AN ARCHITECTURE FOR REAL-TIME INTERPRETATION AND VISUALIZATION OF STRUCTURAL SENSOR DATA IN A LABORATORY ENVIRONMENT

William Doggett, Sixto Vazquez; NASA Langley Research Center, Hampton, Va.

Abstract

A visualization system is being developed out of the need to monitor, interpret, and make decisions based on the information from several thousand sensors during experimental testing to facilitate development and validation of structural health monitoring algorithms. As an added benefit the system will enable complete real-time sensor assessment of complex test specimens. Complex structural specimens are routinely tested that have hundreds or thousands of sensors. During a test, it is impossible for a single researcher to effectively monitor all the sensors and subsequently interesting phenomena occur that are not recognized until post-test analysis. The ability to detect and alert the researcher to these unexpected phenomena as the test progresses will significantly enhance the understanding and utilization of complex test articles. Utilization is increased by the ability to halt a test when the health monitoring algorithm response is not satisfactory or when an unexpected phenomenon occurs, enabling focused investigation potentially through the installation of additional sensors. Often if the test continues, structural changes make it impossible to reproduce the conditions that exhibited the phenomena. The prohibitive time and costs associated with fabrication, sensoring, and subsequent testing of additional test articles generally makes it impossible to further investigate the phenomena.

A scalable architecture is described to address the complex computational demands of structural health monitoring algorithm development and laboratory experimental test monitoring. The researcher monitors the test using a photographic quality 3D graphical model with actual sensor locations identified. In addition, researchers can quickly activate plots displaying time or load versus selected sensor response along with the expected values and predefined limits. The architecture has several key features. First, distributed dissimilar computers may be seamlessly integrated into the information flow. Second, *virtual* sensors may be defined that are complex functions of existing sensors or other virtual sensors. Virtual sensors represent a calculated value not directly measured by particular physical instrument. They can be used, for example, to represent the maximum difference in a range of sensors or the calculated buckling load based on the current strains. Third, the architecture enables autonomous response to preconceived events, where by the system can be configured to suspend or abort a test if a failure is detected in the load introduction system. Fourth, the architecture is designed to allow cooperative monitoring and control of the test progression from multiple stations both remote and local to the test system. To illustrate the architecture, a preliminary implementation is described monitoring the Stitched Composite Wing recently tested at LaRC.

Section 1, Introduction

A visualization system is being developed to monitor, interpret, and make decisions based on the information from several thousand sensors during experimental testing. Design and development of the Embedded Decision Advisor and Fault Indicator (EDAFI) grew out of several needs. First, a recognized need within the organizations performing structural testing that a trend toward validation of larger test articles required increasingly more sophisticated monitoring to prevent unexpected failures. Second, a need within our organization to test and validate integrated structural health monitoring systems in a realistic environment. It is our belief that, if we are unable to reliably demonstrate identification of structural faults in articles during laboratory tests, then it is impossible to reliably detect structural faults in the flight environment. Further, it is important to understand how the structural faults have degraded

the load carrying capacity of the structure so that proper actions may be initiated. The availability of a large number of test specimens, often with detailed load response models, makes the structural test environment a perfect environment to test and validate structural health monitoring sensors and systems.

This paper describes the foundations of EDAFI and how it can be applied to the analysis of structural faults. "Health monitoring" systems have been developed for craft in extreme environments [1,2], however these systems do not meet the unique requirements of a laboratory environment. Generally these systems concentrate on efficient design, redundancy and reliability which while important, are not critical in a laboratory environment. In the laboratory distributability, expandabilty, intuitive display, ease of development and modification are more important.

An overview of the design challenges and a description of the project objectives is given in Section 2. In the overview an important construct, the *virtual* sensor, is introduced. Section 3 details the software architecture and describes the data structure used for implementation. Section 4 describes structural tests of the stitched composite wing which provided an opportunity to verify system performance under real test conditions, record a realistic data set, assess setup procedures, and assess user interface effectiveness. The last section, Section 5, summarizes the paper and reviews important conclusions.

Section 2, Overview

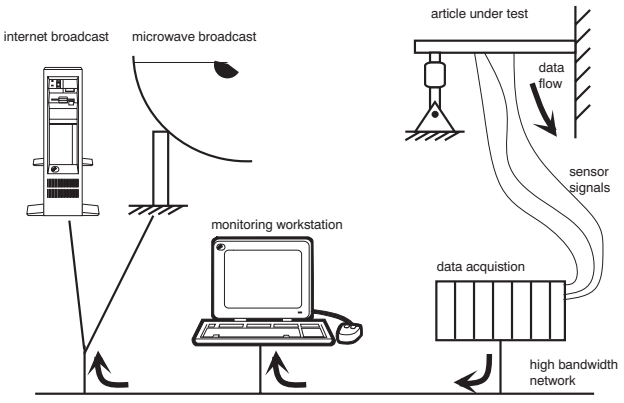

Figure 1, Hardware Overview

EDAFI interacts with a test as shown in Figure 1. In the upper right of the figure is an article under test with clockwise data flow indicated by the arrows. Raw sensor signals from the article are processed by a data acquisition system, which in turn broadcasts the information to monitoring workstations via a high bandwidth network. Although the figure shows a single data acquisition system and monitoring workstation, EDAFI's architecture is designed to support an unlimited number of data acquisition systems and monitoring workstations. In addition, as shown in the upper left of the figure, the network information may be transmitted to geographically distant locations via either internet or microwave broadcast.

The EDAFI architecture is designed to address several key goals. They are:

- *user defined regions of interest*
  individual monitoring workstations can be configured to display subsets of the available information.
- *ease of integration into the testing process*
  introduction of one or more monitoring workstations will have no impact on existing data acquisition systems.
- *scalability*
  there are not practical limits on the number of sensors, nor on the distance from sensors to monitoring workstations.
- *distributability*
  monitoring workstations can be distributed across multiple computing platforms which may be located geographically far from the test location. This is especially beneficial to those researchers monitoring fatigue tests.
- *rich development environment*
  the objective is ease of algorithm development and integration into the test data flow. Researchers can build on the tools provided or the researcher may use their own development environment.

The first two goals are addressed by the user interface which is detailed in Section 4. The remaining goals are addressed by a combination of hardware and software techniques. Key to successful implementation is the use of a dedicated high speed network implemented using a replicated

memory product called Scramnet™ from the Systran Corporation [3]. Oguchi et al evaluated the replicated memory architecture and found that performed well as it was scaled, especially in a highly distributed environment [4]. Computers attached to the Scramnet™ network view the replicated memory as a shared data structure, much like a memory mapped file. Interface cards can be purchased for most computer systems, including those based on VME, PCI, and EISA. This enables dissimilar computers to easily participate in tests. Implementers view the shared data structure as if multiple processes on a single computer were accessing the data. The fact that different computers are accessing the information is transparent to the implementer. An added benefit is that each computer can restrict access to any portion of the data structure for added security/robustness. For example, this feature can be used to prevent an errant monitoring program from corrupting the information flow.

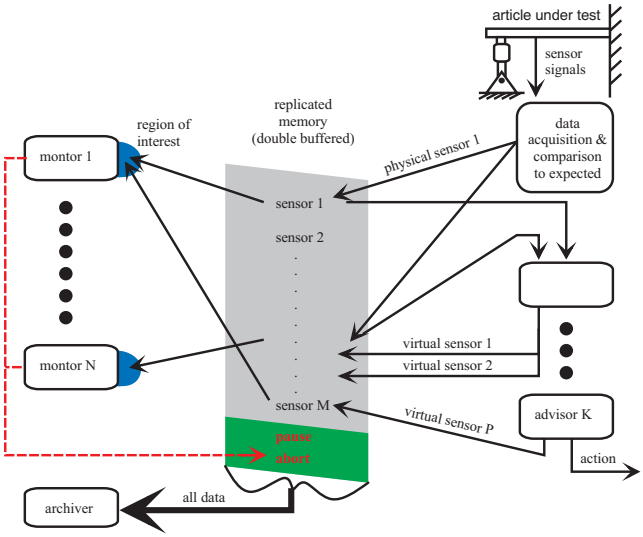

Figure 2, Overview of Software Agents

Via replicated memory the data flow, indicated by the arrows in Figure 2, occurs transparently to the implementer. In Figure 2, multiple agents, indicated by the rounded boxes, are accessing a data structure indicated by the parallelogram at the center of the figure. Four agent types are shown in the figure, they are data acquisition, advisor, archiver, and monitor.

Information flow from a sensor to a user begins in the upper right of the figure where a data acquisition agent processes physical sensor signals and loads the current values into replicated memory. Then advisors, located in the lower right of the figure, are activated to calculate the values of *virtual* sensors (discussed later). After the advisors have completed, the monitors display the selected information to the user while the archiver saves the current data.

The data acquisition agent's distinguishing characteristic, compared to an advisor, is that it processes sensors interacting with the environment. Generally, it also performs simple calculations, such as deviation from expected value. An advisor on the other hand, interprets the results produced by data acquisition agents or other advisors. Advisors can perform very complex analysis, such as implementing a fatigue crack growth prediction algorithm. Monitors, shown to the left of the figure, display information to users. They do not perform calculations; they strictly format the data into plots, histograms, etc.

Virtual sensors are a key feature of the EDAFI architecture. A virtual sensor represents a value calculated by a user-defined algorithm applied to physical sensor values or other virtual sensors. The value can represent a simple calculation such as the maximum deviation in a group, or a more complex calculation such as a crack location prediction. Advisors execute the user-defined algorithm and then update their respective sensor values. To the rest of the system, the virtual sensor's value looks just like another physical sensor value, greatly simplifying the design of the monitors. Currently, dedicated code is used to implement each virtual sensor's calculation, however an effort is underway to provide a library of predefined routines for common calculations. In addition, virtual sensors can trigger events. Currently this is supported through dedicated hardware and software accessible by the advisor. For example, an advisor can be applied to monitoring hydraulic oil temperature during a test. If the temperature exceeds a predefined limit, the fault is logged and a warning given to the test conductor. EDAFI can optionally be configured to automatically initiate a shutdown by tripping a relay.

Advisors will also be used to evaluate structural health monitoring algorithms. Visiting or local researchers simply provide a computer with an

interface card that provides immediate access to the data stream. For algorithm development, a small executable on the development machine simulates the data acquisition by reading data from a file and transferring the information into replicated memory. Thus, once they are satisfied with their algorithm's performance on prerecorded or simulated data they can then participate in a test without any additional software modifications. During the test, data flows into the replicated memory from the data acquisition process instead of from the prerecorded data file.

An archiver, shown in the lower left of the figure, is used to store the data acquired during a test. The resulting archive is important for assessing test article performance and is critical for developing, assessing, and tuning advisor performance. Further, the archive provides the researcher with a mechanism to review a test by replaying the archive through the replicated memory. To the rest of EDAFI, this fully simulates the test so that the researcher has all the capabilities that were present during the test available for post test analysis.

Notice, in Figure 2, that the agents on the left can act in parallel with those on the right, enabling considerable computational power to be used. In addition, by double buffering the replicated memory, both groups can process data concurrently. The data acquisition system and advisors operate on one buffer, while the monitors and archivers read from another, greatly simplifying synchronization. If a further increase in data throughput is needed, then the information from the data acquisition agent can also be doubled buffered.

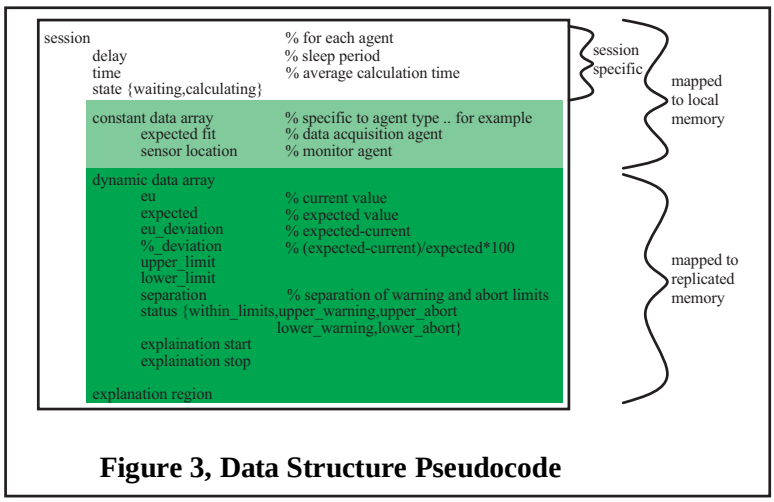

Figure 3, Data Structure Pseudocode

Section 3, Software details

Figure 3 gives pseudocode for the data structures used to support the information flow of Figure 2. A "session" is used to encapsulate data for each task that may include one or more agents of a particular type. Each session contains information specific to each unique instantiation along with a constant data section that is local to the session and a dynamic data section that overlays the replicated memory. The user's region of interest definition dictates the size of the constant data section, while the number of sensors active during the test determines the size of the dynamic data section. EDAFI's architecture is purposely flat to ease development. A startup routine is available to allocate and initialize the data structure of Figure 3 enabling data access. Protections schemes can be easily implemented to restrict data access to none or read-only on a per workstation basis.

The session specific information is primarily used to control the calculation/sleep cycle. *Time* represents the average time between activations. After the agents have completed their calculations for a particular cycle, they sleep for *delay* seconds, then wake up and begin a polling cycle waiting for the new data.

Constant data does not change during a test and is uniquely defined for each agent type. For example, monitors store the location and orientation information for the respective sensors as constant data, while the data acquisition process stores parameters that define the limits and expected curves.

Dynamic data information changes during the test. Primarily, this area is dedicated to sensor information (from both physical as well as virtual) and their associated parameters, listed in Figure 3. Section 4 will describe the mechanism for defining and calculating the parameters. At the bottom of the dynamic data section is an area of memory reserved for explanations from data acquisition or advisor agents. For example an advisor designed to count cracks may also be capable of predicting crack location. The crack location can be passed to a monitor via the explanation region.

Figure 4, Stitched Composite Wing Loaded to 95% of Ultimate Load

Section 4, EDAFI applied to Stitched Composite Wing test

EDAFI's user interface is key to effective initialization and operation. A user typically interacts with EDAFI in two modes; setup and monitoring.

A monitor agent, generally one for each workstation, provides the interface for monitoring a test or for post-test review by replaying archived data. The monitor user interface design focuses on quick, concise display of anomalous conditions. At any time a researcher can elect to plot the data associated with a senor or group of sensors. To illustrate the user interface features, a description of monitoring a small group of sensor is given. These sensors were attached to the stitched composite wing shown in Figure 4 destructively tested at NASA Langley Research Center on June 1st 2000 [5].

Monitoring interface

Figure 5 is a view of the underside of the EDAFI monitor's wing model. COTS software was used to create the 3D model from a series of digital images. The software forms a 3D mesh from points selected in the images and then textures the mess with the images to create the 3D model. The "lighter" areas in the figure are reflections from the digital camera used to capture images of the wing. The light colored plate on the right of the figure is the tip load introduction frame that is also visible at the right of Figure 4. The light color area to the left of center of Figure 5 distributes the landing load. The dark hexagons represent some of the sensors monitored on the underside of the wing. Additional sensors mounted on the top of the wing were also monitored. The hexagons change color based on either sensor magnitude or deviation from expected value. Figure 6 is a zoomed view of the center of Figure 5. The detail from the photographic images

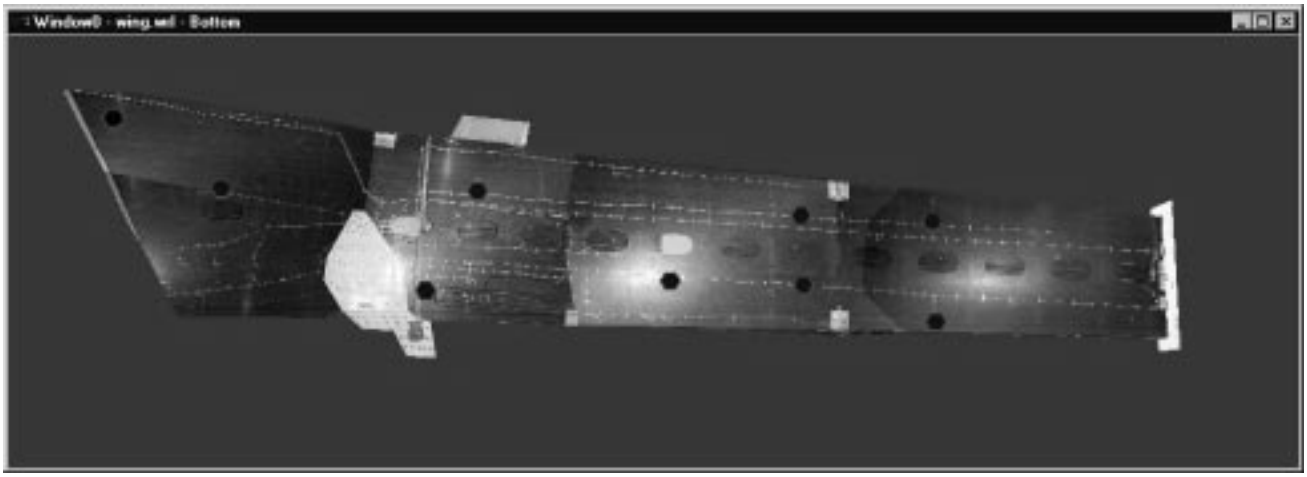

Figure 5, EDAFI View from under Stitched Composite Wing

is an important feature of the EDAFI interface, providing easily recognized visual clues of sensor locations.

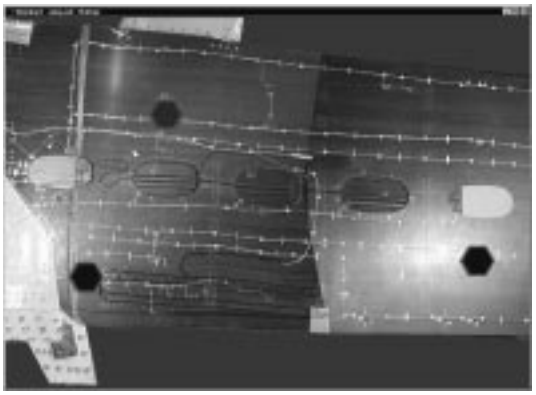

Figure 6, Close-up of Wing Underside

During a test multiple windows may be displayed to view the wing model from different perspectives. The viewing perspective may be rotated or translated by dragging the mouse while depressing the right or left button respectively. Groups of views may be saved and recalled to facilitate quick assessments of different regions during a test. A significant advantage of the graphical model is the ability to view groups of sensors physically collocated, even those located inside the wing. This makes it easy to evaluate one sensor in an area populated with other sensors to determine if an unexpected response is occurring or is a single sensor behaving erratically, due to a mechanical failure.

Figure 7, Monitor Sensor Grid

Figure 7, depicts a second display mechanism supported by the monitors called the sensor grid. This mechanism allows all the sensors present in a test to be monitored simultaneously. Two presentation styles are supported, one, shown in the figure, which displays the sensor name and it's current value and an abbreviated version that only shows the sensor names. The sensors may be grouped by the user in any order to imply relationships. In both representations, the background color behaves the same as the color in the hexagons of Figure 6, making it easy to locate sensors, or groups of sensors that are not behaving as expected.

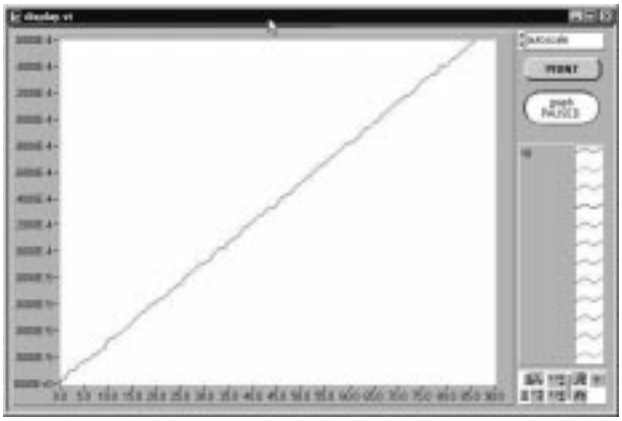

Figure 8, Time History of Selected Sensors

Clicking the mouse on a sensor in the grid of Figure 7 or on the model of Figure 5 initiates a plot similar to that shown in Figure 8. Multiple sensors may be selected and plotted simultaneously in one or more windows. They may be displayed versus time or versus another sensor value. For example, it is often useful to plot strain versus load. In addition, the expected value and associated limits can be displayed on the same graph.

The last interface available through a monitor agent is shown in Figure 9. To the left of the window is a numerical mechanism for adjusting a particular windows viewpoint. At the top left are options primarily related to graphic rendering. The "Position Sensor" option is used to manually set or adjust the location of a sensor on the 3D model. The "Warning" and "Error" boxes contain lists of sensors which have exceeded their warning and error limits respectively. They are ordered by occurrence and remain on the list as long as they exceed the respective limit. The "Pause" and "Abort" buttons (currently not active) enable a researcher to request a test pause or abort. The "Exit" button terminates the EDAFI monitor. In the upper right of the window are the pull downs used to save the current view settings. Finally, at the bottom right of the window is the message log. This log displays messages from the advisors indicating actions that have been taken, notes about sensors exceeding limits, etc. It is saved and can be used to locate times during the test where interesting phenomena occurred. Then the data can

be indexed directly to the time of interest for further analysis.

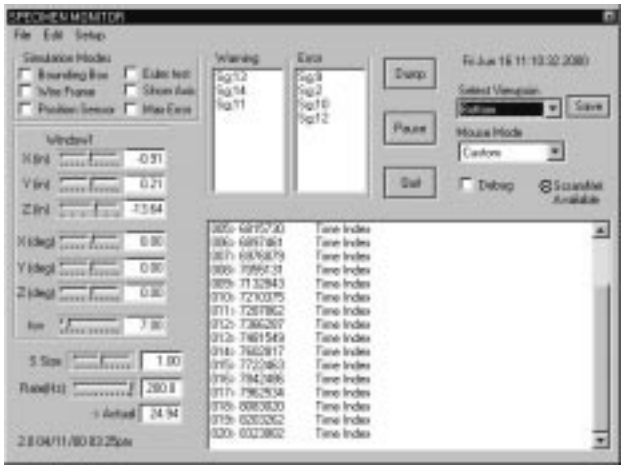

Figure 9, Monitor Control Display

Setup interface

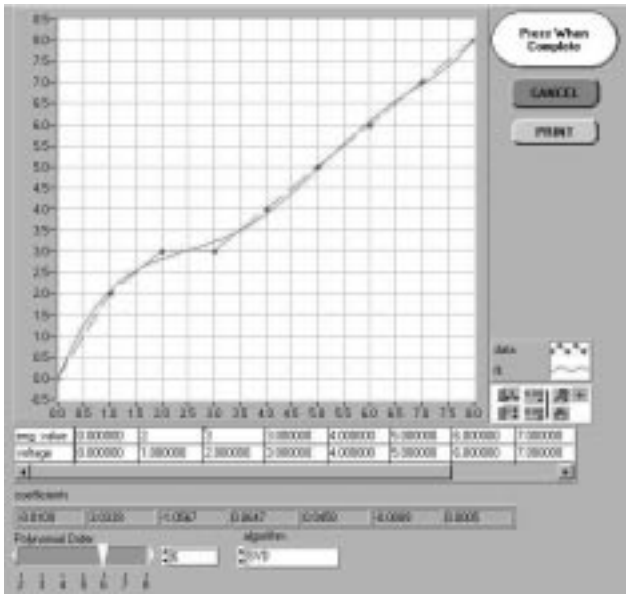

Figure 10, Expected Response

Most of the setup process is straight forward and not unique to EDAFI, for example definitions of sensor channels, voltages ranges, etc. These procedures are not reviewed here. The setup requirements described here are unique to EDAFI, including limit definition and definition of expected sensor response.

The expected sensor response is developed via the window shown in Figure 10. Once the engineering versus sensed quantity data is entered, the user selects a polynomial curve that best represents the response. The solid line indicates the "best fit" curve, while a dashed line connects the data points. The coefficients for the resulting polynomial fit are stored and used to calculate the expected response during the test.

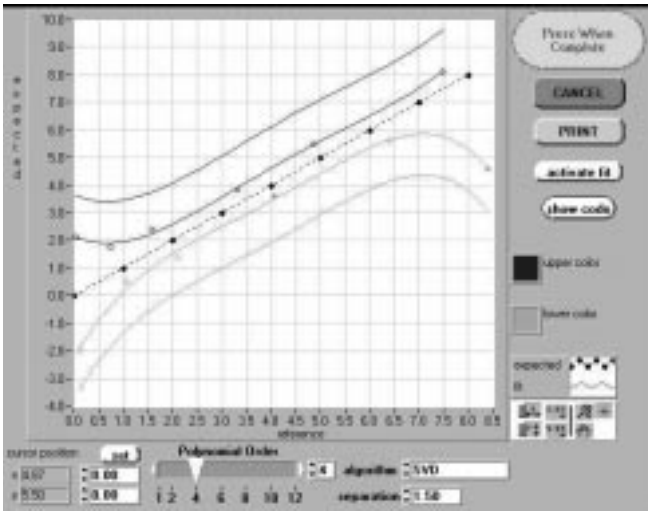

Figure 11, Limits definition

Once an expected curve is defined, the next step is to define the limits using the interface shown in Figure 11. Two pairs of traces are shown bracketing the expected trace in the center of the figure. "Separation" is used to define the distance between each pair of traces, i.e. the separation of the upper limits. The circles are used to define limit shape. The shape can be adjusted by dragging the circles, by adjusting the polynomial order or by changing the algorithm used to perform the fit. The limit traces closest to the expected trace define the warning boundary; those further away define the error boundary. The shapes of the upper and lower traces are independent, however the error trace is always a fixed distance (the separation distance) from the warning limit. As for the expected fit, the coefficients describing the upper and lower limits are stored and used to calculate their respective values during the test.

Lesson's learned

The use of actual photographic images of the article under test to texture the 3D model provided important visual clues while monitoring sensors. This was especially important for close inspection

of an area (high zoom). An improved mechanism for capturing the pictures is needed to avoid the contrast boundaries on the surface at the image boundary.

The expected curve and limit definition phases are too tedious to be applied to a large number of sensors. A mechanism to link to finite element models to automatically load the expected curve is needed. Limit curves based on templates, which are parameterized by the maximum value, are also needed so that large groups of sensors can be processed simultaneously.

Managing the simultaneous triggering of alarms as the test article begins to fail under load also needs to be addressed.

Section 5, Conclusions

A visualization system, the Embedded Decision Advisor and Fault Indicator (EDAFI), is being developed at NASA Langley Research Center to address the need to monitor extremely complex test articles and validate structural health monitoring algorithms in a realistic environment. This paper has described architectural foundations of the system and given a glimpse into how it will be applied.

EDAFI has several key features. First the ability to define virtual sensors to present the result from complex computations provides a simple interface for researchers to access existing sensor data and insert additional intelligence. Second, the use of a replicated memory architecture enables utilization of significant, potentially distributed, computational power during a test. Further the replicated memory architecture makes it possible for geographically distant researchers to monitor and participate in the tests. Third the EDAFI architecture is scalable. As the number of sensors or monitors increases the workload can be partitioned and attacked in parallel with additional data acquisition and/or computational systems. Fourth, by using a flat architecture, the system maintains the flexibility critical to laboratory development and testing. The addition of a sensor is straight forward and immediately available to advising tasks.

EDAFI is expected to enhance the utilization of test articles by enabling assessment of all sensor responses in real-time during tests. When anomalies are detected, testing can be suspended before changes in the structure occur that would destroy the unexpected structural behavior. In addition EDAFI's architecture supports independent algorithmic development while simultaneously enabling test participation with few or potentially no software changes. This feature is critical to timely evaluation and verification of structural health monitoring algorithms.

References

1 Muldoon, Richard C.; Gill, John and Larry D. Brock. "Integrated mechanical Diagnostic Health and Usage Monitoring System; an Open System Implementation Case Study", IEEE 9.B.4-1.

2 Harris, Michael and David Ngo. "An Open Architecture for Next Generation Space Onboard Processing", IEEE 1999, 9.B.3-1.

3 Systran Corporation. 4126 Lindon Ave. Dayton, Oh 45432-3068.

4 Oguchi, Masato; Aida, hitosi, and Tadao Saito. "A proposal for a DSM Architecture suitable for Widely Distributed Environment and its Evaluation", Proceedings of the Fourth IEEE International Symposium on High Performance Distributed Computing, 1995, pp. 32 -39

5 Phillips, Edward H. "Composite Wing Box Tested to Failure", Aviation Week and Space Technology, June 19, 2000, pg. 37.