Neural Networks for Engine Fault Diagnostics

Dawei W. Dong†, John J. Hopfield†, and K. P. Unnikrishnan†‡

†Computation and Neural Systems
California Institute of Technology
Mail Code 139-74
Pasadena, CA 91125

‡General Motors R&D Center, 480-106-285
Warren, MI 48090-9055

Neural Networks for Signal Processing VII Principe J, Giles L, Morgan N, Wilson E (Eds) (IEEE, Piscataway, NJ) page 636–644, 1997

Neural Networks for Engine Fault Diagnostics

Dawei W. Dong*†, John J. Hopfield†, and K. P. Unnikrishnan†‡

†Computation and Neural Systems
California Institute of Technology
Mail Code 139-74
Pasadena, CA 91125

‡General Motors R&D Center, 480-106-285
Warren, MI 48090-9055

Abstract

A dynamic neural network is developed to detect soft failures of sensors and actuators in automobile engines. The network, currently implemented off-line in software, can process multi-dimensional input data in real time. The network is trained to predict one of the variables using others. It learns to use redundant information in the variables such as higher order statistics and temporal relations. The difference between the prediction and the measurement is used to distinguish a normal engine from a faulty one. Using the network, we are able to detect errors in the manifold air pressure sensor (V_s) and the exhaust gas recirculation valve (V_a) with a high degree of accuracy.

1 Introduction

The basic behavior of an automotive engine is well known (Dobner 1983, Cook and Powell 1988). In the intake manifold of an automotive engine, shown schematically in Figure 1, the mass air flow rate (V_i), exhaust gas re-circulation valve position (V_a), engine speed (V_o), and manifold absolute pressure (V_s) are related by a first order dynamics:

$$\mathrm{dV_s/dt = F(V_i, V_o, V_a, V_s).}$$

In many automobiles, sensors directly measure the variables V_s, V_i, and V_o, and the actuator command V_a is also monitored. However, the above equation indicates that there is a redundancy between these variables. The

*To whom correspondence should be addressed. Phone: 818-395-2805. Fax: 818-792-7402. Email: dawei@hope.caltech.edu

consistency of the time-history of the four variables can be used to check for faults in the three sensors and in the actuator. Thus for example by monitoring the variable V_s, we should be able to reliably detect errors in variables such as V_a.[†] We present a neural network model that can capture the above dynamics of a six-cylinder engine on a production vehicle. Even though the neural network presented here is for a specific engine diagnostic problem, the approach is quite general and can be easily used for other applications as well.

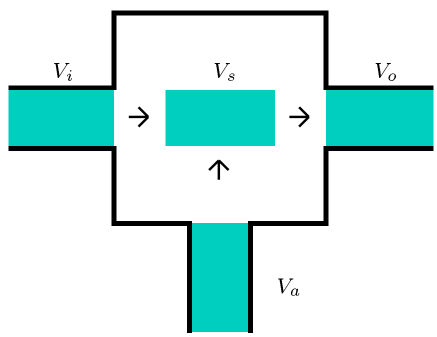

Figure 1: Engine flow diagram. There are two in-flows V_i and V_a and one out-flow V_o. Because the conservation of mass, the change of V_s, which is proportional to the total change of mass in the manifold, is proportional to the net mass flow, which is a function of V_i, V_a, V_o, and V_s.

2 Network

A two layer feedback neural network is developed to predict one variable (V_s) using three others. The architecture of the network is illustrated in Figure 2. The network has 3 feed forward inputs (V_i, V_a, and V_o), 16 first (hidden) layer neurons, and 1 second (output) layer neuron[‡]. The predicted V_s is fed back as the fourth input. The facts that (i) the first layer uses time-delayed output variables and (ii) the input-output relationship of each neuron is sigmoidal, allows the network to capture the knowledge that the physical system is characterized by a first order non-linear dynamics.

[†] There is an extensive body of literature on fault diagnosis. The inherent relationships and redundancies of measured variables of dynamic processes are often used to detect faults (e.g. Isermann 1993).

[‡] Different number of hidden neurons has been tried. 16 gives a good level of performance for this task

The data for training and testing the network was collected by a laptop computer during normal city and highway driving using an experimental hardware setup within the vehicle. The data was then loaded to a SUN workstation for training and testing. We focus on faults in the sensor V_s and the actuator V_a. The latter was chosen for its difficulty in detection. The faults were introduced by a hardware fault generator and simulates an 80% V_s fault or an 80% V_a fault. (80% V_s fault means that the sensor V_s reading is 80% of the real value. 80% V_a fault means that in the local actuator control loop for V_a, the sensor output is 80% of the actual value. This will cause the actuator V_a to open more for a given V_a command, thereby increasing the V_s by roughly 5%.)

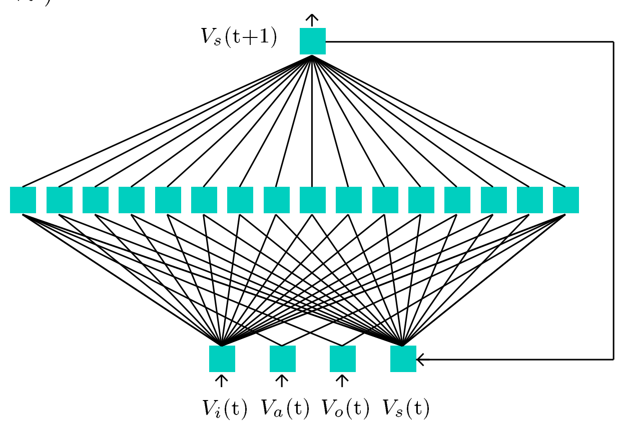

Figure 2: Network architecture. This feedback network has two layers of neurons, there are total of 80 connections (five for each hidden neuron) and 17 thresholds (one for each neuron). Those 97 parameters of the network are trained by back error propagation (BEP).

The connections of the network are trained by back error propagation (BEP) with a momentum term on a training data set. To learn the dynamic correctly, the training data are presented in the following fashion:

1) find a random starting point in a long time sequence of data, set the initial value of the feedback input to the measured V_s;
2) run the input through the network to get an output V_s, calculate the output error (the square difference of the predicted V_s and the real one);
3) set the feed forward input to the next data point and the feedback input to the predicted V_s;
4) repeat steps (2) and (3) for 100 steps to collect the error signal;

5) repeat step (1), (2), (3) and (4) for 4 steps to further collect the error signal;
6) update the connection according the BEP learning rule;
7) repeat (1) through (6) until the error does not reduce any more or until the limit of computation are reached.

The performance of the network is tested on a separate validating data set. In all the plots, except the one mentioned, the validating data set is used.

3 Network Performance

The purpose of training the close-loop (feedback) network is to facilitate the identification of a normal vehicle from a faulty one, and thus diagnose a fault. The variance (root of mean square) of the distribution of the difference between the predicted and the measured V_s is used as the quantitative measure of the network's identification power.

3.1 System Identification

The trained neural network predicts V_s value very well. This is shown in Figure 3 (left) for a segment of the normal data set, sampled every 25 ms for 400 seconds, under normal city driving conditions. The predicted V_s values (dashed line) and the measured V_s values are very close to each other.

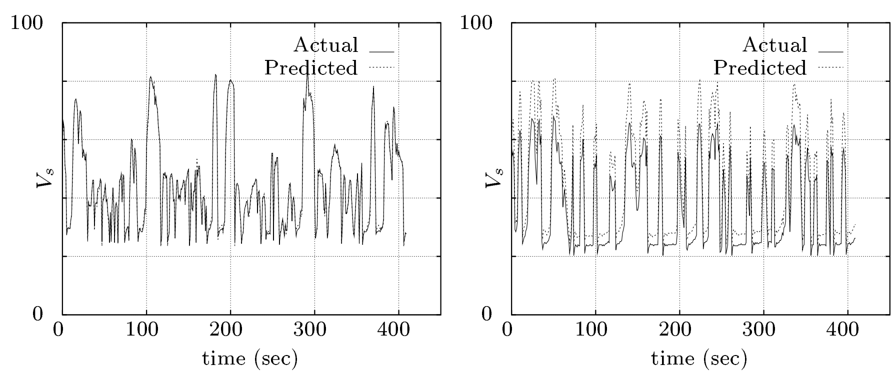

Figure 3: V_s prediction for normal vehicle (left) and faulty vehicle (right).

With a network of this accuracy, it is easy to detect faults in sensor V_s. Figure 3 (right) shows a segment of the data set which was collected with faulty V_s sensor (the reading is 80% of the true value). It is the same vehicle but with an altered V_s sensor. We can see that the prediction is about 20% above the measured value, i.e., the network predicts a V_s value 20% greater than the actual reading, given other variables.

But if a fault only causes small changes in V_s, it is not so easy to see the difference from a plot like Figure 3. For the data set which was collected with the faulty V_a actuator (the V_a actuator opens 20% more than the V_a command), the changes in V_s is only about 5%. The difference between the predicted and measured V_s, i.e., the residual, gives a quantitative measure. The distribution of the residual is quite different for a normal vehicle and a faulty one.

3.2 Diagnostic Variable

Figure 4 (left) shows the V_s residual values for the same segment of the data set as in Figure 3 (left) for a normal vehicle. The residuals are well within 1 with the mean close to 0. Obviously there are many ways to characterize the residual, e.g., binning it for different V_s values and/or V_a values. Even the residuals at different times could give information on whether a vehicle is normal or not.

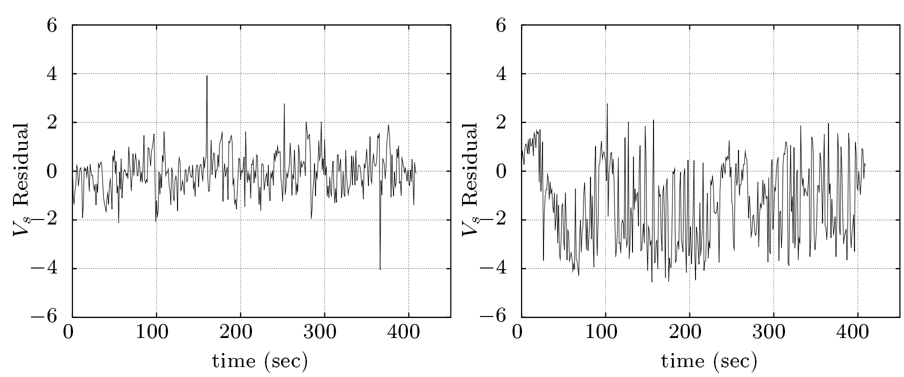

Figure 4: V_s residual for normal vehicle (left) and a vehicle with V_a fault (right).

The residual variance is the most natural one to characterize the spread of the residual distribution. For the current application, this is sufficient to separate a normal vehicle from a faulty one. For the the segment of the normal data set shown in Figure 3 (left) the running average of the residual variance is shown in Figure 5 (left, the curve in the middle). It is clear that the residual variance converges to 0.8 within about 200 sec. This translates to roughly 1.8% of the measured V_s. We can see from Figure 5 (right) that the V_s residual variance of a vehicle with 80% V_a fault is much larger than this, so it is possible to detect the V_a fault with average V_s residual variance.

3.3 Discrimination Power

Since the V_s fault is easy to detect, only the performance of the network for detecting 80% V_a fault is presented in the following. Figure 4 (right) shows the V_s residual values for an 80% V_a fault vehicle. The V_s residuals have much larger variances, in contrast to those in Figure 4 (left) for a normal vehicle.

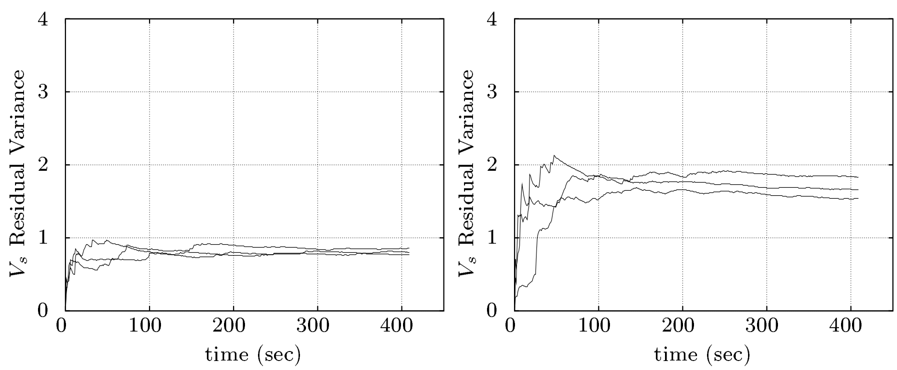

Figure 5: V_s residual variance for normal vehicle (left) and a vehicle with V_a fault (right). Three segments are plotted for each of the normal data set and the faulty data set. The variances for a faulty vehicle are larger than the normal ones by more than a factor of two.

Figure 5 shows the running average of V_s residual variances for three segments of the normal data set (left) and three segments of the V_a fault data set (right). The V_s residual variances for the faulty data set are around 1.8 , more than two times larger than 0.8 variance for the normal data set. Again, the variances approach their asymptotic values within about 200 secs. The small difference from segment to segment reflects the random driving pattern during the data collection (there was no set driving schedule).

4 Network Generalization

The most serious concern for any data dependent model (neural network and math based models alike) is how well the model generalizes. This is investigated in two ways: variations from the training data to validating data, variations for different drivers.

4.1 Training and Validating

Figure 6 (left) shows the V_s residual variances on three segments of the training data set. These can be compared with the three segments of the validating data set in Figure 5 (left). Both the training and validating data were collected for the same driver A in this case. All the running averages of V_s residual variances approach 0.7 to 0.9 after 200 seconds. There is no significant difference in the variance for training and validating data.

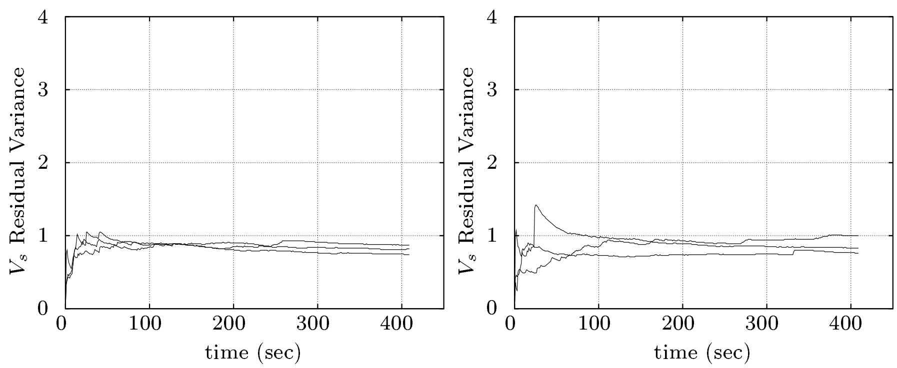

Figure 6: V_s residual variance for training (left) and for different drivers (right). On the right, the lower two curves are for driver B and the upper one is for driver C.

4.2 Different Drivers

Figure 6 (right) shows the V_s residual variances of three segments of normal data from two other drivers (B and C) which can be compared with the three segments of the validating data set as before (Figure 5, left). The network was trained for driver A. Thus the data for different drivers are not part of the training data set. The running averages of V_s residual variances for drivers B and C are only slightly higher, ranging from 0.7 to 1.1.

The small difference in performance for training, validating, and drivers is well below the level of V_s change caused by the V_a fault. Thus the network can still give a reliable signal to detect the fault.

5 Discussion

Based on the intrinsic dynamic of the intake manifold of an automotive engine (shown in Figure 1), We choose a feedback network instead of a feedforward one. To test what a feedforward network can do, we also trained a network without the V_s feedback, i.e., a standard two layer feedforward network.

Figure 7 shows the performance of the trained feedforward network. The running average of V_s residual variances for three segments of the normal data set (left) and three segments of the V_a fault data set (right) are shown in this figure. The prediction accuracy of the feedforward network is much lower than the feedback network (Figure 5). The variances for the normal data set are two times larger than the feedback network and are very close to the variances for the V_a fault data set.

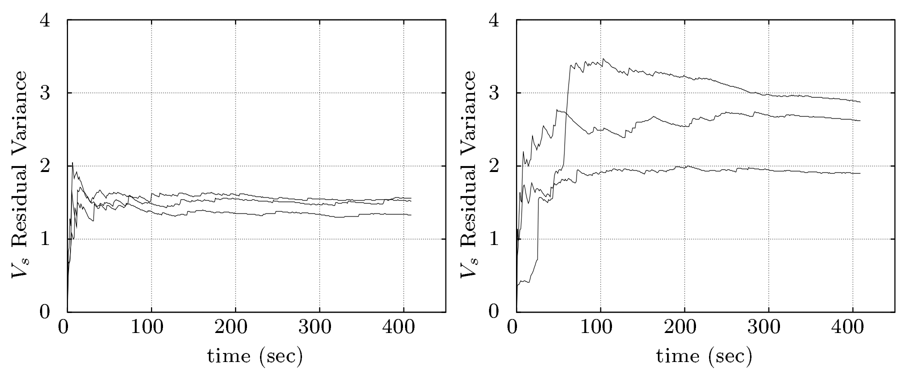

Figure 7: Performance of a feedforward network. As in Figure 5, V_s residual variance for normal vehicle is shown on the left and a vehicle with V_a fault on the right. Three segments are plotted for each of the normal data set and the faulty data set. Different from Figure 5, the variances for normal and faulty vehicles are not very far apart.

Another alternative is to train a feedforward network with true V_s, not the feedback from the output. With the same training schedule, the network learned mostly to follow the V_s input. Thus the performance is even worse than without the V_s input — in term of discriminating faulty and normal vehicles.

We have also trained feedforward networks with multiple time-delayed inputs of V_i, V_a, and V_o. They have similar level of performance as the feedforward network in Figure 7, which is much worse than the feedback one. Thus the feedback element of the network is truly important.

6 Summary

This research demonstrates the usefulness of applying neural network technology for engine modeling and diagnostics. Using a well accepted statistical measure — the variance — the two layer network with feedback achieved a accurate manifold air pressure (V_s) prediction with 1.8% variance which enables the detection of 4.5% V_s variance caused by exhaust gas recirculation valve (V_a) faults of the same vehicle.

We should point out that it is not necessary to collect the data in continuous 400 sec windows. For the current method to work, one only needs to collect small pieces of data say, 2 or 3 seconds long, and collect many pieces to accumulate enough statistics. On the other hand, collecting and processing data continuously every 25 ms itself is not very demanding. The computational needs for processing data after the network has been trained is only about 4000 multiplication per second.

Acknowledgements

We wish to thank Professor Rodney M. Goodman from Caltech for his encouragement. This project is funded by General Motors, Delco Electronics, and the NSF Engineering Research Center in Neuromorphic Systems at California Institute of Technology.

References

[1] J. A. Cook and B. K. Powell, Modeling of an internal combustion engine for control analysis. IEEE Control Systems Magazine. pp 20-26 (1988).

[2] D. J. Dobner, Dynamic engine models for control development — part I: non-linear and linear model formulation. Int. J. of Vehicle Design. Special Publication SP4 pp 54-74 (1983).

[3] R. Isermann, Fault diagnosis of machines via parameter estimation and knowledge processing — tutorial paper. Automatica. Vol 29(4) pp 815-835 (1993).