JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 4, July–August 1994

Modeling Space Shuttle Main Engine Using Feed-Forward Neural Networks

N. Saravanan* and A. Duyar†
Florida Atlantic University, Boca Raton, Florida 33431
and
T. -H. Guo‡ and W. C. Merrill§
NASA Lewis Research Center, Cleveland, Ohio 44135

We present the modeling of the Space Shuttle main engine (SSME) using a feed-forward neural network. The input and output data for modeling are obtained from a nonlinear performance simulation. The SSME is modeled as a system with two inputs and four outputs. The backpropagation algorithm is used to train the neural network by minimizing the squares of the residuals. The inputs to the network are the delayed values of the selected inputs and outputs of the nonlinear simulation. The results obtained from the neural network model are compared with the results obtained from the nonlinear simulation, under two different experimental conditions. The results show that a single neural network can be used to model the dynamics of the SSME. This neural network model can be used for control system design and development, as well as for model-based fault detection studies.

Nomenclature

N = number of samples
P = pressure, psi
S = speed, rpm
t = discrete time integer
β = valve actuator output rotary motion, dimensionless
δu = deviation of the input from nominal
δy = deviation of the output from nominal

I. Introduction

IN recent years, there has been considerable interest in the modeling of the Space shuttle main engine (SSME).[1–3] It is becoming increasingly important to have an accurate model of the SSME that can generate data in real time, for control system design and fault detection purposes.[4–6] Currently, a complete nonlinear dynamic simulation,[7] also known as the digital transient model (DTM), is available for the SSME. This nonlinear simulation is composed of three major subprograms, each of which describes a primary fluid subsystem of the engine and its related processes. These major subprograms are in turn built up from the component process elements. The process elements are used to describe the physical processes that occur in each of the engine subsystems. Because of its size and complexity (20 min of CPU time for 20 s of real-time operation with a SPARCstation 330), this nonlinear simulation (DTM) cannot be used to generate data for real-time applications. Therefore, it is useful to obtain a simple model for the SSME; one that possibly can operate in real time. Recent efforts in the area of modeling of the SSME include the use of single-input single-output (SISO) identification techniques[1] and multi-input multi-output (MIMO) identification techniques[2] (two inputs and two outputs) to obtain linear or piecewise linear[2] point models at different operating points of the SSME. The point models are valid in a limited response region about the operating point corresponding to the power levels. These point models are linked by regressing the parameters of the point models to obtain an overall quasilinear model of the SSME.[3] In contrast to the linked quasilinear model, which requires the identification of point models at several operating points using linear identification techniques, and the nonlinear simulation, which is large and complex, this study demonstrates that a single neural network model can be used to capture the nonlinear dynamics (with sufficient accuracy) of the SSME in the entire 80–100% power level. Furthermore, the identified neural network model can be used to generate data in real time; both for model-based fault detection and diagnosis studies and control system design and development.

The objective of the work is, therefore, to develop a nonlinear model of the SSME using a feed-forward neural network with input–output data obtained from the nonlinear simulation. The feed-forward neural network is employed as a Nonlinear Auto-Regressive with eXogenous inputs (NARX) model [see Eq. (1)] in which the network is fed with past system inputs and outputs to predict the future system output. The neural network model consists of a single hidden layer of sigmoidal units, and it is trained using the backpropagation algorithm.

This paper is organized into seven sections. Artificial neural networks are briefly reviewed in Sec. II. The motivation for using neural networks for modeling along with the different neural network architectures commonly used for modeling are discussed in Sec. III. Section IV gives a short description of the operation of the SSME. A systematic approach for modeling of the SSME using feed-forward neural networks is described in Sec. V. In Sec. VI, the results of the modeling effort are presented. This includes a comparison of the responses of the identified model of the SSME with the nonlinear simulation. Concluding remarks are given in Sec. VII.

II. Background: Artificial Neural Networks

Artificial neural networks (ANN), also called connectionist models, parallel distributed models, or simply neural nets consist of many simple computational elements called nodes, or neurons, each of which collects the signals from other nodes connected to it directionally. The architecture of these models is specified by the node characteristics, network topology, and learning algorithm. Nodes in artificial neural networks are very

Received Nov. 9, 1992; revision received Aug. 2, 1993; accepted for publication Oct. 15, 1993.

*Graduate Student, Mechanical Engineering Department. Student Member AIAA.

†Professor, Mechanical Engineering Department.

‡Research Engineer, Mail Stop 77-1, 21000 Brookpark Road.

§Branch Chief, Advanced Controls Technology Branch, Mail Stop 77-1, 21000 Brookpark Road. Senior Member AIAA.

simple processors inspired by their biological counterparts. The network is inherently parallel so that many nodes can carry out their computations simultaneously. Each node has a rule for combining the inputs from other nodes to provide a new state of activation. This activity is simply the weighted sum of signals that the node receives from other nodes connected to it directionally. This sum, the net input to the node, is processed by a function (usually a sigmoid) resulting in the output or the activation of that node.

ANN can be placed into one of two classes based on their network topology: recurrent (global feedback connections, e.g., Hopfield neural networks,[8,9] and local feedback connections, e.g., cellular neural networks[10]) and nonrecurrent (no feedback connections, e.g., perceptron[11]). A special type of nonrecurrent ANN is the feed-forward neural network. Because feed-forward neural networks are used in this study, a brief description of their architecture follows.

In multilayer feed-forward networks, there are a few layers (the input layer, the output layer, and possibly some hidden layers) across which all of the nodes of each layer are connected to all of the nodes in the layer above it, but there are no connections within the layer. A constant term, a bias, is usually added to each of the nodes. The feed-forward neural network has an input layer that simply transmits the input variables without any processing to the next layer. The bias is a weighted unit input to each node, thus adding a constant term to the net input of the node. Each node in the upper layers (hidden or output layers) receives weighted inputs from each of the nodes in the layer below it. These weighted inputs are summed up to get the net input of the node. The output (net activation) of the node is calculated by an activation function of the net input.

Once the type of neural network has been chosen, we must "teach" the neural network to perform the desired task by iterative adjustment of the weights. This may be done in the following two ways[12]: supervised and/or unsupervised. In supervised learning, the learning is done on the basis of direct comparison of the output of the network with known correct answers. In unsupervised learning, the learning goal is not defined at all in terms of specific correct examples. The only available information is in the correlations of the input data or signals. The neural network is expected to create categories from these correlations, and to produce output signals corresponding to the input category.

A robust learning algorithm called the generalized delta rule or backpropagation is the most commonly used algorithm to train feed-forward neural networks. Outwardly, the backpropagation seems to be a straightforward, gradient-descent technique to minimize the objective function that is differentiable with respect to the weights of the network. However, in the backpropagation technique, the gradient descent expressions appear as discrete, weight-update rules that use only information available locally to each node and a quantity that is back-propagated from nodes in the penultimate layer. Learning via backpropagation involves two phases. In the first phase, the inputs are propagated in a feed-forward manner through the network to produce output values that are compared to the target values, resulting in the error signal for each of the output nodes. In the second phase, the errors are propagated backward through the network and used to adjust the weights.

An important feature of neural networks is their ability to generalize to new situations. After being trained on a number of examples of a relationship, they can often induce a complete relationship that interpolates and extrapolates from the examples in a sensible manner. Furthermore, neural networks with feed-forward architecture can approximate, with arbitrary precision, static nonlinear functions.[13,14] Because of their approximation capabilities as well as their inherent adaptivity features, artificial neural networks have been used in a variety of applications, including sonar target recognition, image compression, navigation, identification, and intelligent control. From a practical perspective, the massive parallelism and fast adaptability of neural network implementations provide further incentives for investigating the neural network approach for modeling of the SSME.

III. Neural Networks in Modeling

The area of system identification has matured greatly over the last two decades.[15,16] However, most of the effort has been directed toward developing techniques for linear systems. Whereas linear time-invariant systems can be characterized by their impulse responses, frequency responses, transfer functions, or state equations, the characterization of nonlinear dynamic systems is considerably more complex because all systems that are not linear are contained in this class. Also, many well-known methods exist for the identification of linear time-invariant systems characterized in any of the above forms.[16] In the case of nonlinear system identification, the problem is in finding an efficient set of nonlinear functions to approximate nonlinear systems for which the system structure is unknown. Even major contributions to the characterization of nonlinear systems, such as Volterra series, the Wiener series, and the Uryson operator, have not yielded generic and efficient methods for identification of nonlinear systems. For example, the use of Volterra series,[17] may lead to the estimation of a large number of parameters, especially for highly nonlinear systems. Previous studies in the area of nonlinear system identification include use of the Hammerstein model,[18] which basically is a dynamic linear model in series with a static nonlinear model, and the polynomial NARMAX (Nonlinear Auto-Regressive Moving Average with eXogenous inputs) model,[19] which gives a description of the system in terms of a nonlinear functional expansion, such as the polynomial expansion, of delayed inputs, outputs, and predicted errors. Billings et al. and other researchers[20,21] have developed several identification procedures based upon the NARMAX model for known model structures. In general, there are various identification algorithms for linear systems[2,15,16] and for nonlinear systems with known structure,[20,21] but a lack of general methodology for nonlinear system identification with unknown structure.

Because of these reasons, neural networks as universal approximators have a wide range of applications in nonlinear dynamic system identification. As mentioned previously, neural networks can be considered to be versatile mappings whose response to a specific input is determined by the values of the adjustable weights. Recent theoretical works[13,14] have proven that, even with only one hidden layer, feed-forward neural networks can uniformly approximate any continuous function. Furthermore, it has been shown[22] that a network with two hidden layers is capable of approximating discontinuous functions. Therefore, increasing the number of layers results in a natural enlargement of the class of functions that can be approximated. Thus, the theoretical basis for modeling nonlinear systems by neural networks is sound.

For identification of nonlinear dynamic systems, when only the input–output data are available, the two transfer function models that are commonly used are the series-parallel identification model[23]

$$\hat{y}(t) = F_f(y(t-1), \ldots, y(t-k_y), u(t-1), \ldots, u(t-k_u)) \tag{1}$$

and the parallel identification model

$$\hat{y}(t) = F_r(\hat{y}(t-1), \ldots, \hat{y}(t-k_y), u(k-1), \ldots, u(t-k_u)) \tag{2}$$

where

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix}, \qquad \boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix}$$

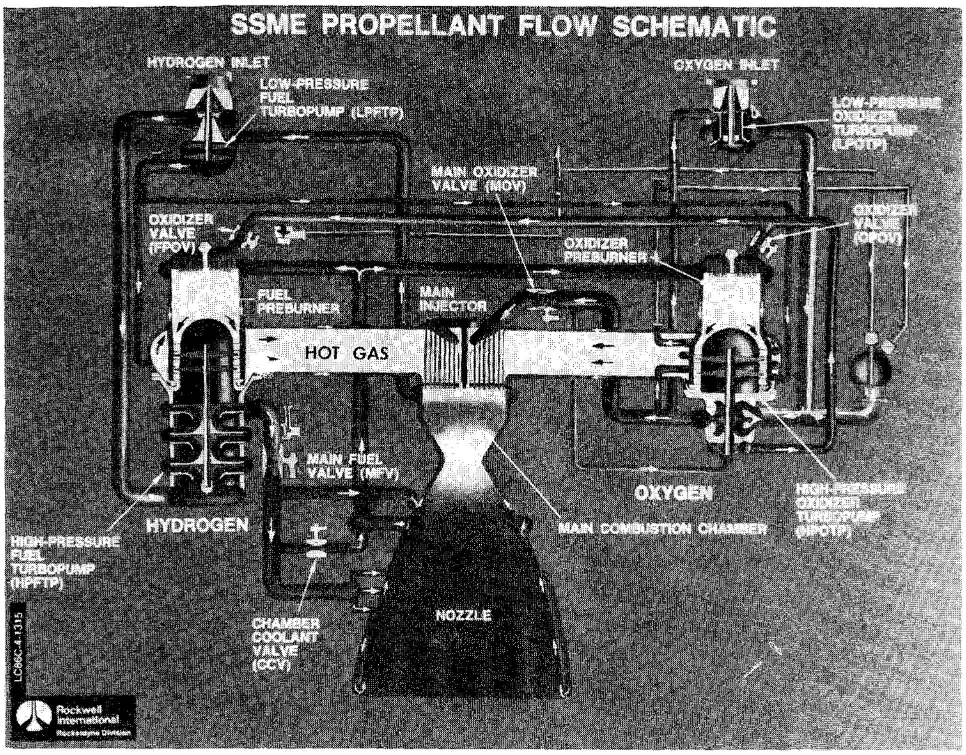

Fig. 1 SSME Propellant flow schematic.

are the system output and input vectors, respectively; k_y and k_u are the maximum lags in the output and input, respectively; r and m are the number of inputs and outputs, respectively; and F_f and F_r some vector-valued nonlinear functions.

When using neural networks for modeling dynamic systems we must train the neural network not only with the current data, but also with the past data. This can result in two different schemes to establish the neural network architecture. A feed-forward neural network can be employed as a series-parallel model in which the neural network is fed with the past inputs and outputs to predict the future system output; i.e., past measurement outputs of the system are used as current inputs to the network. In the parallel model the past system outputs are replaced by the outputs from the network; i.e., estimated outputs from the network are used as current inputs to the network. Whereas the series-parallel method requires only a feed-forward network that can be trained by the generalized delta rule,[24] topologically, the parallel identification method results in a neural network with delayed recurrent connections from its output nodes back to its own input nodes and has to be trained by recurrent learning rules.[25,26] It is desirable to have a parallel model because it depends only on the input data to predict the output. However, it is well known[23] that if a parallel model is used, there is no guarantee that the parameters (network weights) will converge or that the output error will become very small. Even in a linear case, the conditions under which parallel model parameters converge are at present unknown. Also, once a feed-forward network is trained as a series-parallel model it can be used as a parallel model as long as the output error tends to go to a small value asymptotically so that $y(t) \approx \hat{y}(t)$.

The aim of this study is to use a feed-forward neural network with one hidden layer to model the nonlinear dynamics of the SSME. Therefore, it is necessary to decide on the inputs and outputs to the neural network, the architecture of the feed-forward neural network, and the algorithm to determine the weights of the neural network. The input layer to the neural network consists of $n = rk_u + mk_y$ neurons, and receives its input as

$$[u^T(t-1) \cdots u^T(t-k_u)\, y^T(t-1) \cdots y^T(t-k_y)]$$

and the output layer receives desired output vector $y^T(t)$. Here, r is the number of inputs and m is the number of outputs considered in the identification. Another important step in modeling using neural networks involves the use of an algorithm or rule (adaptive or batch), which adjusts the parameters (weights) of the network based on a given set of input-output pairs (given above). If the network's weights are considered as the elements of parameter vector $\boldsymbol{\Theta}$ defined as $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_{k_\theta}]$ (weights and biases of the network are ordered in a chosen way), the learning process involves the determination of the vector $\boldsymbol{\Theta}^*$, which optimizes a performance criterion (or function) J based on the output error. For example, if the backpropagation learning rule is used, the gradient of the performance function with respect to $\boldsymbol{\Theta}$ is computed as $\nabla_{\Theta} J$, and $\boldsymbol{\Theta}$ is adjusted along the negative gradient as

$$\boldsymbol{\Theta} = \boldsymbol{\Theta}_{\text{nom}} - \eta \nabla_{\Theta} J|_{\Theta=\Theta_{\text{nom}}} \tag{3}$$

where η is the step size (learning rate), a suitably chosen constant, and $\boldsymbol{\Theta}_{\text{nom}}$ denotes the previous value of $\boldsymbol{\Theta}$ at which the gradient is computed.

The steps involved in the use of neural networks for nonlinear modeling of the SSME are described in detail in Sec. V.

IV. Description of the SSME

A brief description of the SSME is given here. An extended treatment of the principles of operation and other aspects can be found elsewhere in literature.[27]

The Space Shuttle orbiter main propulsion system is composed of three main engines. The engines use liquid oxygen and liquid hydrogen propellants carried in an external tank attached to the orbiter. To understand the overall flow of fuel and oxidizer to produce the thrust, a schematic diagram of the propellant flows and the control valves is shown in Fig. 1. Pressurized fuel, provided by fuel tank flows through the low-pressure fuel turbopump and the high-pressure fuel turbopump, is fed to the regenerative cooling and the preburners. A pressurized oxidizer tank provides the oxidizer. It flows through the low-pressure oxidizer turbopump, then, the high-pressure oxidizer turbopump where the output flow splits, as shown in Fig. 1.

The two high-pressure turbines are driven by a fuel turbine preburner and an oxidizer turbine preburner, each of which produces hot gas. The low-pressure turbines are driven by corresponding high-pressure pump flows. The fuel from the high-pressure fuel turbopump (HPFTP) goes through the main fuel valve (MFV). After the MFV, the flow divides into fixed nozzle cooling flow, main chamber cooling flow, and the chamber coolant valve (CCV) flow. Heat is absorbed from the combustion chamber nozzle and the fuel flows to the preburners where the combustion and pressure are controlled by the fuel preburner oxidizer valve (FPOV) and the oxidizer preburner oxidizer valve (OPOV). The high-pressure fuel rich hot gas from the preburners goes through the two high-pressure turbines and returns through the hot gas manifolds to the fuel injector (FI) of the combustion chamber. Main chamber cooling fuel flow powers the low-pressure fuel turbopump (LPFTP) and returns directly to the fuel injector of the main combustion chamber.

Oxidizer from the high-pressure oxidizer turbopump (HPOTP) splits into three paths. Flow from the first path drives the low-pressure oxidizer turbine (LPOT). The output flow from the LPOT merges with the output of the low-pressure oxidizer pump (LPOP). The second flow goes through the main oxidizer valve (MOV) to the oxidizer injector (OI) in the combustion chamber. The third flow enters the preburner oxidizer pump and is divided between the preburners as determined by the FPOV and the OPOV. This completes the flow paths for turbine drive and control.

V. Modeling of the SSME

A. Selection of Inputs and Outputs

The existing control system of the SSME uses five valves: FPOV, OPOV, MFV, MOV, and CCV. They control the mixture ratio and the main chamber pressure (which is correlated with the thrust). Open- and closed-loop control of these valves is used to accomplish the SSME mission. For the purpose of modeling the open-loop system, the rotary motion of the five valves (β_{OPOV}, β_{FPOV}, β_{CCV}, β_{MOV}, β_{MFV}) are selected as inputs. The outputs are the chamber pressure P_C, mixture ratio MR, and the speeds of the high-pressure fuel and oxidizer turbines, S_{HPFT} and S_{HPOT}, respectively. From a previous study[1], it is known that β_{CCV}, β_{MOV}, and β_{MFV} are essentially decoupled from the outputs in the main stage of operation. Therefore, in this work, only the valve rotary motions, β_{FPOV} and β_{OPOV}, are used as inputs.

B. Input–Output Data Acquisition

To obtain the input–output data of the SSME, the dynamic nonlinear simulation (DTM)[7] is used. The combustion chamber pressure P_C is the variable that defines a nominal operating condition of the SSME.

To obtain the data for modeling, the nonlinear simulation with the closed-loop control is allowed to converge to the operating condition corresponding to the 90% power level. Then, the control is removed and the actuator outputs, β_{OPOV} and β_{FPOV}, are perturbed simultaneously using two uncorrelated five-level pseudorandom sequences.[28] Five-level sequences are used to satisfy the condition of "persistency of excitation"[16] and to have amplitude levels that would cover the whole domain of interest when training the neural network. The maximum magnitude of each sequence is 10% of the steady-state value (at 90% power level) of the input. The clock-time interval chosen for each sequence is 1.0 s, and the sampling time is 0.04 s. The length of the five-level sequence used is 24 intervals for each input.

C. Architecture Determination

This task in system modeling corresponds to the determination of the architecture of the feed-forward network, that is, the number of lags k_u and k_y (which gives the number of nodes needed in the input layer), the number of hidden layers, and the number of nodes in each of the hidden layers.

As a first step, it is necessary to know the number of lags k_u and k_y in Eq. (1) to model the process. In this work, this is determined by using a structure estimation algorithm,[2] which utilizes a row-by-row rank search of the observability matrix.

Table 1 Standard error of estimates for the test signals (HFTS and LFTS)

Model	Signal	P_C	MR	S_{HPFT}	S_{HPOT}
Model	HFTS	0.0470	0.0742	0.0910	0.0682
	LFTS	0.0912	0.0906	0.0690	0.0868

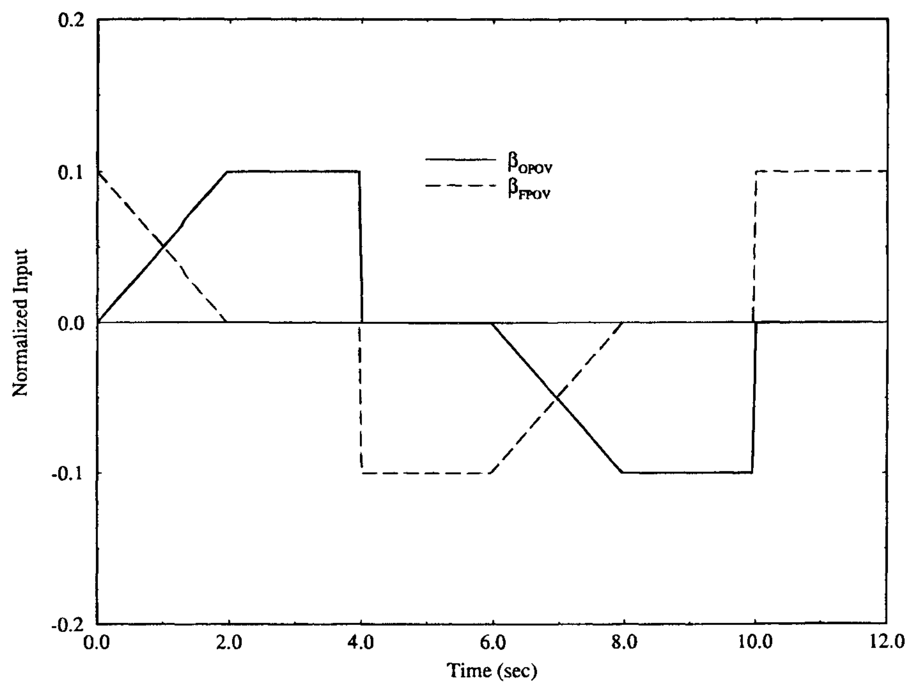

Fig. 2 The low frequency test signal (LFTS).

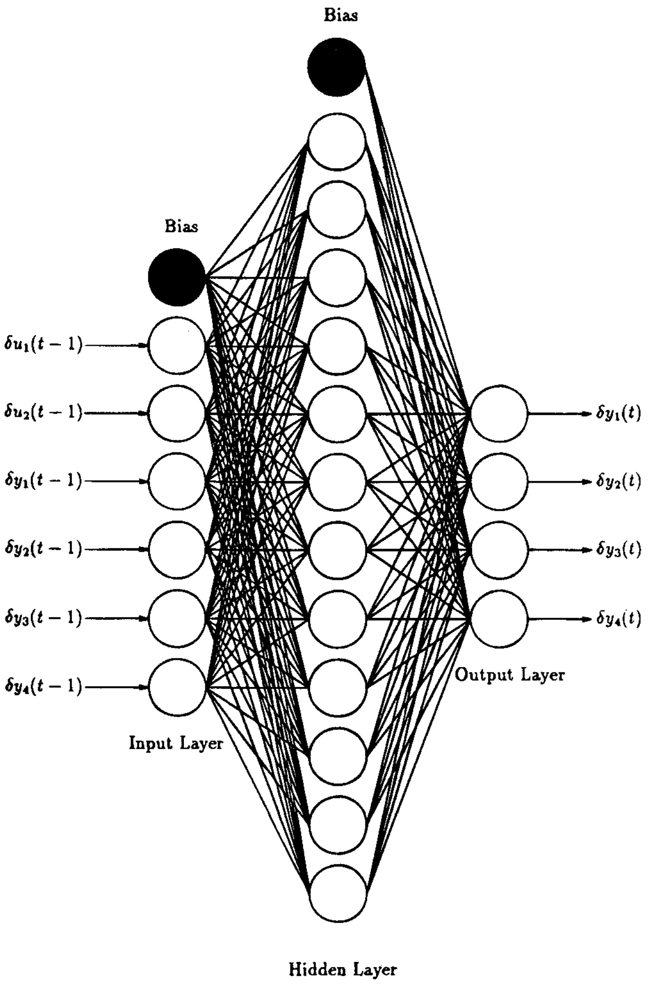

Fig. 3 Structure of the feed-forward neural network used for modeling.

The next step is the determination of the number of hidden layers in the feed-forward neural network. It has been shown that one hidden layer is sufficient to approximate any continuous function uniformly.[13,14] Therefore, a single hidden layer of sigmoidal units is utilized in this study.

The final step in structure determination involves determination of the number of nodes needed in the hidden layer. There is no definitive method for deciding a priori the number of units in the hidden layer. The number of nodes is related to the complexity of the nonlinear function that can be generated by the network. If too few nodes are provided, accuracy might be low because the representational capacity of the network is limited. However, if there are too many hidden layer nodes, the network will be prone to "overfitting"; that is, learning the stochastic variations in the data rather than the underlying functions. One approach to selecting the number of hidden layer nodes involves randomly splitting the data into two subsets, using one of the sets for training and the other for testing the generality of the result. If the average error on the training subset is significantly less than the testing subset (say, by a factor of two or more), overfitting is indicated and the number of nodes should be reduced. In this work, the number of nodes in the single hidden layer is determined by starting with a small number of nodes and increasing the number of nodes until there is no significant increase in modeling accuracy gained by increasing the number of nodes. We also check whether the network performs satisfactorily with the testing subset in order to avoid "overfitting."

The weights of the neural network are determined by using an iterative algorithm. In this study, the neural network weights and biases are determined by using backpropagation algorithm (the generalized delta rule), which is a kind of gradient descent method.[24] (Interested readers may refer to Rumelhart et al.[24] for further details about the generalized delta rule.) The weight changes in the backpropagation algorithm are governed by the learning rate η. One way to increase the learning rate η without leading to oscillation is to modify the generalized delta rule to include a momentum term α. (see page 330 in Ref. 24 for details). In this work, the learning rate and the momentum are set to 0.2 and 0.5, respectively.

D. Model Validation

Model validation forms the final stage of any identification procedure.[16] If the system under test is linear, a number of well-established tests are available for validating the estimated model. Well-known covariance tests, which consist of computing the autocorrelation function of the residuals and the cross-correlation between the residuals and the input developed for linear systems, provide incorrect information whenever nonlinear effects are present in the data. These covariance tests have been extended for use with nonlinear systems[29] by using higher-order correlation functions. These extended tests can be applied in a straightforward manner to validate a neural network model. Because in the present case the data for modeling are obtained from a noise-free environment (the nonlinear simulation), a much more direct approach to model validation would be to compare the responses of the neural network model with the responses obtained from the nonlinear simulation under the following experimental conditions.

1. High-Frequency Test Signal

The high-frequency test signal (HFTS) consists of two uncorrelated full-length pseudorandom binary perturbation sequences. The magnitude of each sequence is 10% of the steady-state value of the input at the 90% power level. The length of the data is 127 with a clock interval of 0.04 s.

2. Low-Frequency Test Signal

The low-frequency test signal (LFTS) consists of two signals made up of steps and ramps (see Fig. 2). The maximum magnitude of each sequence is 10% on top of the steady-state value of the input at the 90% power level. The length of the data is 300 with a sampling time of 0.04 s.

Together, the HFTS and the LFTS can be used to test the validity of the identified model in both the high- and low-frequency ranges, respectively.

A standard error of estimates (SEE) is defined as

$$\mathrm{SEE} = \left(\frac{\sum_{k=1}^{N} [y_m(k) - y_s(k)]^2}{\sum_{k=1}^{N} y_s^2(k)} \right)^{1/2} \tag{4}$$

where y_s is the output of the DTM, y_m is the output of the neural network, and N is the length of the data can then be calculated for each output for both the HFTS and the LFTS.

VI. Results

The input–output data are obtained as discussed in Sec. V. An input signal with a clock interval of 1.0 s, sampling time of 0.04 s, and a length of 24 results in an input–output data of length 600 ($24 \times 1/0.04$). The first 300 data values are used as the training set, and the next 300 are used as the testing set to check for "overtraining." The normalized deviation of the input and output data about their steady-state values corresponding to the 90% power level are used in training the feed-forward neural network. For example, the normalized deviations of a signal x can be obtained as follows

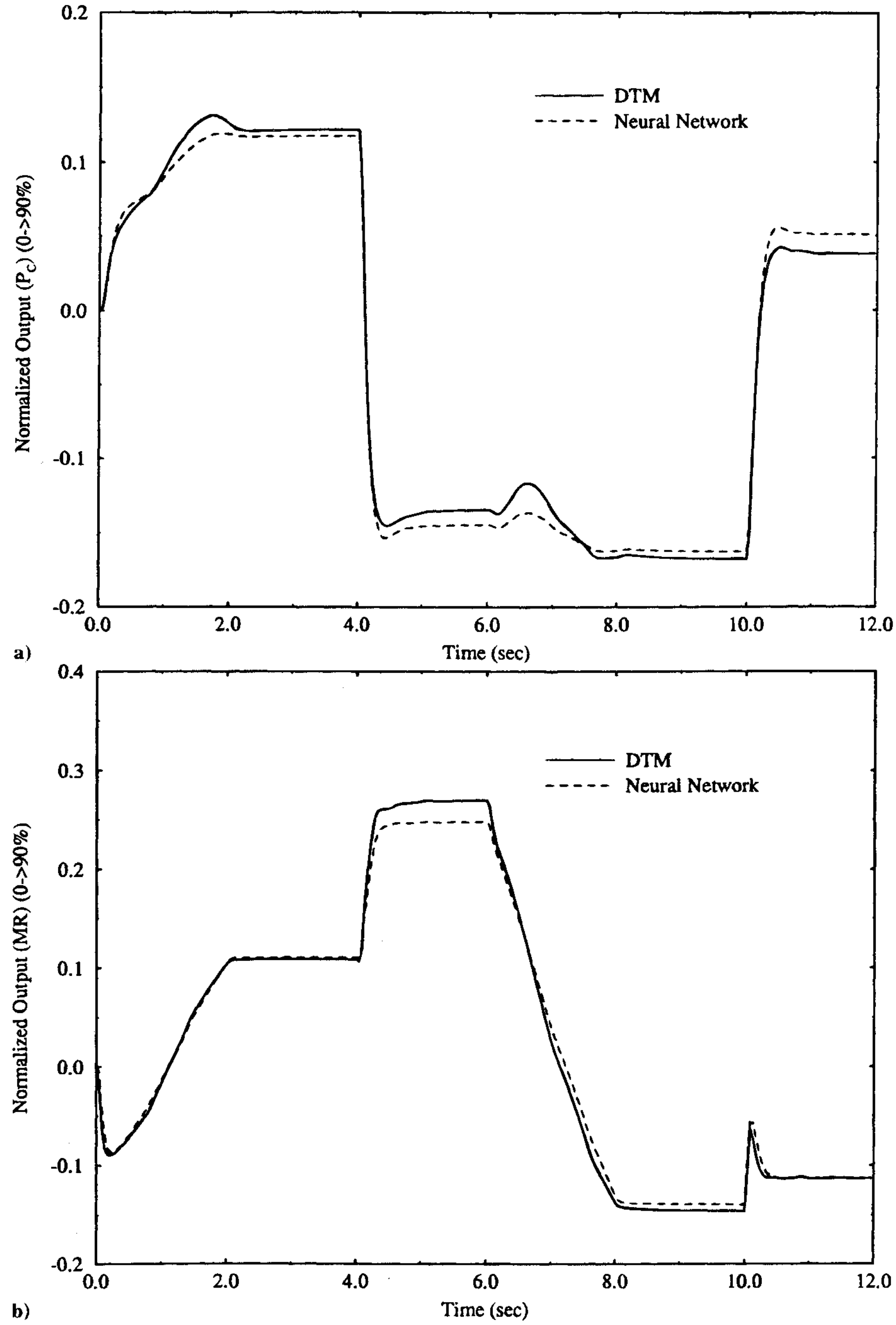

Fig. 4 Comparison of the responses of the neural network model with the nonlinear simulation (DTM), a) P_c chamber inlet pressure; b) *MR* mixture ratio.

$$\delta x = \frac{x - x_{ss}}{x_{ss}}$$

where δx represents the normalized deviations, x_{ss} the steady-state value of x. Using the structure determination algorithm,[2] the values of lags k_u and k_y are calculated. They are found to be equal to one. This means the number of nodes in the input layer is six and corresponds to

$$\delta u_1(t-1),\ \delta u_2(t-1),\ \delta y_1(t-1),\ \delta y_2(t-1),\ \delta y_3(t-1),\ \delta y_4(t-1)$$

and the output layer has four nodes corresponding to

$$\delta y_1(t),\ \delta y_2(t),\ \delta y_3(t),\ \delta y_4(t)$$

where $u_1(\cdot) = \beta_{\mathrm{OPOV}}$, $u_2(\cdot) = \beta_{\mathrm{FPOV}}$, $y_1(\cdot) = P_C$, $y_2(\cdot) = MR$, $y_3(\cdot) = S_{\mathrm{HPFT}}$ and $y_4(\cdot) = S_{\mathrm{HPOT}}$. The number of nodes in the hidden layer is obtained using the incremental procedure explained in Sec. V. It is verified that the network does not overfit the training data by comparing the error of the testing set with the error of the training set. The number of nodes in the hidden layer required to model the system adequately is determined as 12. This results in an overall network with 6 nodes in the input layer, 12 nodes in the hidden layer, and 4 nodes in the output layer (the input layer is fully connected to the hidden layer, which in turn is fully connected to the output layer). This results in a total of 136 weights (including bias connections). The activation function used is given as

$$f(x) = -0.5 + \frac{1}{1 + \exp(-x)}$$

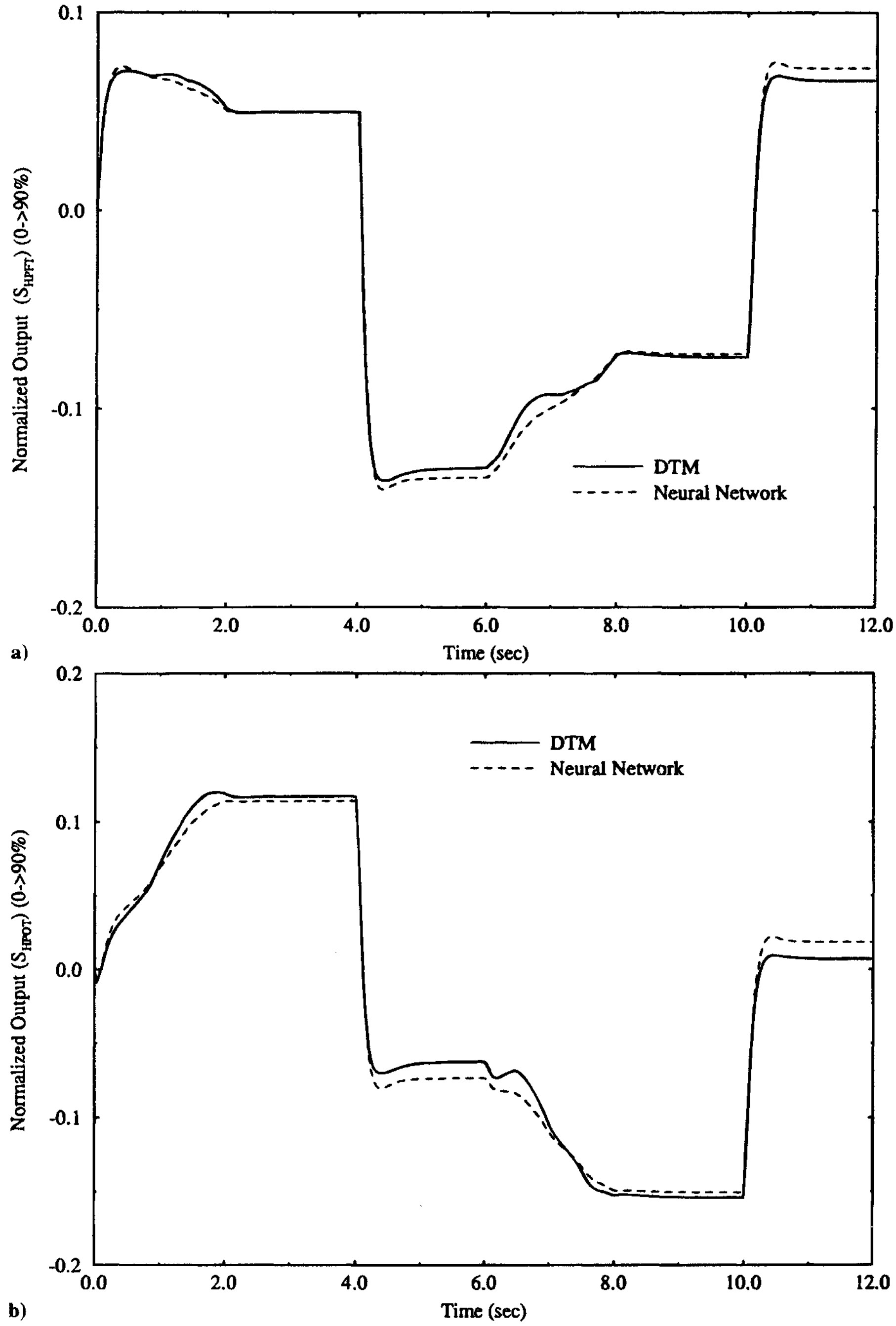

Fig. 5 Comparison of the responses of the neural network model with the nonlinear simulation (DTM). a) S_{HPFT} high-pressure fuel turbine speed; b) S_{HPOT} high-pressure oxidizer turbine speed.

where x is the input (summed signal) going into the node and $f(x)$ is the output of the node. The resulting neural network structure is shown in Fig. 3.

This network is trained with the input–output data (normalized deviations) using the backpropagation algorithm until a sufficient minimization of the error function is achieved; i.e., the sum of squared error for all the patterns in training set is less than 0.001. In this study, the trained neural network is used as a model in the following manner:

$$\delta\hat{y}(t) = F_f[\delta\hat{y}(t-1), \delta u(t-1)] \quad (5)$$

Here, the neural network uses the input $\delta u(t-1)$ (from the DTM) and its own previous estimate $\delta\hat{y}(t-1)$ to predict the output $\delta\hat{y}(t)$. Note that this is different from using both the past inputs and outputs from the DTM to predict the output.

The trained network is tested using the two kinds of test signals explained in Sec. V. The standard error of estimates is calculated for each output for both the HFTS and the LFTS. The SEEs are given in Table 1. For visual comparison, the graphs of the four outputs (normalized) for the LFTS are shown in Figs. 4 and 5.

As seen from Table 1, it is clear that the neural network can model the dynamics of the SSME with sufficient accuracy. It can capture both the high-frequency and low-frequency dynamics of the SSME, as is evident from the SEEs for the HFTS and LFTS, respectively. Figs. 4 and 5 also indicate good agreement between the identified model and the nonlinear simulation for the LFTS.

VII. Conclusions

This work demonstrates that the nonlinear dynamics of the Space Shuttle main engine can be modeled using feed-forward neural networks. The identified neural network model is valid in the 80–100% power level range of the Space Shuttle main engine. Additionally, model accuracy results have shown that the developed neural network model is an accurate and reliable predictor of the highly nonlinear and very complex engine. The comparison of the responses of the nonlinear simulation with the responses of the identified model indicates very good agreement. The identified model can be used for control system design and fault detection and diagnosis purposes.

Acknowledgment

This work was sponsored by the NASA Lewis Research Center, Grant NAG3-1006. The support is greatly appreciated.

References

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

[2] Duyar, A., Eldem, V., Merrill, W. C., and Guo, T.-H., "State Space Representation of the Open-Loop Dynamics of the Space Shuttle Main Engine," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 4, Dec. 1990, pp. 684–690.

[3] Duyar, A., Eldem, V., Merrill, W. C., and Guo, T.-H., "A Simplified Dynamic Model of the Space Shuttle Main Engine," *ASME Journal of Dynamic Systems, Measurement, and Control* (submitted for publication).

[4] Duyar, A., Eldem, V., and Saravanan, N., "A System Identification Approach for Failure Detection and Diagnosis," *Proceedings of the IEEE Workshop on Intelligent Motion Control* (Istanbul, Turkey), Aug. 1990, pp. 61–64.

[5] Guo, T. H., Merrill, W. C., and Duyar, A., "Distributed Approach of Real-Time Diagnostic System Using On-Line System Identification Approach," *Proceedings of the Second Annual Conference on Heath Monitoring for Space Propulsion Systems* (Cincinnati, OH), Nov. 1990, pp. 75–93.

[6] Guo, T. H., Duyar, A., and Merrill, W. C., "A Distributed Failure-Detection and Diagnosis Using On-Line Parameter Estimation," NASA TM 104433.

[7] Anon., "Engine Balance and Dynamic Model," Rockwell International Corporation, Rept. FSCM 02602, Spec. RL00001, Canoga Park, CA, Oct. 1981.

[8] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 79, April 1982, pp. 2554–2558.

[9] Hopfield, J. J., "Neurons with Graded Response have Collective Computational Properties like Those of Two-State Nuerons," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 81, May 1984, pp. 3088–3092.

[10] Chua, L. O., and Yang, L., "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, Vol. 35, Oct. 1988, pp. 1257–1272.

[11] Widrow, B. and Lehr, M. A., "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proceedings of the IEEE*, Vol. 78, No. 9, Sept. 1990, pp. 1415–1442.

[12] Hertz, J., Krogh, A., and Palmer, R. G., "Introduction to the Theory of Neural Computation," *Santa Fe Institute Studies in the Sciences of Complexity*, Vol. 1, Addison-Wesley, Redwood City, CA, 1991.

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

[14] Funahashi, K.-I., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, Vol. 2, 1989, pp. 183–192.

[15] Goodwin, G. C., and Payne, R. L., "Dynamic System Identification: Experiment Design and Data Analysis," *Mathematics in Science and Engineering*, Vol. 136, Academic Press, New York, 1977.

[16] Ljung, L., *System Identification: Theory for the User,* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[17] Fu, F. C., and Farison J. B., "On the Volterra-Series Functional Identification of Nonlinear Discrete-time Systems," *International Journal of Control*, Vol. 18, No. 6, 1973, pp. 1281–1289.

[18] Narendra, K. S., and Gallman, P. G., "An Iterative Method for the Identification of Nonlinear Systems Using a Hammerstein Model," *IEEE Transactions on Automatic Control*, Vol. AC-11, 1966, pp. 546–550.

[19] Leontaritis, I. J., and Billings, S. A., "Input–Output Parametric Models for Non-Linear Systems. Part 1: Deterministic Non-Linear Systems; Part 2: Stochastic Non-Linear Systems," *International Journal of Control*, Vol. 41, No. 2, 1985, pp. 303–344.

[20] Billings, S. A., Chen, S., and Korenberg, M. J., "Identification of MIMO Non-Linear Systems using a Forward-Regression Orthogonal Estimator," *International Journal of Control*, Vol. 49, 1989, pp. 2157–2189.

[21] Chen, S., Billings, S. A., and Luo, W., "Orthogonal Least Suares Methods and Their Application to Non-Linear System Identification," *International Journal of Control*, Vol. 50, 1989, pp. 1873–1896.

[22] Sontag, E. D., "Feedback Stabilization Using Two-Hidden-Layer Nets," *Proceedings of the American Control Conference*, Vol. 1, Inst. of Electrical and Electronic Engineers, NJ, June 1991, pp. 815–820.

[23] Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, March 1990, pp. 4–27.

[24] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Volume 1: Foundations,* The MIT Press, Cambridge, MA, 1986.

[25] Werbos, P. J., "Generalization of Backpropagation with Application to a Recurrent Gas Market model," *Neural Networks* Vol. 1, No. 4, 1988, pp. 339–356.

[26] Williams, R. J., and Zisper, D., "A Learning Algorithm for Continously Running Fully Recurrent Neural Networks," *Neural Computation*, Vol. 1, 1989, pp. 270–280.

[27] Klatt, F. P., and Wheelcock, V. J., "The Reusable Space Shuttle Main Engine Prepares for Long Life," *Proceedings of the Winter Annual Meeting*, edited by J. W., Robinson, American Society of Mechanical Engineers, New York, 1982, pp. 33–44.

[28] Davies, W. D. T., *System Identification for Self-Adaptive Control*, Wiley-Interscience, London, 1970.

[29] Leontaritis, I. J., and Billings, S. A., "Model Selection and Validation Methods for Non-Linear Systems," *International Journal of Control*, Vol. 45, 1987, pp. 311–341.