PREDICTING ENGINE PARAMETERS USING THE OPTICAL SPECTRUM OF THE SPACE SHUTTLE MAIN ENGINE EXHAUST PLUME*

Ashok N. Srivastava
Dept. of Electrical & Comp. Engineering
Center for Space Construction
University of Colorado at Boulder
Boulder, CO 80309-0529
srivasan@colorado.edu

Wray Buntine
NASA Ames Research Center
Moffett Field, CA 94035
wray@ptolemy.arc.nasa.gov

Abstract

The Optical Plume Anomaly Detection (OPAD) system is under development to predict engine anomalies and engine parameters of the Space Shuttle's Main Engine (SSME). The anomaly detection is based on abnormal metal concentrations in the optical spectrum of the rocket plume. Such abnormalities could be indicative of engine corrosion or other malfunctions. Here, we focus on the second task of the OPAD system, namely the prediction of engine parameters such as rated power level (RPL) and mixture ratio (MR). Because of the high dimensionality of the spectrum, we developed a linear algorithm to resolve the optical spectrum of the exhaust plume into a number of separate components, each with a different physical interpretation. These components are used to predict the metal concentrations and engine parameters for online support of ground-level testing of the SSME. Currently, these predictions are labor intensive and cannot be done online. We predict RPL using neural networks and give preliminary results.

1 Optical Spectrum & Engine Parameters

The optical spectrum of the SSME is measured using two dispersing instruments, the first of which is the OPAD spectrometer, which has two 2048 element linear array detectors in the exit plane of a half-meter spectrometer. The second instrument consists of 16 discrete solid-state detectors [Whitaker *et al.*, 1994]. The spectrum is measured at 0.5 second intervals for a test duration of approximately 200 seconds, and consists of readings of intensity as a function of discretized wavelengths. The wavelengths range from 3000-9500 Angstroms. Thus, a single test produces a data matrix Y of size $N \times p$ where $N \approx 400$ and $p \approx 4500$. We were given data from approximately 30 tests, each test being taken under different operating conditions.

Figure 1 shows a typical spectrum for a fixed instant in time. The spectrum normally contains metal signatures (indicated by sharp peaks) at startup. The region between 3000 and 3500 Angstroms is mostly due to hydroxide (OH) emissions from the hydrogen burning engine. Given the spectra, our objective is to predict the corresponding RPL and MR, whose time plots are shown in Figure 2. Notice that the RPL signal is comprised of level shifts, a fact which we will exploit when making predictions.

This prediction problem is difficult because of the high dimensionality of the observable; each observable is of dimension 4500 [1]. In order to obtain good predictions, we developed a method to extract salient features from the spectrum, and then used those features for prediction in a neural network. Previous experience indicates that simply running a learning algorithm such as neural network backpropagation on the raw data can yield poor results.

[1] Note that the observable is a discretized optical spectrum and thus is of high dimension.

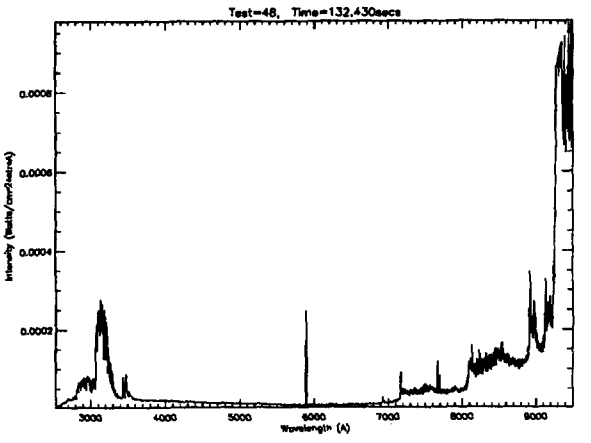

Figure 1: A typical optical spectrum from the exhaust plume of the Space Shuttle Main Engine. The region between 3000 and 3500 Angstroms is mainly due to OH emissions, the spike at 5900 Angstroms is due to sodium, and the region beyond 7000 Angstroms is from water emissions.

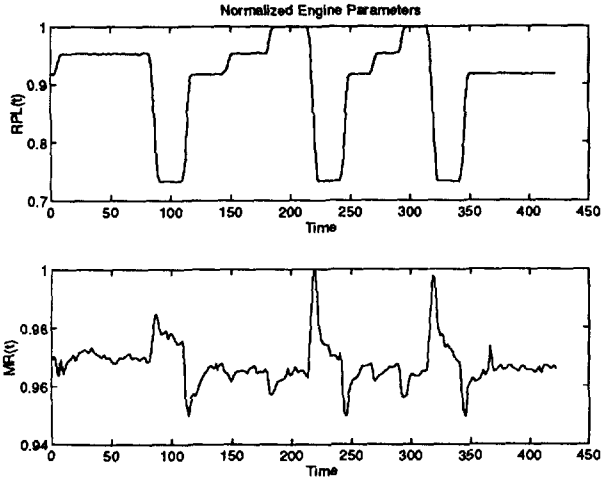

Figure 2: Typical traces of the rated power level (RPL) and the mixture ratio (MR) of the Space Shuttle Main Engine. These traces are difficult to obtain on-line. Our goal is to use the optical spectra to predict these parameters.

2 OH Component Decomposition

We decompose the OH spectrum (ranging from 3000-3500 Angstroms) into m features for each time step using the following method. We begin by assuming a model in which each spectrum $\mathbf{y}_t$ can be written as a linear combination of m stationary components $\mathbf{s}_i$, $i = 1, \ldots, m$, where $\mathbf{s}_i$ is of size $p \times 1$. If we take $S = [\mathbf{s}_1, \mathbf{s}_2, , \ldots, \mathbf{s}_m]$, S is a $p \times m$ matrix, and

$$\mathbf{y}_t = S\mathbf{a}_t + \eta_t, \tag{1}$$

where $\mathbf{a}_t$ is a vector of size $m \times 1$ of time varying coefficients, which we will use as the features in our predictions. The vectors $\mathbf{y}_t$ and η_t are of size $p \times 1$. The vector η_t is assumed to be a zero mean Gaussian random variable with small variance. We compute S and $\mathbf{a}_t$ using the following derivations. It is important to note that a priori, we have no knowledge of S or $\mathbf{a}_t$. In terms of the assumed noise model (which we denote by λ) we have for a given time t,

$$\begin{aligned} P(\lambda \mid \mathbf{y}_t) &\propto P(\mathbf{y}_t \mid \lambda) \\ &\propto \exp -\frac{(S\mathbf{a}_t - \mathbf{y}_t)^T(S\mathbf{a}_t - \mathbf{y}_t)}{2\sigma^2} \end{aligned} \tag{2}$$

Taking the negative log-likelihood, ignoring all constants of proportionality, and summing over $t = 1 \ldots N$, we obtain the standard sum-squared error criterion:

$$C = \sum_{t=1}^{N} (S\mathbf{a}_t - \mathbf{y}_t)^T(S\mathbf{a}_t - \mathbf{y}_t) \tag{3}$$

We minimize this quadratic form in the usual manner by computing the solution to the corresponding normal equations (the solutions are given below). Minimizing this quadratic form will not necessarily yield orthogonal components $\mathbf{s}_i$. In order to obtain orthogonal solutions, (so that $S^T S \approx I_m$, where I_m is the $m \times m$ identity matrix) we use the following iterative algorithm. We compute each column of the matrix S (denoted by $\mathbf{s}_i$) separately in a manner which guarantees orthogonal solutions. If we denote the i^{th} component of the vector $\mathbf{a}_t$ by a_t^i, we have the following algorithm:

Algorithm for OH Component Decomposition

1. *Loop* for $i = 1, \ldots, m$ do:
2. *Initialize* $\mathbf{s}_i$ = random(p,1). Obtain a random $p \times 1$ vector.

3. *Repeat*

$$a_t^i = (\mathbf{s}_i^T \mathbf{s}_i)^{-1} \mathbf{s}_i^T \mathbf{y}_t$$

$$\mathbf{s}_i = \sum_{t=1}^{N} \mathbf{y}_t a_t^i (\sum_{t=1}^{N} (a_t^i)^2)^{-1}$$

$$C = \sum_{t=1}^{N} (\mathbf{s}_i a_t^i - \mathbf{y}_t)^T (\mathbf{s}_i a_t^i - \mathbf{y}_t)$$

4. *Until* $C < \epsilon$, where ϵ is a user defined positive stopping criterion.

5. *Compute* $\mathbf{y}_t \leftarrow \mathbf{y}_t - \mathbf{s}_i a_t^i$ for all $t = 1, \ldots, N$

6. *End loop.*

In step 5, we replace the current value of $\mathbf{y}_t$ with the residuals and repeat the loop. This step insures that the OH components will be orthogonal. This algorithm converges to a solution quickly (no more than 5 iterations), and can be shown to produce results similar to the singular value decomposition of the data matrix.

After the algorithm has been executed, we are left with a factorization of the data matrix $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T$ which is an $N \times p$ matrix such that $Y = (SA)^T$, where A is an $m \times N$ matrix, and S is a $p \times m$ matrix with $S^T S \approx I_m$.

We interpret the A matrix as a matrix of features for the data matrix Y. Since we assume that the matrix of components S does not change with time or test set, once we have obtained an S matrix, we can use it to compute feature matricies A. The rows of the feature matricies are used for predicting the engine parameters.

Figure 3 shows a typical value of S for $m = 3$. Each panel corresponds to a different column of the S matrix. These components contain greater than 99.99% of the variance in the spectral signal. Since a high percentage of the variance is captured, approximating each optical spectrum by a linear combination of these components is not a poor approximation.

The first component captures the greatest variability in the signal, and has the same order of magnitude as the original signal. Each subsequent component successively captures the smaller variability in the signal. The second component is at the level of the noise in the system, while the remaining component is a factor of 10 *below* the noise level in the system. The third component corresponds to known sinusoidal fluctuations in the measurement device.

Figure 4 shows the variation of these components with time. Each panel corresponds to a different row

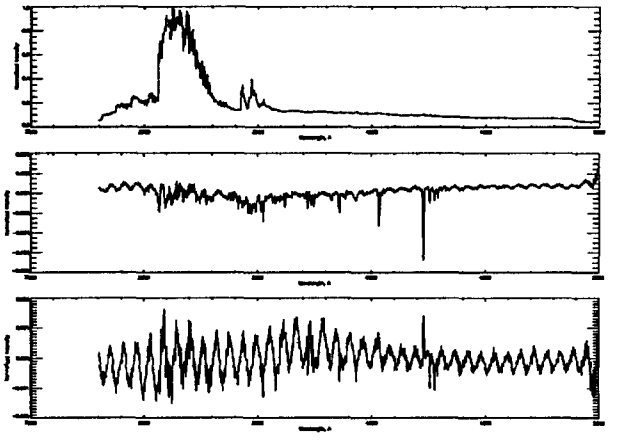

Figure 3: After applying our algorithm to the spectral data, we found that the spectral signals could be decomposed into a linear combination of the above normalized components.

in the feature matrix A. The features exhibit level shifts, a fact which will aid in predicting the rated power level of the engine.

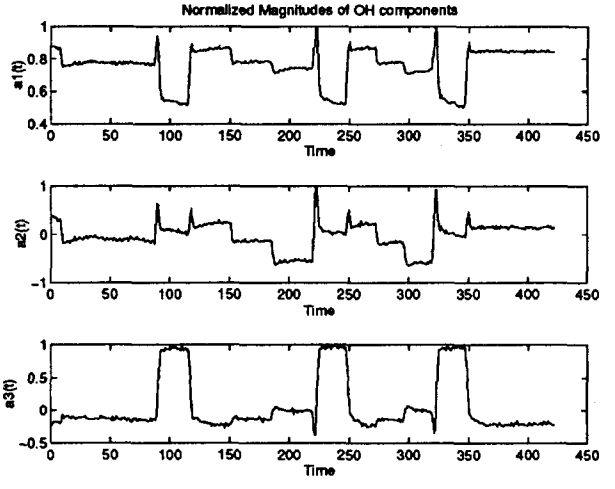

Figure 4: This plot shows how the components in the previous figure vary with time. These variations will be used for the prediction of the engine parameters.

Comparison with Principal Components

Our algorithm does not yield the same results as standard principal component analysis (PCA) [Rao, 1965]. PCA yields the eigenvectors of the covariance matrix

$$\Delta = Y^T Y - (\frac{1}{n} Y^T \mathbf{j})(\frac{1}{n} \mathbf{j}^T Y) \qquad (4)$$

where $\mathbf{j}$ is an appropriately sized column vector of ones. The covariance matrix is centralized to have zero mean, whereas the original data matrix Y is not centralized. Our algorithm does not operate on the covariance matrix, but directly on the data matrix Y. We did not use PCA because the matrix Δ is of size $p \times p$ (which is about 4500×4500 in this example) which could have led to numerical or computational problems.

3 Prediction with Neural Networks

When mapping from the optical spectrum to the engine parameters, we are implicitly assuming that the spectrum contains sufficient information to reconstruct the parameters. More precisely, we are assuming that the system has the following form:

$$\begin{aligned} \dot{\mathbf{x}}_t &= F(\mathbf{x}_t, \theta, \mathbf{u}_t, \epsilon_t) \\ \mathbf{y}_t &= G(\mathbf{x}_t, \eta_t) \end{aligned} \tag{5}$$

where $\dot{\mathbf{x}}_t$ denotes the derivative of the system's state with respect to time, $\mathbf{x}_t \in \mathbf{R}^n$, $G \in \mathbf{R}^p$, u is a possible input into the system, ϵ_t process noise, η_t measurement noise, and $\mathbf{y}_t$ is the observable. The model parameters are $\theta \in \mathbf{R}^q$. This system is potentially nonlinear and time-varying. In our case, we consider a restricted state $\mathbf{x}_t = [\mathrm{RPL}_t, \mathrm{MR}_t]^T$ and assume that the observable $\mathbf{y}_t$ contains the information necessary to reconstruct the state $\mathbf{x}_t$. We implicitly assume that G is an unknown, possibly nonlinear, invertible function, which is in contrast with the classical *observer* problem, where G is assumed to be noninvertible. Said differently, we are assuming that the observable $\mathbf{y}_t$ contains the necessary information to reconstruct $\mathbf{x}_t$.

For the prediction, we reconstruct an element of the state $\mathbf{x}$ given example pairs $(\mathbf{x}, \mathbf{y})$ without knowing the nature of the functions F and G or the model parameters θ. This problem lends itself naturally to a neural network approach, since the networks can be designed to learn the 'inverse' function G^{-1}. Although the mapping G may not have a unique inverse, it can be shown that, given enough training examples, the neural network will converge to a solution such that $\mathbf{x} = G^{-1}(\mathbf{y})$ [Jordan and Rumelhart, 1992].

Our preprocessing algorithm reduces the dimensionality of $\mathbf{y}_t$ and leaves us with the easier problem of finding a mapping from the feature matrix A to the engine parameters. In this discussion, we will focus on prediction of the rated power level (RPL). We are developing methods to predict the mixture ratio (MR) which is expected to be a more involved prediction task.

Neural Network Design

Neural networks are mathematical structures that can learn from implicit or explicit rules. These are a powerful class of models, and have been used for a variety of applications such as speech recognition [Nadas and Mercer, 1995], control [Miller *et al.*, 1990], and time series prediction [Weigend, 1995]. In the late 1980's, [Cybenko, 1989] and several other independent researchers [Hornik *et al.*, 1989, Funahashi, 1989] produced a celebrated result which showed that if the number of hidden units in the hidden layer of a two layer network is sufficiently large, the network can approximate any continuous function to any degree of accuracy over a compact set.

Figure 5 shows a diagram of a full multilayer neural network. This network is comprised an input layer, a hidden layer (so called because the user only indirectly gives this layer an input), and an output unit. Although we are only considering networks with a single hidden layer, it is possible to include more hidden layers.

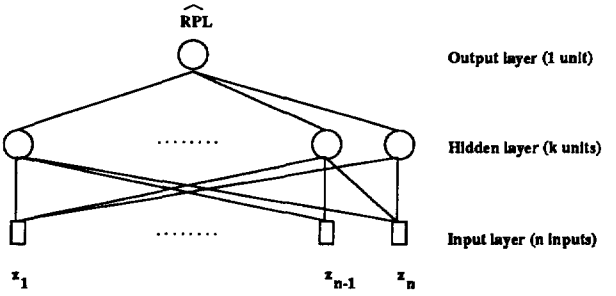

Figure 5: A multilayer, feedforward neural network for estimating the RPL of the SSME. The network consists of n inputs, k hidden units (tanh) and one output corresponding to the estimate $\widehat{\mathrm{RPL}}_t$. The network is trained using the backpropagation algorithm.

The neural network produces an estimate of the RPL_t, $\widehat{\mathrm{RPL}}_t$, using the following nonlinear function:

$$\widehat{\mathrm{RPL}}_t = \psi(W_2 \sigma(W_1 \mathbf{z} + B_1) + B_2) \tag{6}$$

where $\mathbf{z} \in \mathbf{R}^n$ is the input to the network, $W_1 \in \mathbf{R}^{n \times k}$ is the weight (or parameter) matrix between the input layer and the hidden layer, $B_1 \in \mathbf{R}^{k \times 1}$ is the corresponding bias vector, $W_2 \in \mathbf{R}^{k \times 1}$ are the weights between the hidden layer and the output layer, with $B_2 \in \mathbf{R}^{m \times 1}$ being the corresponding

biases. The function $\sigma(\cdot) \in \mathbf{R}^k$ is the tanh function and $\psi(\cdot) \in \mathbf{R}^1$ could be a linear or nonlinear function. The neural network is thus a well-defined functional mapping from $\mathbf{R}^n \mapsto \mathbf{R}^1$.

Details of the RPL Prediction Network

The following are the important elements of the network which we used to predict the RPL of the SSME.

- **Inputs.** The neural network had 9 inputs corresponding to $[\mathbf{a}_t^1, \mathbf{a}_t^2, \mathbf{a}_t^3, \ldots, \mathbf{a}_{t-2}^1, \mathbf{a}_{t-2}^2, \mathbf{a}_{t-2}^3]$. The inputs consist of the past three values of each row of the feature matrix. These inputs are used to predict RPL_t. We used three past values of the feature vectors (as opposed to using a single value) in order to make the network robust to noise in the feature vectors.

- **Hidden Layer.** 8 tanh hidden units.

- **Output Layer.** 1 sigmoidal unit. [2] We chose to use the sigmoidal output unit instead of a linear unit because the RPL is a piecewise constant function. Thus, we can view the problem as one of *classifying* the input into the correct output. The choice of a sigmoidal output unit is appropriate for classification problems such as this [Chauvin and Rumelhart, 1993]. Usually, the sigmoidal output unit is used for two-class classification problems. In our problem, however, we have a continuum of classes. Since the range of the sigmoidal function is between 0 and 1, we scaled the target data to fall within this range.

- **Training and Cost Function.** Sum-squared error between the estimated and the actual RPL using the backpropagation algorithm [Chauvin and Rumelhart, 1993] with no momentum term.

$$C = \sum_{t=1}^{N} (\widehat{\mathrm{RPL}}_t - RPL_t)^2 \qquad (7)$$

4 Results

Figure 6 shows the performance of the neural network for predicting the RPL. The top panel shows the performance on the training set, and the other two panels show out-of-sample performance. Each panel contains three curves. The solid curve is the actual RPL signal, the dotted line is the neural network's prediction, and the dash-dot line is the prediction of a simple linear predictor. The linear predictor generates a linear mapping from the input vector $[\mathbf{a}_t^1, \mathbf{a}_t^2, \mathbf{a}_t^3, \ldots, \mathbf{a}_{t-2}^1, \mathbf{a}_{t-2}^2, \mathbf{a}_{t-2}^3]$ to the output RPL. Listed above each panel is the ratio of the sum squared errors of the linear predictor to the neural network predictor.

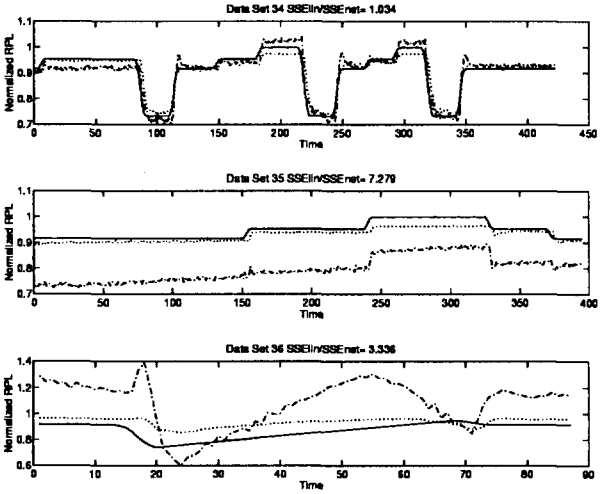

Figure 6: This figure shows the performance of the neural network for predicting the RPL of the Space Shuttle Main Engine using the features produced by our algorithm for three different test sets. The first panel is the training set, and the lower two panels show out-of-sample performance. The solid line corresponds to the actual RPL signal, the dotted line is the neural network's prediction, and the dot-dash line is the output of a simple linear predictor.

The performance of the two predictors is comparable for the training set (first panel), but the network has higher out-of-sample performance. The bottom panel in the figure also illustrates that the network can predict the trend of the RPL signal more reliably than the linear predictor.

5 Discussion

We have demonstrated the use of a neural network for predicting the rated power level (RPL) of the Space Shuttle Main Engine (SSME) using the optical spectrum of the exhaust plume. The problem is challenging because the observable is of high dimension. We resolved the spectrum into a number of orthogonal components, and then expressed each observable as a linear combination of the components. The coefficients in this linear combination were used

[2] The sigmoidal function is defined as $f(x) = \frac{1}{1+e^{-x}}$.

as inputs to the network.

The algorithm which we designed delivers orthogonal component vectors. One direction of future research will be to modify our algorithm to create orthogonal feature vectors. These features may yield better predictions. Another area of research will be to analyze other, more complex, linear predictors and compare their performance to our neural network.

We will use similar strategies to those discussed here to predict the mixture ratio (MR) of the engine. That signal is expected to be more difficult to predict than the RPL, mainly because it is not piecewise constant. The neural network predictors for RPL has been implemented in a real-time system at the Marshall Space Flight Center; however, at the time of publication, no results are yet available.

References

[Chauvin and Rumelhart, 1993] Y. Chauvin and D. E. Rumelhart, editors. *Backpropagation: Theory, Architectures and Applications.* Lawrence Erlbaum, 1993.

[Cybenko, 1989] G. Cybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[Funahashi, 1989] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[Hornik *et al.*, 1989] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[Jordan and Rumelhart, 1992] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.

[Miller *et al.*, 1990] T. W. Miller, R. S. Sutton III, and P. J. Werbos. *Neural Networks for Control.* MIT Press, Cambridge, MA, 1990.

[Nadas and Mercer, 1995] A. Nadas and R. L. Mercer. Hidden markov models and some connections with artificial neural networks. In P. Smolensky, M. C. Mozer, and D. E. Rumelhart, editors, *Mathematical Perspectives on Neural Networks.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[Rao, 1965] C. R. Rao. *Linear Statistical Inference and its Applications.* John Wiley and Sons, New York, 1965.

[Weigend, 1995] A. S. Weigend. Time series analysis and prediction. In P. Smolensky, M. C. Mozer, and D. E. Rumelhart, editors, *Mathematical Perspectives on Neural Networks*, Hillsdale, NJ, 1995. Lawrence Erlbaum Associates.

[Whitaker *et al.*, 1994] K. W. Whitaker, K. Krishnakumar, R. V. Ravikrishna, and R. C. Lattus. Predicting species concentrations in the ssme plume using neural networks. *Unpublsihed*, 1994.