DATA ACQUISITION AND ANALYSIS FOR AUTOMOTIVE AIR BAG TESTING

D. Simon & D. Dragotoniu *

TRW Test Lab, 4051 N. Higley Rd., Mesa, AZ 85215. Email: d.simon@ieee.org

Abstract. Testing is an important part in all production processes, particularly in processes such as automotive air bag manufacturing. At TRW, both production air bags and new engineering designs are tested. Because of the high volume of tests and the high volume of data which are collected during tests, automated data collection and analysis is a highly desirable feature of the data acquisition software (DAS). This paper discusses TRW's DAS and how automatic data analysis was incorporated in the DAS. In particular, a neural network is used to classify the results of the test and hence provide lab personnel with a helping hand in their disposition of the test data.

Key Words. Automotive Air Bags, Data Acquisition, Neural Networks.

1. INTRODUCTION

An air bag is comprised of three main components:

- The inflator;
- The bag itself (usually made of fabric); and
- Associated hardware.

The inflator, which is the most critical component of the air bag, is currently based on pyrotechnic technology and is comprised of four main subcomponents:

- Solid propellant;
- An initiator;
- A filter; and
- Structural hardware.

The activation of the air bag is begun at the start of an accident by applying an electrical stimulus to the initiator. This results in ignition of the propellant, which rapidly generates a high volume of gas (primarily nitrogen). The gas inflates the bag so as to provide a supplemental restraint for occupants of the vehicle.

About 90% of the air bag tests which are performed are called *inflator tests* and are geared toward determining or verifying the performance of the inflator. An inflator test is a test in which the air bag inflator is fired after being attached to an opening at the top of a tank. Various data are collected during the test, including the firing current, the firing voltage, two channels of pressure in the tank, pressure in the combustor (inside the inflator), and temperature in the tank. This paper discusses the acquisition and analysis of inflator test data.

The inflator test needs to be properly set up, and data acquisition needs to be reliable. If the test is not properly set up, then the test may be wasted. For example, one customer may require a particular amperage for the firing pulse. If the wrong amperage is used, the test results will not be acceptable to the customer. The data acquisition needs to be reliable, because a loss of test data may mean that the test was wasted. Loss of test data could result in missing a scheduled shipment (if the test was a production test). A loss of test data could also result in the waste of a carefully manufactured prototype unit. So the loss of test data could be very expensive. It should be a high priority of any data acquisition system to be as robust and reliable as possible.

Automated analysis of the test data minimizes the need for tedious, manual inspection of test results. Each test results in a graph and a one-page report summarizing the results of the test. The manual review of the graph and report for such a large number of tests would require tedious work from trained personnel. Automating the analysis frees up lab personnel to perform tasks which are better suited to their training, although final decisions are always made by qualified test personnel rather than by automated software systems. Several data analysis functions have been automated at TRW's test lab. These include the comparison of the tank pressure data between the two channels, checking that the tank pressure is within a certain envelope, and heuristically checking the shape of the pressure curve to ensure that the test instrumentation is functioning properly.

2. DATA ACQUISITION

Until recently, data acquisition was performed on PCs in a Windows 3.0 environment. The data acquisiton software (DAS) was written in the C programming language several years ago. As it went through many modifications it became complex and extremely difficult to maintain. Any new feature which was desired (such as a new type of filtering of pressure data) required many months and much expense. In addition, the DAS did not have any automated test setup or data analysis functions. As such, setup errors were possible. Problems with the test instrumentation could be undetected until several tests had been wasted.

The first decision to be made was whether to use canned DAS, or write our own (House, 1995). It was fairly clear that we would need to write our own due to the many special capabilities and functions which we would require. The general requirements for the new DAS were as follows:

1. Provide a user-friendly interface;
2. Smoothly interface with the Laboratory Information Management System (LIMS);
3. Easily maintain and update the software;
4. Allow for data acquisition at the rate of up to 1,000,000 samples / second;
5. Allow testing even if the computer network is down;
6. Check that the test has been set up correctly before the unit is fired;
7. Automatically analyze test data in order to provide a helping hand to test lab personnel.

Requirement 1 is important because the inflator tests are conducted by test operators who are not, in general, computer experts. The DAS needs to be easy to use for someone with minimal computer experience. The test lab cannot afford to have test operators waste time trying to figure out how to use the software. So Requirement 1 dictated that the DAS be written in a language which provides a Graphical User Interface (GUI).

Requirements 2 and 3 above dictated that the new DAS be written in Microsoft's Visual Basic (VB) programming language. The lab's LIMS is written in VB, so the use of VB-based DAS makes the sharing of routines easy. It also makes the sharing of data easier, since both LIMS and DAS have straightforward interfaces with Microsoft Access databases. (VB incorporates the same database engine that powers Microsoft Access.) VB is a relatively powerful and well-supported language which makes Windows programming quite easy.

Requirement 4 is given because current and voltage may be requested at up to 200 KHz each, the three pressure channels (two tank channels and one combustor channel) may be requested at up to 40 KHz each, and the temperature may be requested at up to 20 KHz. This results in a total throughput of up to 540 KHz. So a throughput of 1 MHz satisfies the requirement with plenty of cushion. The best solution for this requirement was a purchased data acquisition board. Fortunately, data is collected for only 120 msec, so even though the throughput is high, the total amount of data is manageable.

Requirement 5 is important because testing needs to continue even if the computer network is not functioning. The test lab cannot stop testing just because there is a problem with the network server. If the network is up, then test instructions are retrieved from the network. The test instructions include such things as tank size, firing current duration, whether or not the inflator should be weighed before and after testing, and whether or not the inflator should undergo environmental conditioning before testing (e.g., vibration, drop, altitude). The test instructions are stored on the network by one of two mechanisms. For engineering tests, the test instructions are stored by engineers who request a certain test. For production tests, the test instructions are stored as "template files" based on requirements from the customer. If the network is down, then the DAS uses a local copy of the test information to conduct testing. A PC in the test lab runs an archive program and periodically downloads test instructions from the network. That way, if the network is down, there is a PC available which has the latest test instructions available. So the operator can transfer the test instructions from the Archive PC to his PC via floppy disk. The test results are uploaded to the network immediately after testing. Again, if the network is down, the test results are maintained on the local PC and the network status is checked periodically. As soon as the network comes back up, the test results are uploaded.

Requirement 6 was met by retrieving test information from the LIMS database, and automatically configuring the DAS based on the test information, or else interactively verifying with the test operator that the test is set up correctly. For example, the LIMS database specifies which data channels should be recorded, so the DAS is automatically configured based on that information. The LIMS database also specifies the temperature of the inflator which is to be tested, so the DAS interactively verifies with the test operator that the inflator has been conditioned at the required temperature. A partial list of test setup checks is as follows:

- Inflator temperature;
- Channels which should be recorded;
- Whether or not a chemical sample should be taken after the unit is fired;

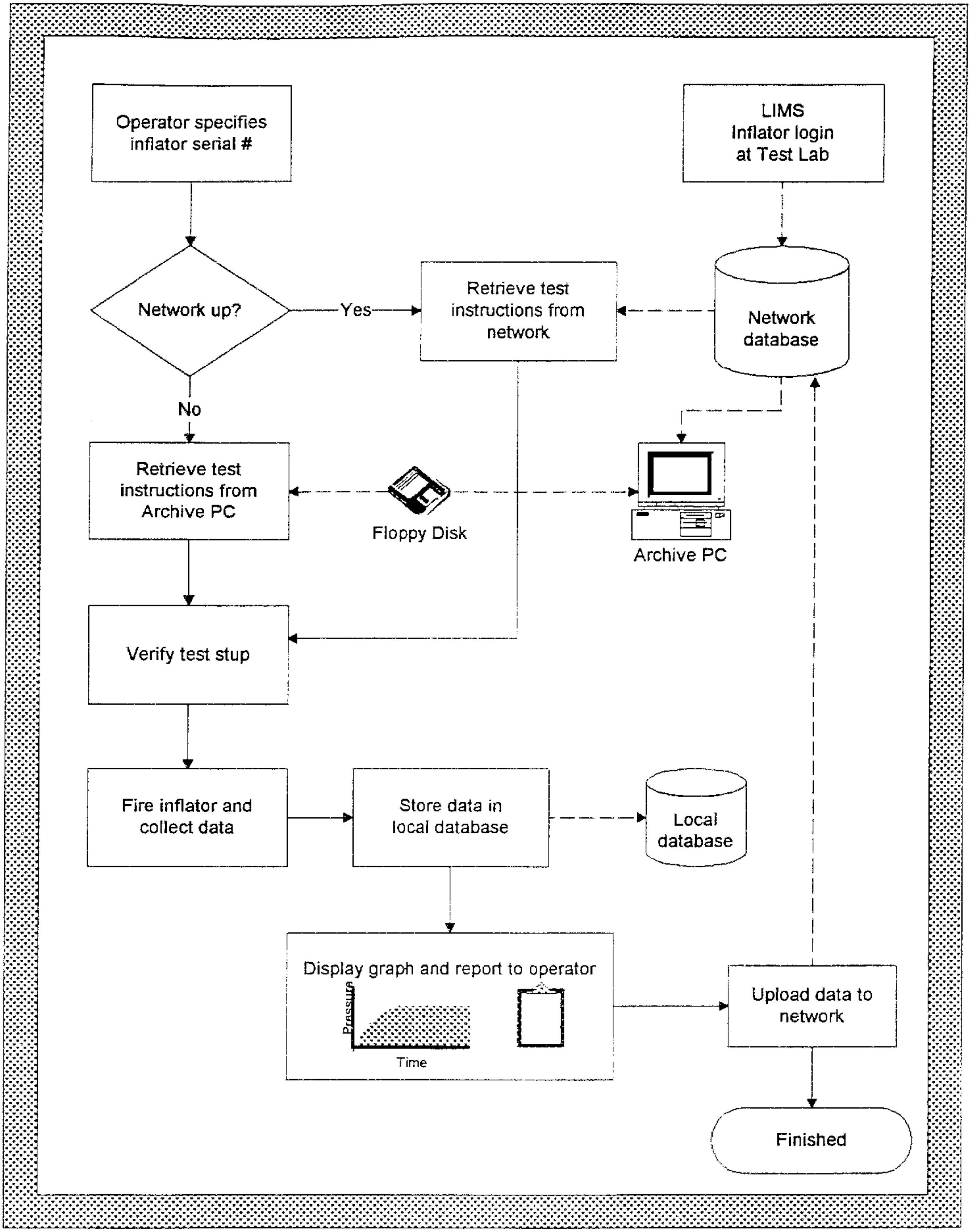

Fig. 1 Data Acquisition Process

- Whether or not the inflator has been "logged in" at the lab;
- Whether or not the inflator should be weighed before and after it is fired;
- Whether or not a video or photograph of the test should be taken;
- What the firing pulse should be (amps and time duration).

The overall flow of the new data acquisition process is depicted in Figure 1. Requirement 7 was satisfied by incorporating data analysis algorithms in the DAS, and is discussed in detail in the following section.

3. DATA ANALYSIS

There are three main data analysis functions in the new DAS. These include pressure curve envelope checking, channel comparison, and neural pressure curve classification. Future additions to the DAS analysis capability will include pressure curve slope inspection and test trend analysis. If any of the data analysis algorithms indicate an issue, then the test operator is notified and the appropriate information is attached to the test record in the LIMS database.

Customers supply pressure curve envelopes which bound the expected tank pressure curves. If the pressure curves fall outside of these envelopes, then a problem with the instrumentation may be indicated. Once the DAS collects the pressure data, it is an easy matter to verify that the pressure curves are within the envelopes.

Tank pressure is collected on two channels to ensure the integrity of the data. Theoretically, the two pressure curves should lie virtually on top of each other. In practice, the pressure curves differ at times due to slight differences in the instrumentation or asymmetry in the inflator. If the difference between the two pressures exceeds predefined tolerances, then an issue with either the inflator or the instrumentation may exist and appropriate action needs to be taken.

The shape of the pressure curves is analyzed by a backpropagation-trained neural network (Caudill, 1990). The shape of the curve should be fairly smooth. Any "wiggles" or "bumps" in the curve may indicate faulty instrumentation or an inflator which is out of spec. The problem is that quantification of how much "wiggle" or "bump" is necessary to arouse suspicion is largely an intuitive judgment by trained lab supervisors. So the expertise of the lab supervisors was used to train a neural network to perform the classification for the supervisor. If the neural network classifies the pressure curve as "need to inspect," then the test operator and the supervisor are notified electronically in real time. The supervisor can then take appropriate action, e.g., inspecting the test instrumentation.

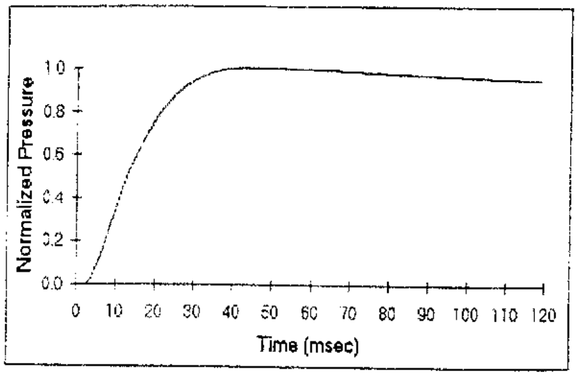

Fig. 2. Typical Pressure Curve

4. NEURAL PRESSURE CURVE CLASSIFICATION

A typical tank pressure curve for an inflator test is given in Figure 2. The curve has been filtered with a moving average filter, and looks quite "smooth" upon visual inspection. By way of contrast, Figure 3 shows an unusual-looking pressure curve. The undulations of this curve may be due to an issue with the pressure transducer, which would indicate that the test data is of questionable value, and the pressure transducer needs to be replaced. Or the undulations may be due to an assymetric inflator, causing much more gas to be expelled from one side of the inflator than from the other side, which may indicate an issue with the product or the process. The sooner these unusual-looking pressure curves are flagged, the sooner the underlying issues can be rectified. The difficulty with catching these unusual-looking pressure curves is that recognition of anamolies in the shape of the curve is largely an intuitive judgment. The judgment can be made easily by trained and experienced lab personnel. But the experienced lab personnel do not have time to inspect every pressure curve, and the test operators do not have the experience to recognize anamolous curves.

Automating the pressure curve classification thus seems to be a issue which could be solved by some sort of "artifical intelligence." We chose to use neural networks due to the difficulty of linguistically expressing the details of the problem. The primary challenge was not in choosing a network architecture, but rather in choosing how to preprocess the pressure curve so as to obtain the inputs to the network. It has been said that if you don't preprocess the data correctly, no network architecture will give you good results; but if you

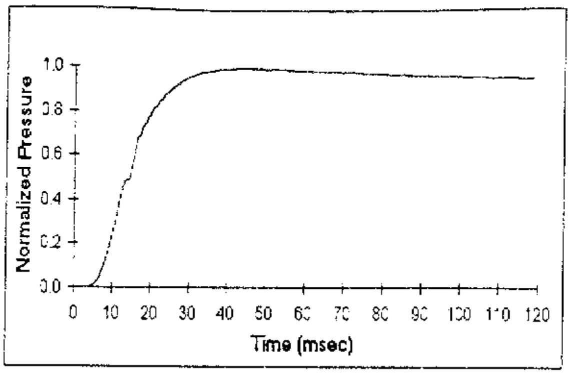

Fig. 3. Unusual Pressure Curve

do preprocess the data correctly, any network architecture will give you good results. While this statement is obviously hyperbole, it does convey the importance of preprocessing.

There are 72 inputs to the neural network, 10 hidden neurons, and two outputs which indicate how to classify the pressure curve. It was determined after much trial and error that the second derivative of the pressure curve was critical for pressure curve classification. Since the pressure curve is actually a discrete set of points obtained from a digital computer, we needed to compute the second derivative using a numerical approximation. The standard approximation was used:

$$\begin{aligned} P''(t_1) &\approx \tilde{P}''(t_1) \\ &= [P(t_2) - 2P(t_1) + P(t_0)]/h^2 \end{aligned} \tag{1}$$

where $t_2 - t_1 = t_1 - t_0 = h$. A bound of the error between this approximation and the actual value of the second derivative is given as (Atkinson, 1989)

$$|\tilde{P}''(t_1) - P''(t_1)| \leq \frac{h^2}{12}|P^{(4)}(\zeta)| + \frac{4E}{h^2} \tag{2}$$

where $t_0 \leq \zeta \leq t_2$ and E is some unknown constant. It can be seen from this equation that the optimal value of h depends on the relative magnitudes of $P^{(4)}(\cdot)$ and E. We chose to try three different values of h (say h_1, h_2 and h_3) in the approximation, and let the neural network decide which value to use for pressure curve classification. That is, we fed approximations of $P''(\cdot)$ into the neural network using three values of h, and the training algorithm for the network ended up giving greater weight to the approximation which had the least error. The method of determining the network inputs can be summarized as follows.

Preprocessing Algorithm:

For $i = (1, 2, 3)$ do the following:

1. Find the maximum second derivative approximation using the stepsize h_i. Denote this estimate as $\tilde{P}_i''(t_{max})$
2. Find the inflection points on either side of $\tilde{P}_i''(t_{max})$ (i.e., find the zero-crossing points of $P_i^{(4)}(t)$, where $P_i^{(4)}(t)$ is a numerical estimate of the fourth derivative of $P(t)$ using the stepsize h_i). Denote these inflection points as t_{i0} and t_{i1}. Do not consider any point of the pressure curve where $t \in [t_{i0}, t_{i1}]$ on subsequent passes through this loop.
3. Use t_{max}, $\tilde{P}_1''(t_{max})$, $\tilde{P}_2''(t_{max})$, and $\tilde{P}_3''(t_{max})$ as inputs to the neural network.

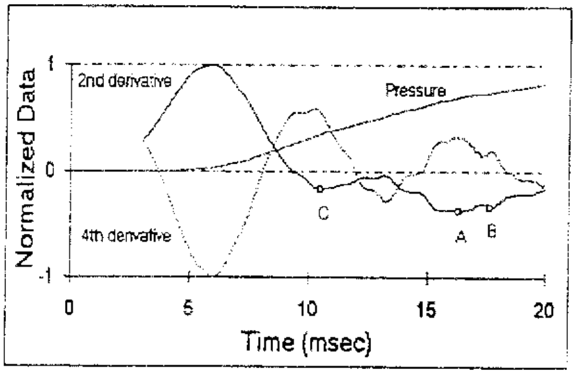

Fig. 4. Second and Fourth Derivatives of a Typical Pressure Curve

Now execute the above loop three more times, except find the three *smallest* second derivatives rather than the three *largest* second derivatives. This process results in a total of 72 inputs to the network: four inputs from each pass through the loop, looping for three extreme maxima and three extreme minima, and three stepsizes h_i for each maximum and minimum ($4 \times 3 \times 2 \times 3 = 72$).

As an example, Figure 4 shows a pressure curve and its second and fourth derivatives. Consider the search for the relative minima of the second derivative. The first input to the network is the second derivative relative minimum at about $t = 16$ msec (Point A). The second most negative second derivative relative minimum occurs at about $t = 18$ msec (Point B). But Point B is *not* used as an input because it occurs between the same two inflection points as Point A. That is, Points A and B occur between the same two zero-crossings of the fourth derivative. Instead, Point C is used as the next input to the neural network since it occurs between different inflection points than Point A.

The training process was an interactive exercise, with the neural network parameters changed by the neural network engineer based on the training progress. Some of the parameters which were changed during training and which made a noticeable impact on training progress included the learning rate, the momentum factor, and the er-

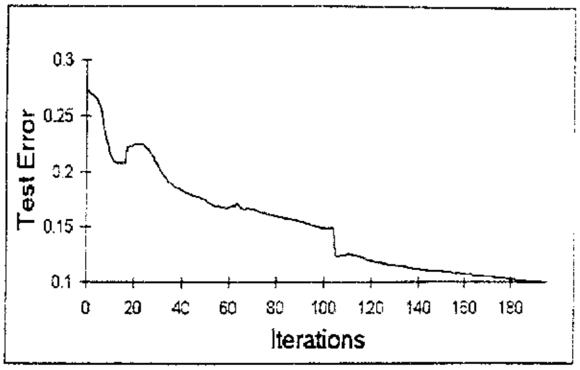

Fig. 5. Neural Network Training Progress

ror tolerance. The error tolerance is a threshold below which neural network responses are considered perfect (Caudill, 1991). For instance, if the error tolerance is ϵ, then any neural network response which has an error less than ϵ is considered a perfect response, so the error is set to zero. This has the effect of preventing weight changes due to small errors, thus precluding unnecessary oscillation of the weights. In general, as training progressed and the error decreased, the neural network engineer decreased the learning rate, increased the momentum factor, and decreased the error tolerance. Figure 4 shows the decrease of the test error as a function of the number of backpropagation iterations. We used several hundred pressure curves for the training inputs, and several hundred curves for the test cases.

Three stepsizes h_i were used in Equation 1. The inputs to the neural network consisted of second-derivative approximations; 1/3 of the inputs used h_1 msec, 1/3 of the inputs used h_2, and 1/3 of the inputs used h_3 msec to compute the approximations. Upon inspection of the neural network weights after training, we saw that the inputs which came from the h_1 msec approximations were given consistently greater weights that the inputs which came from the h_2 msec and h_3 msec approximations. This gives us reason to believe that the h_1 stepsize gives more information about the pressure curve than the other stepsizes, and hence that the h_1 stepsize gives a better approximation for the second derivative.

Recall that (for a given stepsize h_i) the steps in the Preprocessing Algorithm are executed three times for the maximum second derivative. The three times through the algorithm correspond to searches for the largest, second largest, and third largest second derivatives of the pressure curve. Similarly, the algorithm is executed three times for the smallest, second smallest, and third smallest second derivatives. Upon inspection of the neural network weights following training, we saw that the weights from the largest and smallest second derivatives were given noticably higher weights than the second and third largest and smallest second derivatives. As expected, the neural network training gave more significance to more extreme values of the second derivative.

The relative magnitudes of the neural network weights lead us to believe that the network could be pruned quite extensively to speed up training, and possibly improve generalization (Karnin, 1990).

5. RESULTS AND CONCLUSIONS

Initial requirements definitions and coding for the air bag DAS were accomplished in six man-months, after which a working prototype was available. Another six man-months were required for testing, debugging, and refinements of the DAS. As of the present time, we are in the process of installing the new DAS in the test lab. Initial results are encouraging, and we anticipate a marked increase in productivity due to the new DAS. Fewer setup issues will occur, more instrumentation issues will be caught immediately, and valuable lab supervisor resources will not be wasted on tedious and mundane inspection of routine test results.

ACKNOWLEDGEMENT

The authors wish to acknowledge the efforts of Rick Alan, who developed the theory and the initial prototype of the neural pressure curve classifier described in Section 4. He can be reached c/o Travelogix, Inc. 9420 Research Blvd. Suite III-260, Austin TX 78759 or via email at r.alan@ieee.org.

6. REFERENCES

Atkinson, K. (1989) *An Introduction to Numerical Analysis.* John Wiley & Sons.

Caudill, M. (1991) Neural Network Training Tips and Techniques. *AI Expert* **January 1991**, 56-61.

Hecht-Nielsen, R. (1990) *Neurocomputing.* Addison-Wesley.

House, R. (1995) Data Acquisition Software for Engineers and Scientists. *IEEE Spectrum*, **32**, 23-39.

Karnin, E. (1990) A Simple Procedure for Pruning Back-Propagation Trained Neural Networks. *IEEE Transactions on Neural Networks.* **1**, 239-242.