1996
NASA/ASEE FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

SSME CONDITION MONITORING USING NEURAL NETWORKS AND PLUME SPECTRAL SIGNATURES

Prepared by: Randall Hopkins, M.S. AE
Daniel Benzing, B.S. AE

Academic Rank: Instructor
Graduate Student

Institution and Department: The University of Alabama at Tuscalloosa
Department of Aerospace Engineering

NASA/MSFC:

Laboratory: Astrionics Lab
Division: Propulsion Systems
Branch: Instrumentation

MSFC Collegue: Anita Cooper, W.T. Powers

Introduction

For a variety of reasons, condition monitoring of the Space Shuttle Main Engine (SSME) has become an important concern for both ground tests and in-flight operation. The complexities of the SSME suggest that active, real-time condition monitoring should be performed to avoid large-scale or catastrophic failure of the engine.

In 1986, the SSME became the subject of a plume emission spectroscopy project at NASA's Marshall Space Flight Center (MSFC). Since then, plume emission spectroscopy has recorded many nominal tests and the qualitative spectral features of the SSME plume are now well established. Significant discoveries made with both wide-band and narrow-band plume emission spectroscopy systems led MSFC to develop the Optical Plume Anomaly Detection (OPAD) system.

The OPAD system is designed to provide condition monitoring of the SSME during ground-level testing. The operational health of the engine is achieved through the acquisition of spectrally resolved plume emissions and the subsequent identification of abnormal emission levels in the plume indicative of engine erosion or component failure. Eventually, OPAD, or a derivative of the technology, could find its way on to an actual space vehicle and provide in-flight engine condition monitoring. This technology step, however, will require miniaturized hardware capable of processing plume spectral data in real-time.

An objective of OPAD condition monitoring is to determine how much of an element is present in the SSME plume. The basic premise is that by knowing the element and its concentration, this could be related back to the health of components within the engine. For example, an abnormal amount of silver in the plume might signify increased wear or deterioration of a particular bearing in the engine. Once an anomaly is identified, the engine could be shut down before catastrophic failure occurs.

Currently, element concentrations in the plume are determined iteratively with the help of a non-linear computer code called SPECTRA, developed at the USAF Arnold Engineering Development Center. Ostensibly, the code produces intensity versus wavelength plots (*i.e.*, spectra) when inputs such as element concentrations, reaction temperature, and reaction pressure are provided. However, in order to provide a higher-level analysis, element concentration is not specified explicitly as an input. Instead, two quantum variables, number density and broadening parameter, are used.

Past experience with OPAD data analysis has revealed that the region of primary interest in any SSME plume spectrum lies in the wavelength band of 3300 Å to 4330 Å. Experience has also revealed that some elements, such as iron, cobalt and nickel, cause multiple peaks over the chosen wavelength range whereas other elements (magnesium, for example) have a few, relatively isolated peaks in the chosen wavelength range.

Iteration with SPECTRA as a part of OPAD data analysis is an incredibly labor intensive task and not one to be performed by hand. What is really needed is the "inverse" of the computer code but the mathematical model for the inverse mapping is tenuous at best. However, building

generalized models based upon known input/output mappings while ignoring details of the governing physical model is possible using neural networks.

Thus the objective of the research project described herein was to quickly and accurately predict combustion temperature and element concentrations (*i.e.*, number density and broadening parameter) from a given spectrum using a neural network. In other words, a neural network had to be developed that would provide a generalized "inverse" of the computer code SPECTRA.

Radial Basis Function Networks

With most neural networks the objective is to estimate a function $y(x)$ from a training set of representative input/output pairings:

$$\left\{\left(\vec{x}_i, \vec{y}_i\right)\right\}_{i=1}^{P}$$

It has been shown that traditional backpropagation (BP) neural networks can sufficiently perform this mapping with only two hidden layers provided the function is well behaved over the domain of interest. The required hidden layers are usually composed of neurons with sigmoidal activation functions which combine to form localized "bumps" or response regions within the input space. In this manner, the network paves the input space with response bumps that are only non-zero in a small region. Instead of using two hidden layers of sigmoidal units, this local behavior could also be obtained with a single hidden layer of radial basis functions. By definition, a radial basis function is one which decreases monotonically away from a central point thereby giving it an inherent bump form. Classic functions that exhibit this propensity are the Gaussian, Cauchy, and the Inverse Multiquadric. For RBF networks, the choice of function does not make much difference as long as it possesses the local response region. The Gaussian function is written as:

$$g_j(\vec{x}) = -\exp\left[\frac{-(\vec{x}-\mu_j)^2}{2\sigma_j^2}\right]$$

Thus, a given hidden RBF neuron will be centered at μ_j within the input space and have a "receptive field" which is proportional to σ_j. Moreover, it will give a maximum response for input vectors ($\vec{x}$) which are near the center (μ_j).

The goal of the RBF algorithm is to arrange an assortment of these receptive fields so that response areas are created which sufficiently cover the input space. This is done in a two part learning scheme known as hybrid learning. The initial layer of the network contains the RBF centers which are obtained through unsupervised assimilation. This is called unsupervised because there is not a well defined learning goal; instead categories must be developed from correlation's within the input training data. The layer essentially clusters the inputs and specifies where to position the RBF centers so that the desired response coverage is obtained. Having a layer with unsupervised learning offers a significant savings in training time over the traditional BP scheme. However, the hybrid scheme is not optimal in the sense that the BP algorithm is. The reason for this has to do with the non-linear nature of the BP front-end layer. The weight parameters which contribute to the hidden layer responses are adjusted with respect to the performance of the outputs. In contrast, the RBF centers are chosen *a priori* and remain fixed throughout the output layer training. As a result, RBF networks normally require more neurons

to attain the same error goal. This disadvantage, however, is usually outweighed by the savings in training time.

The output layer of the RBF network is trained in a supervised fashion. Supervised means that the learning is based on comparison of the network output with the known "correct" answers. Thus, given that the basis function centers are fixed (making this a *linear* network), the optimal weight array for the output layer which gives the proper functional mapping can be found using multiple linear regression. Details of this procedure can be found in any decent regression book; the results are simply stated here. For a set of training input vectors, x, with corresponding RBF centers μ, there will be an array of Gaussian neuron responses, **G**. Given this, the optimal weight array can be stated as:

$$\mathbf{w} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{y}$$

where, **y** is the corresponding target values. The mathematical form of the weight array is also known as the normal equation in classical regression theory.

Network Performance

Separate RBF network's for the monitoring of each metal were created using training data generated from the aforementioned SPECTRA code. To test the validity of the network configuration a set of spectral data obtained during a January 1996 SSME Failure event at Stennis Space Flight Center were used. The spectral scans were taken every half second until the engine failure. Post test analysis revealed that the major spectral anomalies occurred at the times shown in Table 1 given below.

Table 1.
Major Anomalous Event Times (Seconds)

130-131
276
283
404-405.5
531-Engine Failure

The neural networks were used to evaluate plume temperature, elemental number densities, and broadening parameters for every half second of data. The results are shown in Figure 1 on the next page. Temperature predictions were near perfect; a quick study of the number density predictions shows that all the major anomalies were indicated by number density spikes at the appropriate times specified in Table 1. Figure 1 represents two years of concerted effort and can speak justly of this entire paper. The neural networks passed the anomaly detection objective. Future work will involve the extension of the technology to real time flight systems.

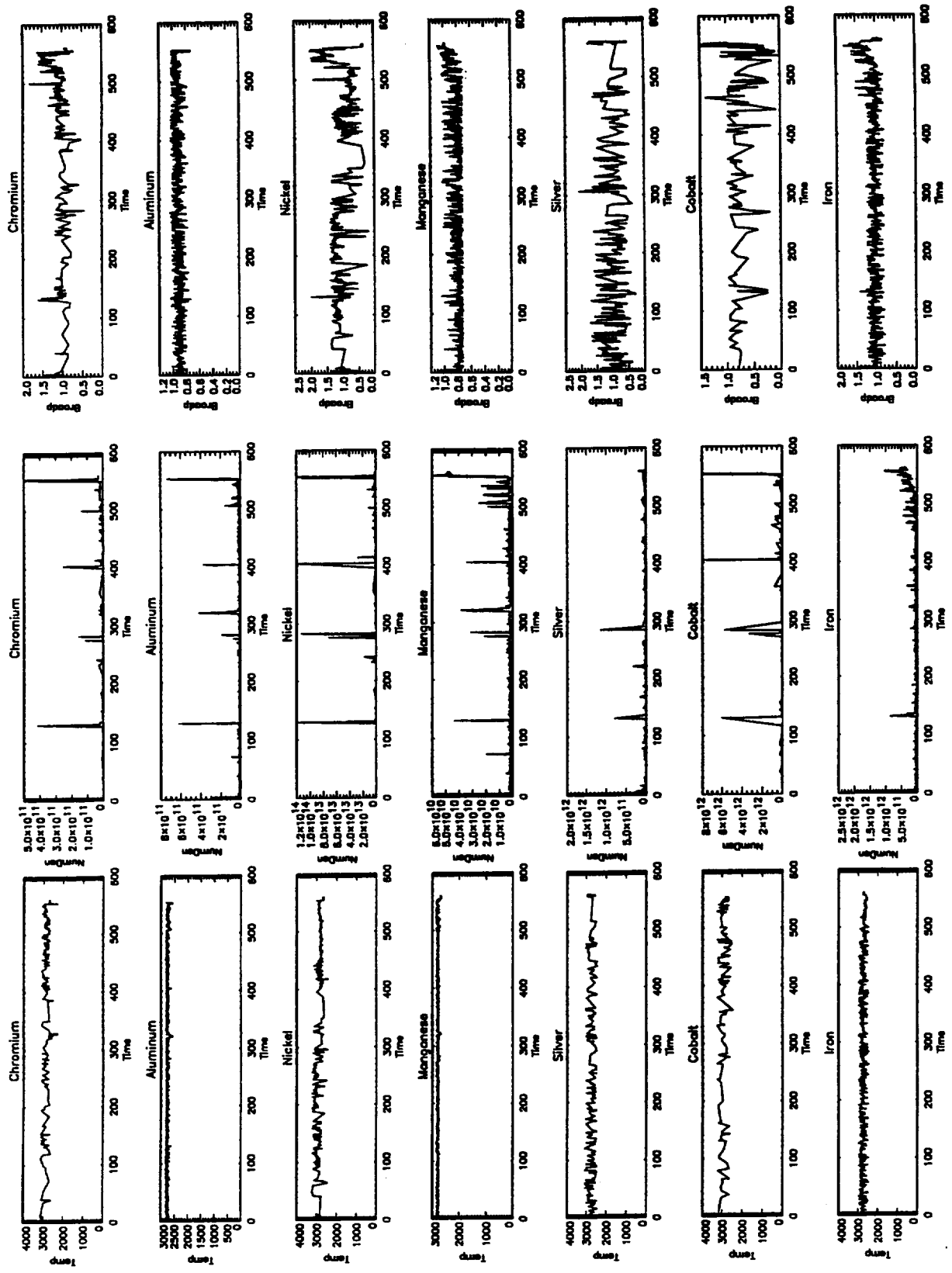

Figure 1.
Neural Predictions for the Stennis Failure Event