Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Prototype Expert System Workstation for Satellite Anomaly Resolution

S. R. Turner,* P. K. Mangan,† M. D. Meloan,* and C. B. Simmons*
The Aerospace Corporation, Los Angeles, California 90009

Introduction

SPACE vehicle fault diagnosis and correction are difficult problems facing the areospace industry. The typical military space vehicle is extremely complex. Thousands of telemetry items that indicate satellite status are sent to ground stations, where human operators must reliably interpret the data and respond to anomalies. Current methods for detecting and correcting anomalies are primarily manual and, therefore, slow and prone to error. The increasing number and complexity of space vehicles and the diminishing supply of qualified experts are leading to even greater difficulties. Further, there is a growing requirement for austere staffing, less contractor support, and more rapid anomaly detection, especially in mobile stations. It is increasingly clear that current methods can no longer cope with the enormous scope of the space vehicle fault diagnosis and correction requirements.

In response to these concerns, we are developing a real-time operator workstation prototype based on expert system technology for the Defense Meteorological Satellite Program (DMSP).[1] This workstation, the Satellite Ground Expert (SAGE), combines expert systems with a graphical interface to provide the ground station operator with the information and tools needed to analyze the incoming telemetry stream, detect anomalies, and take corrective action.

SAGE addresses three important issues in ground workstation support:

1) Design and implementation of an overall system architecture that integrates real-time telemetry servers, expert system servers, and graphical interfaces. Concerns in this area include interprocess communications, shared knowledge, and reliability.

2) Design and implementation of a user interface for the wideband, real-time telemetry environment. Concerns in this area include operator overload, interface flexibility, and display design.

3) Design and implementation of expert systems to detect, diagnose, and suggest corrective actions for satellite faults. Concerns in this area include the decomposition and interaction of knowledge bases and appropriate recommendation policies.

Presented as Paper 90-0323 at the AIAA 28th Aerospace Sciences Meeting, Reno, NV, Jan. 8–11, 1990; received June 7, 1990; revision received Oct. 26, 1990; accepted for publication Oct. 27, 1990.

*Member of the Technical Staff, Expert Systems Section, System Software Engineering Department, P.O. Box 92957.
†Manager, Expert System Section, System Software Engineering Department, P.O. Box 92957. Member AIAA.

This Note is primarily concerned with the first issue. The following sections describe the individual components of SAGE, their organization, and their communication pathways.

System Architecture

The SAGE system has three principle components: a knowledge-based expert system, a telemetry server, and a graphical user interface. Each component is implemented as a separate process. These processes operate independently and communicate by message passing. The organization of these processes is shown in Fig. 1.

There are two advantages to this architecture:

1) Decomposing system components into separate processes and using a message passing scheme for interprocess communication allows SAGE to take advantage of a multiprocessing environment. Of course, in a single processor environment SAGE can be executed through timesharing. This architecture can also be expanded to accommodate multiple interfaces and expert systems.

2) Maintenance and development of each component is independent of the other components. Different classes of users can be provided with different interfaces with no impact on the rest of the system. Similarly, knowledge bases can be chosen for particular applications without requiring a massive system redesign.

Expert System

SAGE's expert system is implemented in Nexpert.[2] Nexpert provides a graphical development environment with sophisticated tools for debugging knowledge bases and for displaying rules. It uses both objects and rules to represent facts and the relationships between facts in knowledge bases. Facts include telemetry measurands and their values, derived mea-

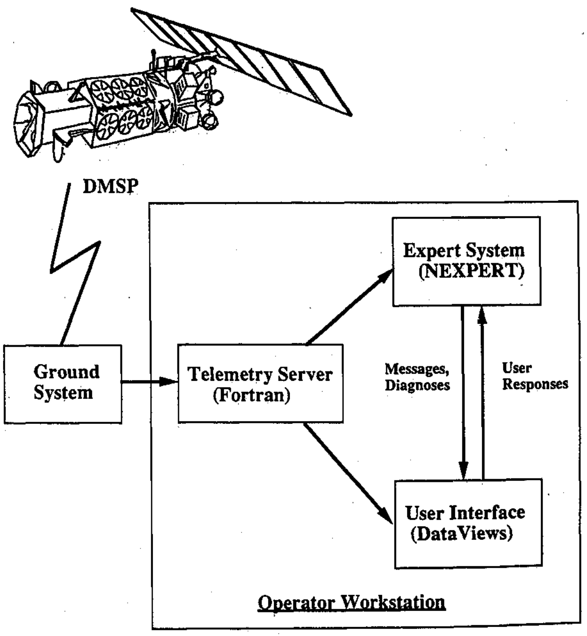

Fig. 1 Expert system workstation architecture.

surands and their values, and deduced knowledge about the state of the satellite. Relationships between facts are represented as rules such as "IF 1SCPOK > 0 THEN Computer_1_is_in_control."

SAGE's expert system consists of a supervisor knowledge base, subsystem knowledge bases, and cleanup procedure knowledge bases that implement the cleanup procedures recorded within the DMSP operations manuals.[3] The supervisor knowledge base is the top-level expert. Definitions of telemetry parameters, allowable values, and most of the information needed for ingesting telemetry are in this knowledge base.

The supervisor knowledge base is responsible for noting anomalous telemetry items. When a telemetry item is out of limits or otherwise noted as bad, the supervisor determines which subsystem knowledge base is suitable for analysis of the problem and loads that base. The particular subsystem knowlege base is then responsible for diagnosing the anomaly and loading a cleanup knowledge base. The cleanup knowledge base suggests the appropriate cleanup procedure to the operator.

Telemetry Server

SAGE's telemetry server is a Fortran program that can read telemetry from either stored telemetry files or a real-time data stream. The telemetry server reads telemetry values, converts them to a usable numeric format, filters out irrelevant data, and sends these values on to both the knowledge server and the interface server.

To reduce SAGE's overall processing load, the telemetry server monitors only the telemetry items of interest to the other processes. Knowledge of needed parameters is passed from the requesting process to the telemetry server as the need arises. For example, when a problem is detected in the Solar Array Drive (SAD), the knowledge server requests telemetry items from that subsystem.

SAGE's overall processing load is further reduced by passing telemetry items on an if-change basis. Telemetry values arrive at a rate of N per second, but typically change far more slowly. By sending telemetry items only when a change in their value occurs, the telemetry server reduces both the communication and the processing loads of the other servers.

User Interface

SAGE's user interface is implemented in DataViews.[4] DataViews provides facilities for creating pictorial displays, graphical representations of data, and interactive user environments.

The user interface displays the results of the expert system and the telemetry server in a variety of ways. Messages and recommendations from the expert system are displayed in a message window, and knowledge bases are also used to drive "status displays"—compact, readable displays that summarize the state of satellite subsystems. The expert system can also drive animated pictorial displays. This capability is used to display satellite schematics that reflect the internal state of the satellite in real time.

Telemetry items provided directly to the user interface can be displayed in a variety of formats: graphs, charts, or raw values.

Figure 2 illustrates the SAGE interface. The boxes on the left-hand side of the screen are user options. By clicking on Help, the user can receive online information about the use of the SAGE system. Subsystem status blocks, critical telemetry status charts, and user options are located on the perimeter of the screen. A working window in the center of the screen supports the following user functions: display of telemetry in alphanumeric or graphical form, display of warning messages and information from the inference process, hierarchical schematic diagrams animated in real time, browsing of diagnostic conclusions and recommendations, and browsing of cleanup procedures for resolving anomalous conditions.

Warning messages received from the knowledge server are color coded based on severity: green (less importance), yellow (moderate importance), and red (urgent). Schematic diagrams are updated in real time to reflect the current state of the onboard hardware. The schematics are structured hierarchically, and the user may browse the various levels by clicking the mouse on appropriate component icons. Anomalous components within each level are highlighted in red.

REDMN System

The primary use of SAGE is to monitor the redundancy manager (REDMN) software. To insure the health of the DMSP spacecraft, many hardware and software components have both a primary and a backup configuration. In the event of an onboard malfunction in which one of the critical components becomes nonoperational, reconfiguration of the vehicle occurs automatically under the direction of the redundancy manager (REDMN) flight software.

Unfortunately, REDMN actions are difficult to deduce from the satellite telemetry. Often the operator is not immediately aware that reconfiguration has occurred at all, for that information is embedded in a number of telemetry parameters that may not be under consideration at the time the anomaly occurs.

The function of SAGE's REDMN knowledge base is to monitor the telemetry measurands that reflect REDMN actions, detect anomalies, and communicate the anomalous conditions to the operator. The operator will be notified which component of the spacecraft has failed and the exact nature of the failure by English messages and by the highlighting of the failed components in the appropriate structural diagrams. In addition to recognition of the anomaly, procedural knowledge bases guide the user through those steps necessary to correct the anomaly if correction is possible.

Currently, the SAGE REDMN knowledge base detects and diagnoses nine anomalies that provoke action by the redundancy manager. Cleanup knowledge bases also provide recommendations for cleanup procedures for reconfiguration of the satellite from the ground. These nine anomalies include all the possible failures in the Attitude Determination and Control Subsystem (ADACS) and the Electrical Power and Distribution Subsystem (EPDS). Since there is no telemetry item that directly reports these anomalies, these knowledge bases provide the operator with information not otherwise available.

Two knowledge representation paradigms were used to design the knowledge bases for DMSP. A hierarchy of both classes and objects defines the satellite subsystems in terms of their associated telemetry parameters. Objects are telemetry measurands with their associated values from the satellite telemetry stream. These objects are organized into three major classes: 1) analog signals, 2) bivalued discrete signals, and 3) digital CPU telemetry. Each telemetry object also belongs to a subsystem such as ADACS or EPDS. Subsystems are represented as class objects.

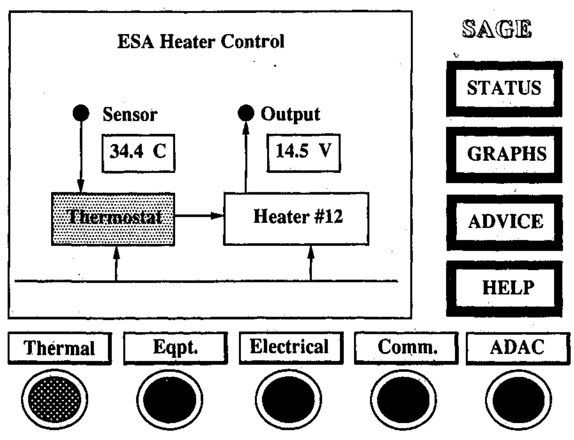

Fig. 2 SAGE interface example.

The second paradigm captures process knowledge about the class and instance objects using if-then rules. DMSP rule bases are organized into three types of rules: rules of observation, rules of interpretation, and rules of communication.

Rules of observation have telemetry mnemonics as rule conditions and derived telemetry objects as rule conclusions. Rules of observation are evaluated when the rule conditions receive values from the telemetry server process, either through a polling procedure or through a request for a particular telemetry parameter. Rules of interpretation evaluate derived telemetry objects. Rules of interpretation are evaluated in a forward chaining (data driven) or backward chaining (goal directed) fashion until a diagnosis of an anomalous condition is found. Rules of communication provide information to the user by message passing to the interface server.

Solar Array Drive (SAD) Hangup

The Solar Array is the major source of satellite power. It is the job of the SAD to maintain the orientation of the solar array towards the sun and to transfer power to the satellite subsystems. Due to the polar orbit of the DMSP satellite, the array must rotate continuously about the spacecraft. If the SAD becomes nonoperational, power loss will occur.

A Solar Array Drive malfunction (called a "SAD Hangup") is typical of the types of faults handled by REDMN. The following sections describe what happens onboard the satellite and in SAGE when a SAD Hangup occurs. This scenario is based on a SAD Hangup anomaly generated by the DMSP high-fidelity simulator, and reflects the actual operation of SAGE.

The nominal configuration of the spacecraft dictates that one of the onboard computers is in control of the spacecraft subsystems, but that the other computer is in control of attitude determination. It controls the SAD, as well as torque commands to the reaction wheels. The SAGE user interface indicates by lighted subsystem boxes that Computer 2 is in control, Computer 1 has attitude determination, and the SAD is operating nominally on side 2. The conditions that occur due to a SAD anomalous performance and REDMN actions follow: REDMN switches to the redundant side of the array drive electronics, crosses buses, and makes both of the computers "not OK." The operator is notified of these occurrences by status displays showing "Computer 1 not OK" or "SAD side 1" in red. English messages such as "REDMN has switched ADE sides, crossed buses, and made both SCPs not OK" and "RECOMMEND send ground command to clear SADSWIN in both SCPs" are posted and saved for future use. After the operator is notified of the anomalous conditions and has performed the recommended cleanup procedures, further information about the spacecraft is available from the user interface.

Conclusions

SAGE is an exploration of the issues in designing and building an intelligent satellite operator's workstation. The overall system architecture of SAGE has enabled the rapid development of a flexible system. Although only a prototype, SAGE has proven the value of expert systems technology in the interpretation of REDMN actions.

References

[1]Bost, J. D., Le, T. C., Mangan, P. K., Meloan, M. D., Sutton, S. A., and Turner, S. R., "Expert Systems and Hypermedia Systems Applied to Space Vehicle Monitoring and Control," *International Telemetering Conference Proceedings,* Instrument Society of America, San Diego, CA, 1989, pp. 97–101.

[2]Neuron Data, Inc., *Nexpert Object Fundamentals,* Palo Alto, CA, 1989.

[3]Johnson, T., Jr., *REDMN Analysis and Cleanup Procedures,* preliminary ed., RCA, 1984.

[4]V.I. Corporation, *DataViews User's Guide,* Amherst Research Park, Amherst, MA, 1989.

A. L. Vampola
Associate Editor

Rapid Prediction of Static Stability Characteristics of Slender-Wing Aerospace Vehicles

L. E. Ericsson* and H. H. C. King†
Lockheed Missiles & Space Company, Inc., Sunnyvale, California 94089

Introduction

MOST aerospace vehicles, both civil and military, although designed for hypersonic cruise at low angles of attack, often have to perform maneuvers at high angles of attack from low supersonic to low subsonic speeds. There is, therefore, a need for rapid prediction of the nonlinear high-alpha aerodynamics of slender-wing-aircraft configurations in that speed range.

The complexity of the flowfield on aircraft and aircraft-like configurations that operate at high angles of attack prohibits the use of numerical computational methods for preliminary design. Because of the continual changes in the early design, a purely experimental approach cannot be used either. One needs rapid computational methods to guide the early stages of preliminary design until a firmer design has evolved on which experimental and numerical methods can be applied.

Discussion

A fast prediction method, developed earlier for the unsteady nonlinear aerodynamics of pitching sharp-edged delta wings at high angles of attack,[1] has been extended to include the roll degree-of-freedom.[2] In the present Note, the predicted static lateral stability characteristics are compared with experimental results for slender wing and wing-body configurations.

Figure 1 shows a comparison between predicted and measured[3] lateral stability of a 74-deg sharp-edged delta wing. The agreement is good for $\alpha < 30$ deg. At high angles of attack, the leading edge vortices experience vortex breakdown. The breakdown boundary, determined by experiment,[4] is shown in Fig. 2. At angle of sideslip, the effective sweep angle becomes $\Lambda \pm \beta$ for the two leading edges. As C_{ℓ_β} in experiments usually is determined for $|\beta| \leq 5$ deg, the measurements will be affected by vortex burst when $\alpha > \alpha_{burst}$ for leading edge-sweep $\Lambda - \beta$ in Fig. 2. Thus, the measurements in Fig. 1 should be affected at $\alpha \geq 30$ deg, as $\Lambda - \beta = 74$ deg $- 5$ deg $= 69$ deg in Fig. 2. Thus, the prediction is not expected to agree with experiment beyond $\alpha = 30$ deg, a fact acknowledged by ending the prediction at $\alpha = 30$ deg. The experimental data indicate that vortex burst occurred in the test already at $\alpha < 30$ deg. This is the likely result of support interference.[5]

The agreement between prediction and experiment is certainly satisfactory for pure delta-wing configurations (see Ref. 1 and Fig. 1). However, the current interest is the low-speed aerodynamics of a hypersonic aerospace configuration, such as the one shown in Fig. 3. It can be seen that, aside from the zero shift expected to result from model wing and body camber, not accounted for in the prediction, the agreement with experiment[6] is satisfactory for preliminary design purposes. Again, the deviation at high α, $\alpha \geq 20$ deg, is probably caused by vortex burst, produced by support interference.[5]

Presented as Paper 90-0301 at the AIAA 28th Aerospace Sciences Meeting, Reno, NV, Jan. 8–11, 1990; received Feb. 3, 1990; revision received May 25, 1990; accepted for publication June 8, 1990.

*Senior Consulting Engineer. Fellow AIAA.
†Research Specialist. Member AIAA.