposed adaptive filtering method to specific problems and because of the complexities involved in identifying accurate NN predictors.

Acknowledgments

This research has received financial support by NASA Grant NTG-70248 administered through NASA Johnson Space Center. The authors thank David Seymour and Louis Trevino of NASA Marshall Space Flight Center for their valued assistance and for providing the Stennis Space Center test data.

References

[1]Xiaowen Dai, R. A., "Damage-Mitigating Control of a Reusable Rocket Engine: Part I—Life Prediction of the Main Thrust Chamber Wall," *Journal of Dynamic Systems, Measurement and Control*, Vol. 118, No. 3, 1996, pp. 401–408.

[2]Benzing, D. A., and Whitaker, K. W., "Approach to Space Shuttle Main Engine Health Monitoring Using Plume Spectra," *Journal of Spacecraft and Rockets*, Vol. 35, No. 6, 1998, pp. 830–836.

[3]Boskovic, J. D., and Mehra, R. K., "Intelligent Adaptive Control of a Tailless Advanced Fighter Aircraft Under Wing Damage," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, 2000, pp. 876–884.

[4]Musgrave, J. L., Guo, T.-H., Wong, E., and Duyar, A., "Real-Time Accommodation of Actuator Faults on a Reusable Rocket Engine," *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 1, 1997, pp. 100–109.

[5]Neidhoefer, J. C., and Krishnakumar, K., "Intelligent Control for Near-Autonomous Aircraft Missions," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, Vol. 31, No. 1, 2001, pp. 14–28.

[6]Goodwin, G. C., and Sin, K. S., *Adaptive Filtering, Prediction, and Control*, 1st ed., Prentice–Hall, Upper Saddle River, NJ, 1984, pp. 150–350.

[7]Haykin, S., *Adaptive Filter Theory*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1991, pp. 299–358, 477–504.

[8]Garulli, A., Kacewicz, B., Vicino, A., and Zappa, G., "Error Bounds for Conditional Algorithms in Restricted Complexity Set Membership Identification," *IEEE Transactions on Automatic Control*, Vol. 45, No. 1, 1997, pp. 160–164.

[9]Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.

[10]Ljung, L., "Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems," *IEEE Transactions on Automatic Control*, Vol. 24, No.1, 1979, pp. 36–50.

[11]Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27.

[12]Narendra, K. S., and Parthasarathy, K., "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, 1991, pp. 252–262.

[13]Parlos, A., Chong, K. T., and Atiya, A., "Application of the Recurrent Multilayer Perceptron in Modeling Complex Process Dynamics," *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, pp. 255–266.

[14]Narendra, K. S., and Mukhopadhyay, S., "Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks," *Neural Networks*, Vol. 3, No. 2, 1994, pp. 737–752.

[15]Parlos, A. G., Rais, O., and Atiya, A. F., "Multi-Step-Ahead Prediction Using Dynamic Recurrent Neural Networks," *Neural Networks*, Vol. 13, No. 4, 2000, pp. 765–786.

[16]Saravanan, N., Duyar, A., Guo, T.-H., Merrill, W. C., "Modeling Space Shuttle Main Engine Using Feed-Forward Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 641–648.

[17]Lo, J. T.-H., "Synthetic Approach to Optimal Filtering," *IEEE Transactions on Neural Networks*, Vol. 5, No. 5, 1994, pp. 803–811.

[18]Alessandri, A., Baglietto, M., Parisini, T., and Zoppoli, R., "A Neural State Estimator with Bounded Errors for Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 44, No. 11, 1998, pp. 2028–2042.

[19]Haykin, S., Yee, P., and Derbez, E., "Optimal Nonlinear Filtering," *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, 1997, pp. 2774–2786.

[20]Feldkamp, L. A., and Puskorius, G. V., "A Signal Processing Framework Based on Dynamic Neural Networks with Applications to Problems in Adaptation, Filtering, and Classification, *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2259–2277.

[21]Parlos, A. G., Menon, S. K., and Atiya, A. F., "An Algorithmic Approach to Adaptive State Filtering Using Recurrent Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 12, No. 6, 2001, pp. 1411–1432.

[22]Parlos, A. G., Menon, S. K., and Atiya, A. F., "An Adaptive State Filtering Algorithm for Systems with Partially Known Dynamics," *Journal of Dynamic Systems, Measurement and Control*, Vol. 124, No. 3, 2002, pp. 364–374.

[23]Williams, R. B., "Adaptive State Filtering with Application to Reuseable Rocket Engines," Ph.D. Dissertation, Nuclear Engineering Dept., Texas A&M Univ., College Station, TX, May 1997.

[24]Duyar, A., Guo, T.-H., and Merrill, W. C., "Space Shuttle Main Engine Model Identification," *IEEE Control Systems Magazine*, Vol. 10, No. 4, 1990, pp. 59–65.

[25]Guo, T.-H., and Musgrave, J., "Neural Network Based Sensor Validation for Reusable Rocket Engines," *1995 American Control Conference—ACC'95*, Vol. 2, American Automatic Control Council, Evanston, IL, 1995, pp. 1367–1372.

[26]Lorenzo, C. F., Holmes, M. S., and Ray, A., "Nonlinear Life-Extending Control of a Rocket Engine," *Journal of Guidance, Control, and Dynamics,* Vol. 23, No. 4, 2000, pp. 759–762.

[27]Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974, pp. 120–126.

[28]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.

[29]Barron, A. R., "Approximation and Estimation Bounds for Artificial Neural Networks," *Journal of Machine Learning*, Vol. 14, No. 1, 1994, pp. 115–133.

[30]Cybenko, G., "Approximation by Superposition of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, 1989, pp. 133–141.

[31]Hornik, K., Stinchcombe, M., and White, H., "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, Vol. 3, No. 5, 1990, pp. 551–560.

[32]Nelson, R., Taniguchi, M., Von Bieren, L., and Landis, C., "Engine Balance and Dynamic Model," Rocketdyne Div., Rept. RL00001, rev. H, Rockwell International, Canoga Park, CA, Jan. 1992.

M. P. Nemeth
Associate Editor