ESTIMATING GAS TURBINE INTERNAL CYCLE PARAMETERS USING A NEURAL NETWORK

Nicolas W. Chbat and Ravi Rajamani
Control Systems & Electronic Technologies Laboratory
GE Corporate Research & Development
Schenectady, New York

Todd A. Ashley
Controls, Accessories & Systems Engineering
GE Power Generation Engineering
Schenectady, New York

ABSTRACT

We show that a neural network can be successfully used in place of an actual model to estimate key unmeasured parameters in a gas turbine. As an example we study the combustion reference temperature, a parameter that is currently estimated via a nonlinear model inside the controller and is used in a number of critical mode-setting functions within the controller, such as calculating the fuel-split between various manifolds. We show that a feedforward neural network using simple backpropagation learning can be built for estimating combustion reference temperature. The neural network matches the accuracy of the current estimate; and it is more robust to errors in its internal parameters. This is advantageous from the point of view of implementation since a number of errors creep in when running the algorithm on a digital controller, and an estimator that is not robust with respect to such errors can degrade the performance of the whole system.

INTRODUCTION

When controlling a gas turbine, a number of critical system parameters cannot be measured reliably; hence estimates are made based on the available sensor data. One such parameter is the combustion reference temperature (TTRF) that is calculated by feeding sensor measurements into a simplified aero-thermal equations for the turbine. While not related to any physical temperature in the turbine, this parameter is nevertheless an important control variable, since a number of critical functions are governed by it. Dynamically the TTRF correlates well with the average fuel-air-ratio in the combustor and hence is used to control the division of fuel going to various parts of the combustor. For instance, in Figure 1 the split between the primary and the secondary fuel manifolds is governed by TTRF. It is therefore critical that accurate information about this parameter be available to the controller at all times. The model that presently calculates TTRF is usually represented in the controller as a set of nonlinear equations, with sensed parameters such as the
compressor discharge pressure, turbine exhaust temperature, exhaust air flow, ambient temperature, and guide vane angle, serving as inputs. Since the system is nonlinear, even complex models can accurately estimate this parameter only over a limited range of operating points. The reality of dealing with such a highly nonlinear plant led us to consider neural networks as an alternative methodology for accomplishing the same estimation. We summarize our results in this paper. In the first section we describe the process that we are trying to estimate. We follow this with a brief discussion on neural network model estimation and the backpropagation training method. In the final section we present some results to back up our claims.

A number of researchers have used neural networks in the diagnosis and control of gas turbines—but not many have been implemented commercially. Song, et al. (1994) consider the use of an integrated robust neural controller for a gas turbine. Pomfret (1994) and Brownell (1992) consider neural nets for use in diagnosis in aircraft engines. Sakawa, et al. (1993), use a fuzzy logic/neural networks based method to predict NOx in a gas turbine, a problem that is critical to continuous emissions monitoring.

COMBUSTION REFERENCE TEMPERATURE

A schematic of a low emissions combustor design is given in Figure 1; this is referred to as a Dry Low NOx combustor since it does not depend on the injection of steam or water to reduce the formation of polluting gases such as Nitrogen Oxides and Carbon Monoxide. Compressor discharge air enters mainly through the head-end of the combustor. Fuel is split between the primary zone and the secondary zone, in proportions varying with the operating point. Primary fuel is mixed with the air before it reaches the burning zone. Pre-mixing the ingredients of combustion before burning them is the key to reducing the formation of unwanted byproducts. The turbine controller uses the combustion reference temperature to control the split between the primary and secondary zones. This is critical to maintaining safe and stable operation of the combustor.

The set of equations that estimate $TTRF$ in the controller are complex, and their exact nature is not relevant to the discussion here. Let $TTRF_i$ designate this estimate of $TTRF$. We can write it as the output of a nonlinear map

$$TTRF_i = f(CPD, TTXM, TCD, WEXH). \quad (1)$$

The exhaust air flow, $WEXH$, is estimated if instrumentation for measuring it is not available. What is relevant, however, is the computational complexity of the algorithm. The present algorithm requires roughly 55 multiplies and divides, 45 additions and subtractions, and sundry other mathematical and logical operations. We will compare these numbers with those for the neural network in the section on results.

NEURAL NETWORKS

A Neural Network (NN) is a nonlinear estimator which can be trained to map inputs to outputs in a framework that very loosely mimics how learning is performed in the brain (Hassoun, 1995). The NN used in this application is shown in Figure 2.

![Figure 2: Neural Network for TTRF Estimation](image)

It has a feedforward structure, i.e. no backward or lateral connections. This is especially helpful when implementing the NN on a real-time controller. This network has four input nodes (or neurons), four hidden (or middle) nodes, and one output node. To avoid clutter only a subset of the lines are shown; the actual network is fully connected. The hidden node $h$ is identified with the sigmoid function, and its output is

$$o_h = \frac{1}{1 + e^{\exp(-\sum_{i=1}^{n_i} W_h i_i)}}. \quad (2)$$

Since in this application each input neuron $(i)$ has a linear threshold function associated with it, its output is the same as its input. The characterizing function of the output neuron $(j)$ is the weighted sum of its inputs:

$$o_j = \sum_{h=1}^{n_h} W_{jh} o_h. \quad (3)$$

Generally an NN operates in two phases, a learning phase, and an operating phase. The purpose of the learning phase is to determine (or tune) the NN parameters which will enable the network to function properly...
in the operating phase. A learning algorithm is usually used in the learning phase.

The parameters of an NN are the weights which represent the strengths of the connections between nodes, and the thresholds (or biases) of the nodes. The topology of the NN defines the number of parameters to be tuned. Hence there are \( n_a(n_i + n_o + 1) \), here 24, parameters to be tuned.

Backpropagation (BP) is used for training the net. An abbreviated explanation of this method follows—for details refer to Hassoun (1995) or Rumelhart and Williams (1986). The BP is a steepest-descent optimization technique, where at every iteration the new set of weights is an improvement over the old set towards the direction of the minimum of \( E \), the error function, defined as

\[
E = \frac{1}{2} \sum_{j=1}^{n_a} (d_j - o_j)^2.
\]

At every iteration, the weights are updated according to:

\[
W_{jh}^{N+1} = W_{jh}^N + \Delta W_{jh}
\]

where

\[
\Delta W_{jh} = -\frac{\partial E}{\partial W_{jh}} = (d_j - o_j) o_h,
\]

and,

\[
W_{hi}^{N+1} = W_{hi}^N + \Delta W_{hi}
\]

where

\[
\Delta W_{hi} = -\frac{\partial E}{\partial W_{hi}} = \sum_{j=1}^{n_a} ((d_j - o_j) W_{hi}) o_h(1 - o_h)i_i
\]

Starting with an arbitrary set of weights, a forward pass is first attempted, i.e., the first input pattern is presented to the input layer and \( o_1 \) (note: \( j = 1 \)) is computed for the output neuron. The weights \( W_{jh} \) are then updated according to equations (5) and (6). Next, moving backward to the adjacent layer, \( W_{hi} \) are updated according to equations (7) and (8). The repetition of this process for all the patterns marks the completion of one epoch. Iterations continue until either the desired accuracy between \( o_j \) and \( d_j \), or the maximum allowable number of epochs, is achieved. At this point the network is dubbed ‘trained,’ and is ready for the operating phase.

Tests with different learning algorithms were also attempted in this study. A test of the neural network with BP with momentum learning yielded results similar to those of the simple BP described here. A novel method that adaptively tunes the power of the step size in a line search technique is currently under investigation.

**RESULTS**

The experiment consisted of first training the NN to predict \( TTRF \) for a set of data points consisting of roughly a third of the total data. Once trained using the BP algorithm, the net was presented with the full data set, and the predicted values were compared to the actual ones. The actual \( TTRF \) was obtained from a validated thermodynamic model of the turbine cycle.

This model was run so that it converged to a selected \( TTRF \), for a given operating point. Six different values of \( TTRF \) were chosen within the operating range of the turbine, and at each value, data for 12 operating points were calculated by varying various input parameters to the model. These 72 cases formed the total data set. The training set was selected at random from these 72 cases, in this preliminary study. We will revisit the choice of the training set as we make the estimation algorithms more sophisticated.

In terms of computational complexity, the neural network uses approximately 40 multiplies, 50 additions, and 4 look-up tables to approximate the threshold function. As indicated above, the conventional model uses 55 multiplies and 45 additions, and a host of logical (if-then) statements. Because there are no logical blocks in the neural network implementation, except in the look-up table, the computational speed of the neural network algorithm is higher than that of the conventional model.

Figure 3 shows the output of the neural network for all 72 test cases, plotted as a function of the operating points. The mean squared error (MSE) is calculated as follows:

\[
MSE = \left( \sum_{i=1}^{n} ||TTRF - TTRFn_i||^2 \right)^{0.5}.
\]

The MSE of 0.2% is about 10% better than that of the conventional model while the maximum error of 0.464% is nearly the same. However, the main difference between the two approaches arises when we consider the robustness of the model to variations in the model parameters.

A detailed sensitivity analysis of the algorithm has not yet been carried out. Nevertheless, we ran a few tests that suggest that on the average, the neural network is more robust than the conventional model. These tests consisted of perturbing the weights and the biases in the neural network model in a random manner. The perturbation was uniformly distributed between 0% and 1%. Figure 4 shows the output of one such run. After 50 runs with randomly selected perturbations, the average increase in MSE was about 1%, and in the maximum error was about 1.5%. The corresponding values for the conventional model were 2% and 3%, respectively, suggesting that while the NN model delivers the same accuracy as the conventional model, it is less sensitive to errors in the model parameters.

From an engineering point of view, this gives us a significant advantage, since implementing these algorithms
on finite-word length digital controllers leads to truncation and round-off errors. In the case of the neural network, the estimation accuracy is not affected as much as in the conventional model.

CONCLUSIONS

The combustion reference temperature is a critical parameter that is used in a number of functions within the controller. Currently it is estimated via a nonlinear model implemented in the controller. In this paper we demonstrate that a neural network provides an alternate means of estimating this parameter. Preliminary calculations show that the neural net uses fewer computations to obtain estimates as accurate as the present method, and that it is more robust. Hence we conclude that this is a technique that has merit in the control of gas turbines and should be investigated further.

REFERENCES


Figure 4: Robustness of Neural Network to Parametric Errors