An Expert System
for
Satellite Control

Michael Pazzani and Anne Brindle
The Aerospace Corporation
2350 El Segundo Blvd.
El Segundo, California

ABSTRACT

As on-board satellite systems develop increased sophistication and autonomous capabilities, failures become fewer, but the diagnosis of the remaining failures becomes more complex. In addition, autonomy requirements for space vehicles are being issued along with requirements for reduced staffing of ground stations. Thus successful ground-based fault handling in the future will require greatly increased automation of fault detection and diagnosis. This paper investigates the use of an expert system as a ground system component for diagnosis.

The diagnostic cycle of the system is presented, along with requirements for its knowledge base. The results of implementing the design to diagnose part of a satellite attitude control system are given. Knowledge acquisition for this problem centered on the generation and analysis of terminal displays of telemetry which look much like strip charts. Correct diagnosis by the expert system derived from the use of extensive telemetry analysis, operations and satellite status databases, and satellite modeling.

Keywords: Expert Systems, Artificial Intelligence, Diagnosis, Satellite Control.

Satellite programs are now faced with requirements for increased survivability. Increased satellite autonomy will be a key to achieving survivability. However, some of the goals for autonomy efforts [6], such as automatic on-board component failure detection, diagnosis, and correction, have not yet been implemented even on the ground. It is not clear how many of the goals can be met by on-board automation. Maintenance procedures and some simple error checking and correction may eventually be placed on board, but diagnosis and correction of non-trivial anomalies cannot safely be divorced from development and review by staff on the ground.

In order to enhance survivability of ground functions, small, mobile ground systems are being designed [10]. Fixed ground stations are working towards more automation in order to reduce staffing requirements. Part of the automation, in both fixed and mobile ground environments, must be software to assist with anomaly diagnosis. However, to date there has been little work on systems for satellite diagnosis.

Many satellite programs are hoping that expert systems techniques arising from artificial intelligence research will solve the diagnosis automation problem. This paper outlines the issues involved in creating software for satellite diagnosis, and describes a prototype expert system, ACES, for diagnosis of a portion of the DSCS III Attitude Control System. The construction of a prototype revealed some of the difficulties of knowledge acquisition in the satellite domain, as well as of analysis of a dynamic system. However, it also illustrated the advantages of using artificial intelligence methods, software tools and hardware.

The next subsection describes the high level requirements of a system for satellite anomaly diagnosis. Subsequent sections define the problem addressed in the prototype system, how the system was built, the system components, a sample diagnosis by the system, and the conclusions reached.

System Requirements and Characteristics

Expert systems may be applied to anomaly diagnosis in two different environments. Large ground stations need an engineer's assistant, a system which can interact with experts to solve problems. Mobile units need a simpler, more compact, and more automatic diagnostic system. Both systems should perform the following functions:

- Fault detection. Generation of alarms indicating anomalous behavior.
- *Fault localization*. Diagnosis to reveal the locality and cause of the behavior.
- *Display focus*. Generation of operator displays which focus on pertinent information about the problem.

To achieve these functions, a system requires a sophisticated database, or knowledge base, containing satellite structure models, fault behavior models, and satellite status information. It must be capable of continuous, real-time operation. In particular, it must process telemetry such as sensor readings, satellite configuration information and command verifications. It must display telemetry intelligently, as well as producing diagnostic messages.

In addition, an engineer's assistant should include the following capabilities:

- *Database retrieval*. Display of the underlying knowledge base.
- *Explanation*. Tracing of the logic paths traversed during diagnosis.
- *Automatic Acquisition*. Acceptance of input from the engineer to update and expand the knowledge base and system processes.
- *Test generation*. Recognition of the importance of information which is currently missing in the input, and the capability to request completion of this information.

An engineer's assistant requires a query system, a knowledge acquisition system, and an explanation system, all working on the knowledge base and control mechanisms of the diagnostician. In order for an engineer to interact directly with these components, the interface must be domain oriented, and may involve natural language. The design of the diagnostician must include testing for incompleteness of information and risk assessment of the procedures which could procure the missing information.

In contrast, a system targeted for mobile ground environments must meet these requirements:

- *Passive diagnosis*. Diagnosis which makes the best assessment possible on the basis of incomplete information.
- *Limited modifiabilitv*. A knowledge base which is not easily altered by operators.
- *Field maintainability*. Military standard hardware and software.

It is not clear whether the last two items can be attained in a system which does anomaly diagnosis, or whether the problem of diagnosis requires special, flexible software and perpetual knowledge base modification.

A SAMPLE DOMAIN

To investigate the potential of expert systems for satellite control, we selected a single arena in which to do anomaly diagnosis, gearing the design towards the mobile ground station environment. The domain was the Attitude Control System of the DSCS III satellite, specifically the reaction wheels. Attitude control was selected because the majority of anomalies which seriously impact satellite mission occur within the attitude control subsystem. Also, the Attitude Control System is the most complicated and difficult to analyze.

Expert systems are suited to complex domains: their advantages are not visible when they are applied to straightforward problems.

DSCS III was chosen because at the time it was the test vehicle for another autonomy effort [1] and results could be compared, because the satellite is unclassified, and because the Aerospace Corporation possesses a simulator for the DSCS III Attitude Control System which could be used to generate telemetry tapes reflecting faulty behaviors. These tapes formed the test inputs to ACES, the Attitude Control Expert System for anomaly diagnosis.

SYSTEM OVERVIEW

ACES is a small expert system, intended to be a feasibility demonstration for automatic anomaly diagnosis. The core software is rule-based, although some components of the system are written in algorithmic code. The primary advantages of the rule-based approach are high level description of control mechanisms and easy extensibility. These allow the system to be constructed incrementally, while the domain is studied. The system, even when only partially constructed, acts as a catalyst to trigger analyses and criticisms during discussions with the domain expert.

Processing Stages

The processing in ACES has a number of steps:

Feature Extraction. The telemetry data is processed to look for unusual values. In addition to checking the bounds of telemetry values, the extractor also looks for unusual rates of change of various telemetry points over time. The purpose of this step is to reduce the amount of data with which the expert system must deal and to raise the level of abstraction of the data.

Fault Identification. The features extracted from the telemetry data stream are used as symptoms of particular faults. This step postulates a fault which could account for the behavior of the satellite as reflected by the symptoms.

Fault Confirmation. The hypothesis of a fault, proposed by the fault identification process, is confirmed or denied. Confirmation of a fault may require the examination of extracted features other than the symptoms used in fault identification, as well as the extraction of additional features from the telemetry stream.

Fault Implication Analysis. After the presence of a fault has been established, the behavior of the satellite is analyzed to identify those implications of the fault which manifest themselves as features extracted from the telemetry stream. This allows the system to avoid unnecessary diagnosis of symptoms pertaining to previously diagnosed problems, as well as correcting the system’s expectations of future telemetry.

Corrective Action Suggestion. A means of correcting or minimizing the damage caused by the fault is suggested. Typically, this step would recommend switching to a backup component. Ignoring the fault may also be an appropriate action.

Design Approach

ACES is implemented on a Symbolics 3600 computer in a combination of LISP and PROLOG. LISP is a general-purpose programming language widely used in artificial intelligence because of its symbol manipulation and functional capabilities. PROLOG is a logic programming language containing rules and facts. The execution of PROLOG programs consists of the execution of chains of rules leading to goals. If the initial chain followed fails to reach the goals, the PROLOG system backtracks through the links in the chain, searching for alternate rules to use in constructing new chains. The combination of LISP and PROLOG allows a great deal of flexibility in designing the system, since both algorithmic and rule-based approaches are natural to some system components.

There are basically two alternative ways of designing an expert system for diagnosis. The first is to provide the expert system with a description of the structure and function of the system and reason from the behavior of the system to a fault. This approach, which is sometimes called *deep reasoning* or *reasoning from first principles* was taken by Davis in his work [3] and Friedman with FAITH [5]. The alternative is to use a rule-based set of empirical associations which identify faults as a function of the behavior of the system, or fault symptoms. This approach, which is sometimes called *shallow reasoning*, was used by Shortliffe in MYCIN [8].

ACES combines deep and shallow reasoning. The feature extraction and fault identification steps use shallow reasoning. Their knowledge is represented as heuristic rules which represent empirical associations between features and potential faults. These rules were derived from interviews with an expert on the diagnosis of DSCS III anomalies. By using shallow reasoning for feature extraction and fault identification, the expert's experience could be utilized to quickly select a potential fault.

The fault confirmation and fault implication steps operate on a functional model of the satellite. In fault confirmation a model of the satellite is consulted to determine if the output history of the faulty component does indeed deviate from the expected behavior given the components input history. This step obviates the need for explicitly representing all of the exceptions to the fault identification rules. The model of the satellite is also used to predict the future behavior of the satellite in light of the faulty component. Using this sort of deep reasoning is appropriate here since an expert will often resort to a computer simulation of the satellite's behavior. Finally, corrective actions are represented as heuristic rules which suggest means for correcting the fault or for working around the fault to maintain satellite function.

Knowledge Acquisition

Once the architecture of ACES was selected, a major task was to put the specific information about DSCS III into the knowledge base. The Orbit Operations Handbook [4] was a good source of information for building a model of the Attitude Control System as well as for acquiring nominal bounds for the telemetry values. Details about unusual rates of change of telemetry signals were obtained through discussions with the DSCS III expert.

To obtain the expert's advice on the relationship between features and faults, we took advantage of the Symbolics 3600 bit-map graphics and mouse input capabilities to construct an interactive graphical display system. A sample screen display is given in Figure 1. This system allows the rapid display and manipulation of telemetry data from telemetry tapes and facilitated discussions on the features which are indicative of particular faults. The interactive display system is also used during ACES' diagnosis of faults to graphically display the data which is being used in the diagnosis.

SYSTEM COMPONENTS

The five processing stages described above are performed by two components in ACES. The Detector handles feature extraction: fault identification, fault confirmation, fault implication analysis and corrective action suggestion are handled by the Diagnostician. ACES also relies on a third component, the Front End Telemetry Processor, which decommutates and loads telemetry. This section describes these components in more detail.

Front End Telemetry Processor (FETP)

The preliminary telemetry processing is handled by a set of LISP routines, the FETP. These take as input the raw telemetry from a tape, and a file indicating the tape format. This file gives the frame size and subframe organization. Then, for each telemetry point delivered in the telemetry stream, there is an entry giving a unique mnemonic identifier, descriptor, size in bits, and all locations in the frame where the point is transmitted. Using this file, the FETP constructs a list of time-value pairs for every telemetry point and associates it with the mnemonic. The telemetry tape is basically inverted from a list of times with points and values for each time into a list of points with times and values for each point. This facilitates processing by the Detector.

The FETP is not considered to be part of ACES: rather, it mimics the initial telemetry processing already found in most ground stations. However, the FETP is satellite independent. All satellite-specific details of the telemetry reside in the telemetry format file. This approach, isolating satellite-dependent information as data and employing general purpose telemetry processing software, should be considered in all designs for common ground systems.

Detector

Feature extraction is accomplished by LISP routines which monitor the values of telemetry points. For each point monitored, the Detector has information on what it should consider an abnormal value or an unusual rate of change. This information is stored in knowledge frames [7], an approach which allows the material to be represented at its highest level of generality. For example, instead of being repeated for each of the four reaction wheels, bound check information is an attribute of a frame which represents those facts which are true of reaction wheels in general.

If a LISP monitor notices an abnormal value, it notes the value and any unusual rates of change which occur in a surrounding time interval (currently five minutes). These features are asserted as PROLOG facts in a database representing the state of satellite. Next, the diagnostic phase which operates on these features begins.

Diagnostician

Fault Identification. Fault identification is accomplished by PROLOG rules which implement heuristics acquired in discussions with the satellite expert. Each rule proposes a fault to explain the presence of a set of symptoms, or extracted features. For example, one ACES rule states that if a reaction wheel speed is zero and the wheel speed changed toward zero at approximately the rate that friction affects the wheel, then the wheel might not be receiving the signal from its drive.

There are also rules which compare extracted features to a set of predicted features derived from previous problems (see Fault Implication Analysis below). Those extracted features which match predicted features are marked as not requiring further fault identification.

Fault Confirmation. Once a fault has been postulated, other corroborating evidence is checked to confirm that the particular problem identified by the fault identification rules is consistent with the state of the satellite. This process either can confirm that a fault is present, in which case the implications of the fault are considered next, or can deny that the postulated fault is present, in which case an attempt is made to identify another fault via PROLOG's backtracking mechanism.

Ideally, the new evidence is independent of the symptoms which triggered the fault hypothesis. For faults arising in components, the hypothesis is often generated from symptoms in the output of the component. If the component has a functional description in the ACES knowledge base and the inputs to the component are available in the telemetry, an expected output can be generated and compared to the telemetry stream. If the comparison fails, the component can be confirmed as faulty.

Fault Implication Analysis. After a fault has been confirmed, the expected effects of the fault on the values of other telemetry data are assessed. A model of part of the satellite is used to predict values of telemetry which might be affected by the fault. ACES currently has one such model, a set of equations describing the on-board wheel control feedback loop. This model is implemented as a set of PROLOG rules which are derived from the description of the Attitude Control System in the Orbit Operations Handbook.

The predicted telemetry values are analyzed by the feature extraction routines to see if any of the predicted values are unusual. Descriptions of unusual predicted values are then saved and used later by the fault identification rules to avoid further diagnosis of the same problem (see Fault Identification above).

Corrective Action Suggestion. Heuristics obtained from interviews with the expert or from consulting the contingency plans in the Orbit Operations Handbook are utilized to suggest a corrective action. These heuristics are stated as PROLOG rules which produce a message on the display if a fault is confirmed.

AN EXAMPLE DIAGNOSIS

This section contains a detailed example of ACES diagnosing a fault in the Attitude Control System. To understand the logic followed by ACES, it is necessary to know something about the DSCS III reaction wheels. There are four wheels, arranged on the four sides of a pyramid as shown in Figure 2. Each wheel captures momentum around two of the three axes when it spins. One opposing pair contributes to roll momentum, the other to yaw, and all four contribute to pitch. The wheels are named PR+, PR−, PY+ and PY−, for pitch-roll-positive, and so on.

The wheel system is controlled in a tight feedback loop by the flight software. There is implicit one wheel redundancy if any one wheel is turned off, the feedback loop will correct the remaining three wheel speeds and the system will continue to function correctly. However, under nominal conditions all four wheels are kept spinning above a minimum value, to prevent bearing lubrication problems.

The example is as follows. First, a telemetry tape prepared by the DSCS III simulator is loaded by the FETP.

Feature extraction on the tape detects six unusual features in the telemetry stream. The wheel speeds of PR+ and PR− have obtained an unusual value of 0. Additionally, the wheel speeds of all four wheels have changed an unusual amount.

A fault identification rule hypothesizes a potential problem with PR−. It is assumed that this wheel is ignoring its drive signal since the value of the wheel speed is zero and the rate of change of the speed just before reaching zero was approximately the rate of change which friction would cause. (See Figure 3.)

The fault confirmation rules check to see if PR− is deviating from its expected behavior. It is discovered that the change in the wheel speed (the output, of the component) is attributable to the drive signal (the input to the component). Therefore, the hypothesis that PR− is ignoring its input is denied and the fault identification rules must find another possible fault to account for the detected features.

The same rule which suggested that PR− was faulty next implicates PR+. Again, it postulates that the component is ignoring its drive signal.

This time the fault confirmation rules indicate that PR+ is ignoring its drive signal, since the drive signal is decreasing but the wheel speed is increasing. (See Figure 4.)

After the fault has been identified and confirmed the implications of the fault are assessed. The model of the wheel control feedback loop is able to predict the affect on all four wheels of PR+ ignoring its drive signal. Predictions are generated for the speeds of the wheels and checked through the feature extraction routines. A prediction of zero for the value of PR− is noted as unusual, along with unusual rates of change on all the wheels.

Finally, fault identification rules compare the predicted features to the detected features and determine that all of the detected features have been satisfactorily explained. (See Figure 5.) The system suggests no corrective action, but displays a message that one wheel is now inoperative.

This particular problem is interesting in two ways. First, the fault identification rules in ACES know of only two reasons why a wheel speed would reach zero: a broken wheel drive or a broken tachometer. In the example, the speed of PR− reaches zero because it is responding correctly. ACES cannot at once explain this, but it does not rush to conclude that PR− is broken, and eventually the symptom is explained elsewhere.

Second, the situation would have been awkward to handle via shallow reasoning. The rule explaining the speed of zero for PR− could be stated as "if wheel X is known to be ignoring its drive signal, X services non-pitch axis A, the total momentum along A is zero, the wheel speed on wheel Y is zero and Y also services axis A, then no action is needed; Y is behaving correctly." This is not the sort of rule an expert could easily come up with. By reasoning from satellite models, ACES concluded the same thing without this rule.

EXTENSIONS

In order for ACES to operate reasonably within an actual ground station, the system would have to be modified to handle real-time telemetry processing and the dynamics of shifting bounds and telemetry formats.

Real-Time, Continuous Operations

In order for the Diagnostician to operate continually, it must perform garbage collection on its own working memory areas, It must also store a history of its results and conclusions in a database as it proceeds. If the system must halt because of system failure, degradation of satellite status data, or lack of fault models for completion of implications analysis, a procedure for manually updating the data and models and restarting the system must be supplied.

To accomplish real-time telemetry processing, the Detector should be decomposed into parallel, asynchronous monitors, each monitor checking the telemetry stream for its own features of interest. The FETP should fan the telemetry information out to all the monitors and to a storage mechanism. The feature messages and alarms generated by the monitors should be transmitted to an Alarm Center, a process which queues the messages according to a priority system and maintains a database of the messages received. The Alarm Center drives the Diagnostician by sending it messages for fault identification. It also can interrupt or cancel the Diagnostician's activity on one message in order to process a message of higher priority. The components needed for real-time processing are shown in Figure 6.

Automatic Dynamic FETP and Detector Processing

The FETP works from a file that describes the telemetry format. On many satellites, this format can be changed from the ground, or changes automatically with satellite mode changes. The FETP should be augmented with a command verifier which traps all commands that alter telemetry format and automatically updates the format file. It should also have a monitor which recognizes satellite mode changes and signals the required alterations to the format file, perhaps by switching to alternate files.

The Detector uses a database of bounds for checking the behavior of telemetry points. These bounds shift over time. They actually form a model of the expected telemetry, and are used to flag everything which deviates significantly from the expected. In current ground stations, these bounds are updated manually, and often are allowed to slip seriously, so that large numbers of false alarm messages are generated. The bounds database could be updated periodically by a program which employs a sophisticated model of the expected telemetry and a separate model of expected errors. If unexpected shifts occur, manual updates could still be made to the appropriate model, such as to the expected error range of a sensor in the case of a sensor degradation problem.

LIMITATIONS

ACES is a small prototype expert system, focusing on only a small part of the Attitude Control System. The Diagnostician contains on the order of thirty rules. It can diagnose two types of failures on any of four reaction wheels. In addition, the Detector flags sensor or drive failure symptoms on any of nine components related to the solar array panel. The satellite structures modeled are primarily electrical and logical (flight software related). Thermal, mechanical, physical, and environmental relationships were not considered.

Modeling the entire satellite might be difficult. Besides the expansion of ACES to handle the entire Attitude Control System and the creation of similar processing for the other subsystems of the satellite, the modeling and diagnosis of the interactions among subsystems would have to be considered. It is these interactions which often lead to complex problems that are both expensive for human experts to solve and critical to the survival of the satellite.

However, expert systems might be feasible for new satellites if the part of the knowledge base defining the satellite models were delivered along with the satellite as part of the formal documentation. Then the efforts of expert system and satellite engineers could be focused on rule construction for the diagnosis, instead of on the construction of the models in the knowledge base. The only caveat concerns the shortcomings of all documentation: it is not written by the satellite designers themselves, and is never complete, consistent, or correct. Operators and analysts diagnosing the satellite take this into account, and very often augment the documentation by querying the designers personally. An automatic diagnostician which operates from a knowledge base delivered as documentation will perform only as well as the documentation permits, and this will probably always fall far short of the behavior achieved by human analysts.

This leads to the problem of how to contract for the construction of expert systems for diagnosis. Contracts must have requirements documents. The requirements for a diagnostician should consist of a list of symptom combinations and the diagnoses to be delivered by the system when those symptoms are encountered. However, this is often not available, and often not understood prior to the attempted construction of the diagnostician. One major task in building a diagnostician is the describing of symptoms and their resultant diagnoses. The formulation of such requirements, and the organization of the satellite information in order to recognize the possibilities for diagnosis, are the task of the knowledge engineer. This task is undertaken by constructing a prototype expert system. Initial expert system efforts in this area, then, should be viewed as research and prototyping tasks designed to lead to requirements specifications.

Once the requirements are complete, the prototype may also be of assistance in formulating specifications for the diagnostician. The rule based approach may be retained in the final product, in order to provide flexibility and modifiability of the control logic and the database.

CONCLUSIONS

This project has demonstrated the feasibility of using an expert system for satellite anomaly diagnosis. Shallow reasoning about the telemetry stream to detect failures is followed by deep reasoning about satellite models to diagnose failures. During these phases, the system requires access to accurate information about satellite structure, previous failures, satellite state, telemetry format and expected telemetry values. Therefore, in order to maintain continuous operations, the system must be able to evaluate the implications of failures and update satellite state and expected behavior data.

Several subjects were touched upon during the development of ACES which require further investigation. First, the rule base of ACES should be extended. Expert system approaches often work well on toy problems, but collapse under the weight of a large domain, and satellite anomaly diagnosis is a very large domain. Second, the theoretical foundations of reasoning from physical models are now being discussed in the literature [2] and should be studied with respect to satellite systems and diagnosis. Good functional models and carefully designed rules for deep reasoning may provide the key to handling this large domain.

Communications Corporation to build an expert system for diagnosis of the DSCS III power subsystem [9]; much more work of this nature is needed.

REFERENCES

1. Arens, W.E. ARMMS Spacecraft Health and Maintenance Software Performance and Design Requirements. Technical Report 7030-15, Jet Propulsion Laboratory, 1984.

2. *Artificial intelligence, Volume 24, Numbers 1-3: Special Volume on Qualitative Reasoning about Physical Systems*. North-Holland, 1984.

3. Davis, R., Shrobe, H., *et al.* Diagnosis Based on Description of Structure and Function. Proceedings of the National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Pittsburgh. PA, 1982, pp. 137-142.

4. *DSCS III Orbit Operations Handbook, D3- WPC-536.* General Electric Corp., 1982.

5. Friedman, L. Diagnosis Combining Empirical and Design Knowledge. Jet Propulsion Laboratory, 1983.

6. Marshall, M.M. Goals for Air Force Autonomous Spacecraft. USAF Report SD-TR-81-72, Jet Propulsion Laboratory, 1981.

7. Minsky, M. A Framework for Representing Knowledge. In *The Psychology of Computer Vision*, Winston, P.H.. Ed., McGraw-Hill, New York, NY, 1975, pp. 211-277.

8. Shortliffe, E.H.. *Computer-based Medical Consultation: MYCIN*. American Elsevier, New York, NY, 1976.

9. Wagner, R.E. Expert System for Spacecraft Command and Control. Computers in Aerospace IV Conference, American Institute of Aeronautics and Astronautics, Hartford, CT, 1983, pp. 216-223.

10. Wagner, R.E. Modular Survivable Satellite Support. Proceedings of the International Telemetering Conference, Las Vegas, NV, 1984, pp. 409-419.

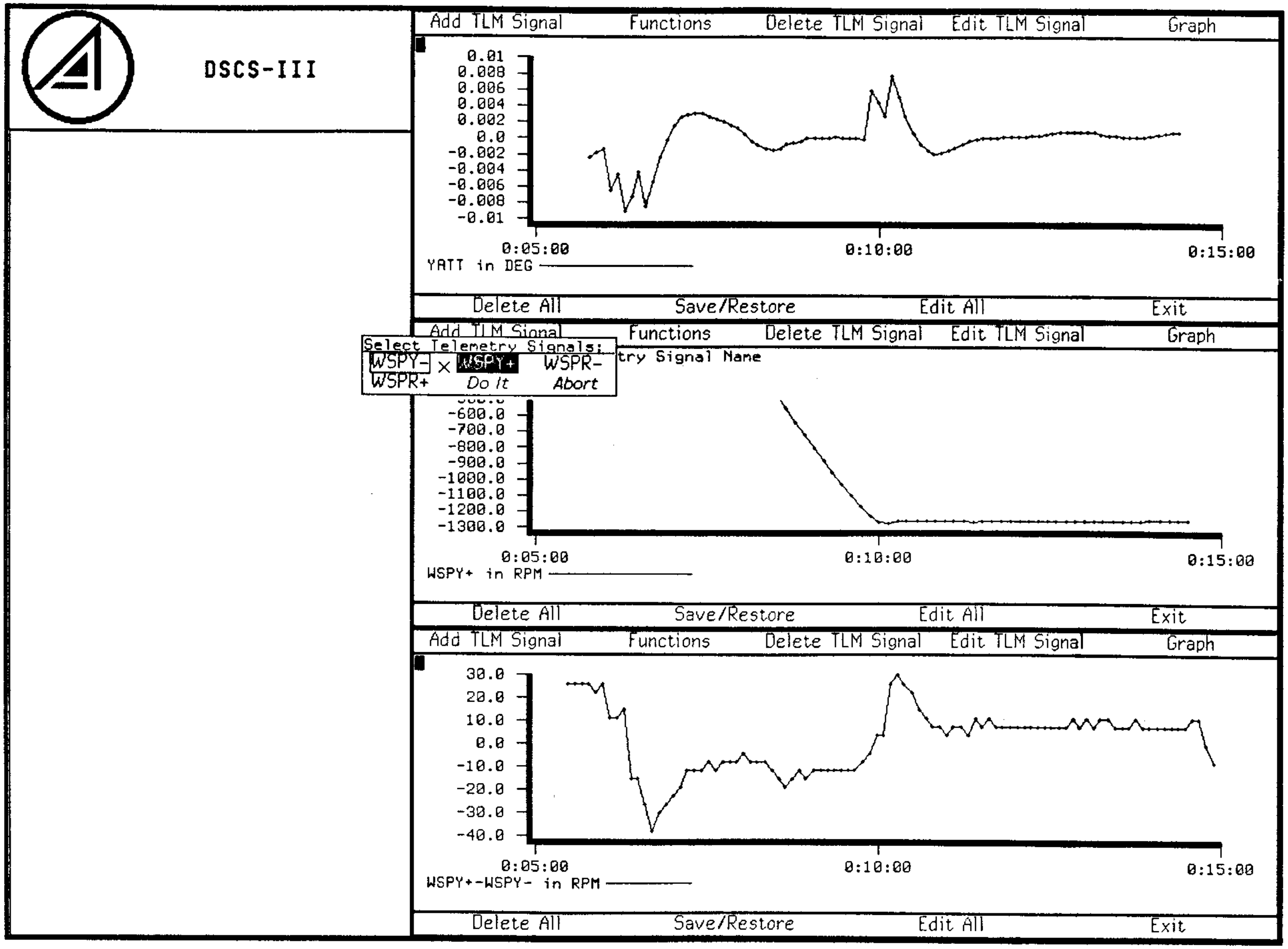

Figure 1. The ACES display. Explanations and advice will be added on the left, plots of telemetry values over time appear on the right.

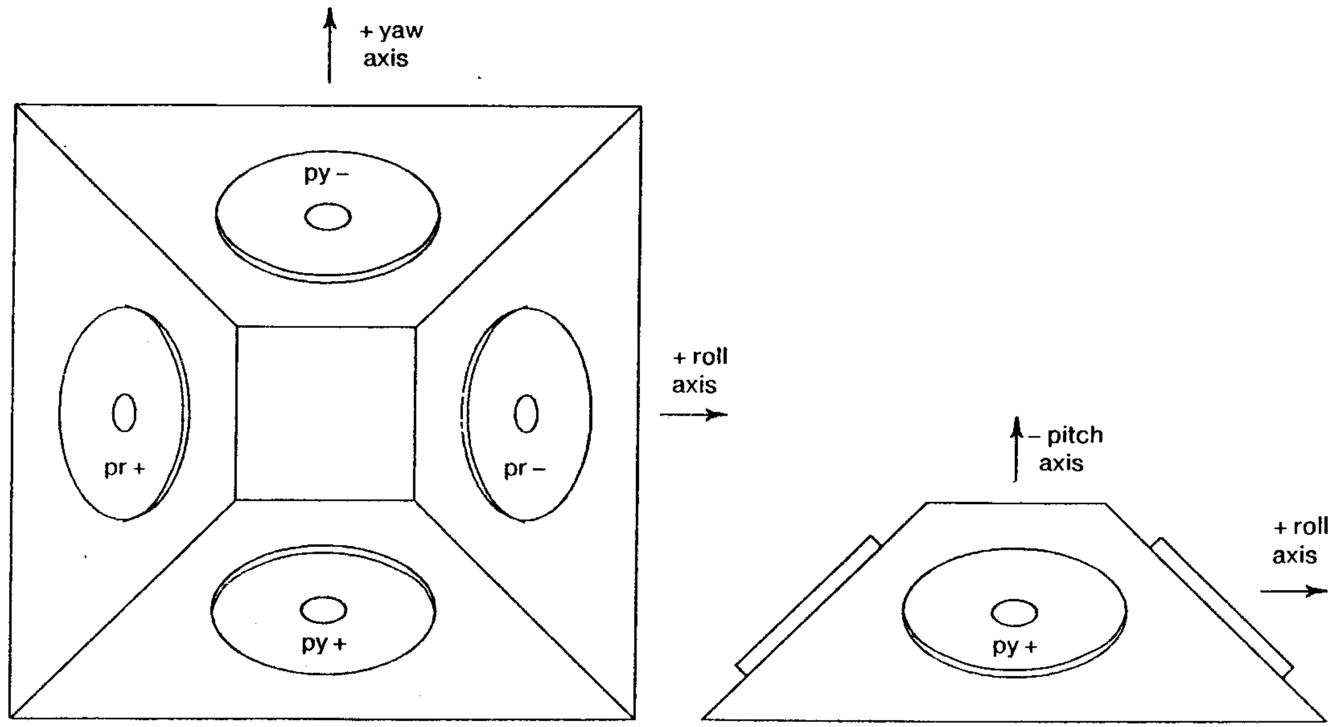

Figure 2. The mounting of the reaction of the Attitude Control System.
The top view (from the −pitch direction) is at left, a side view (from the −yaw direction) is at right.

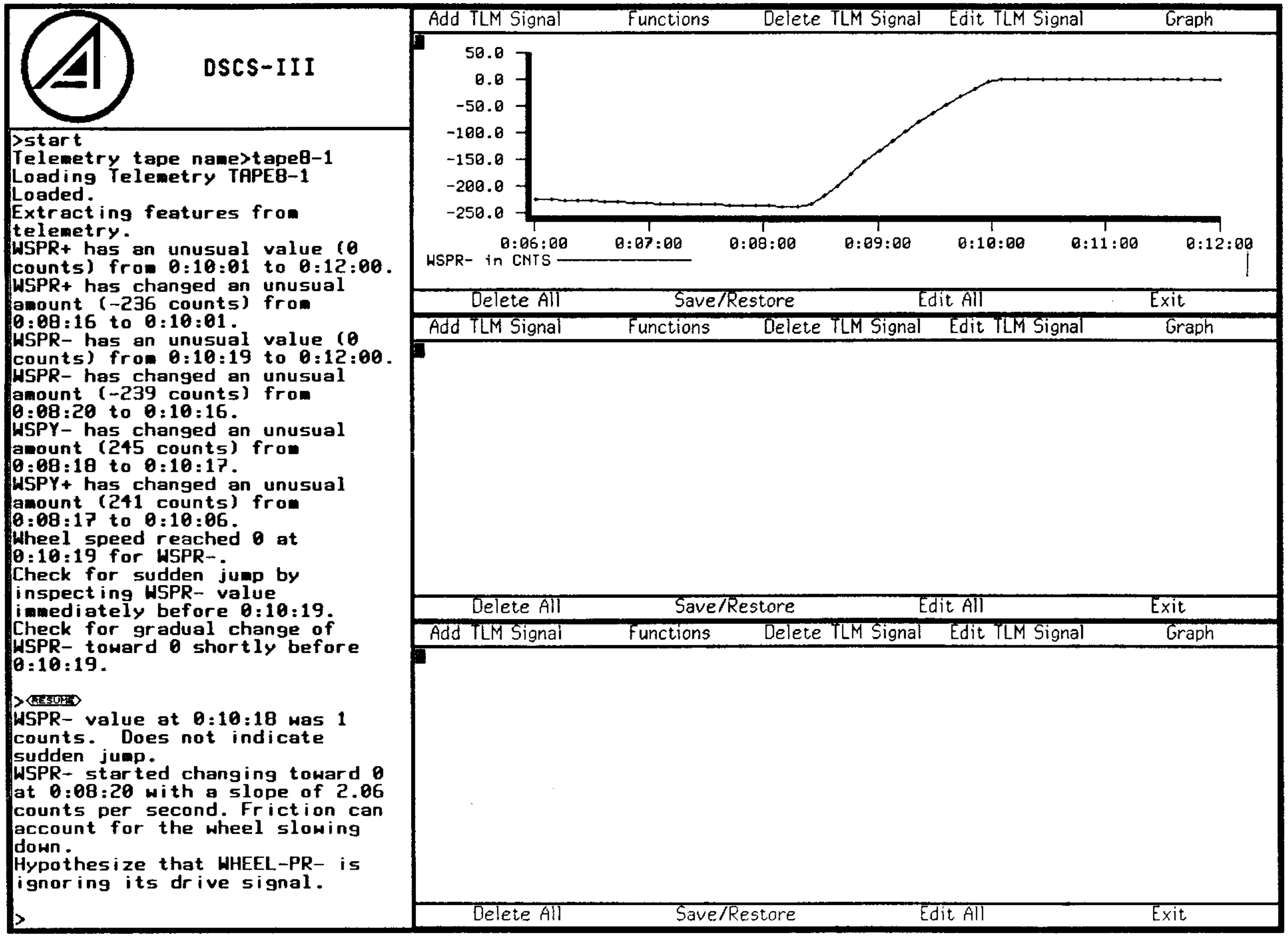

Figure 3. The speed of wheel PR−.

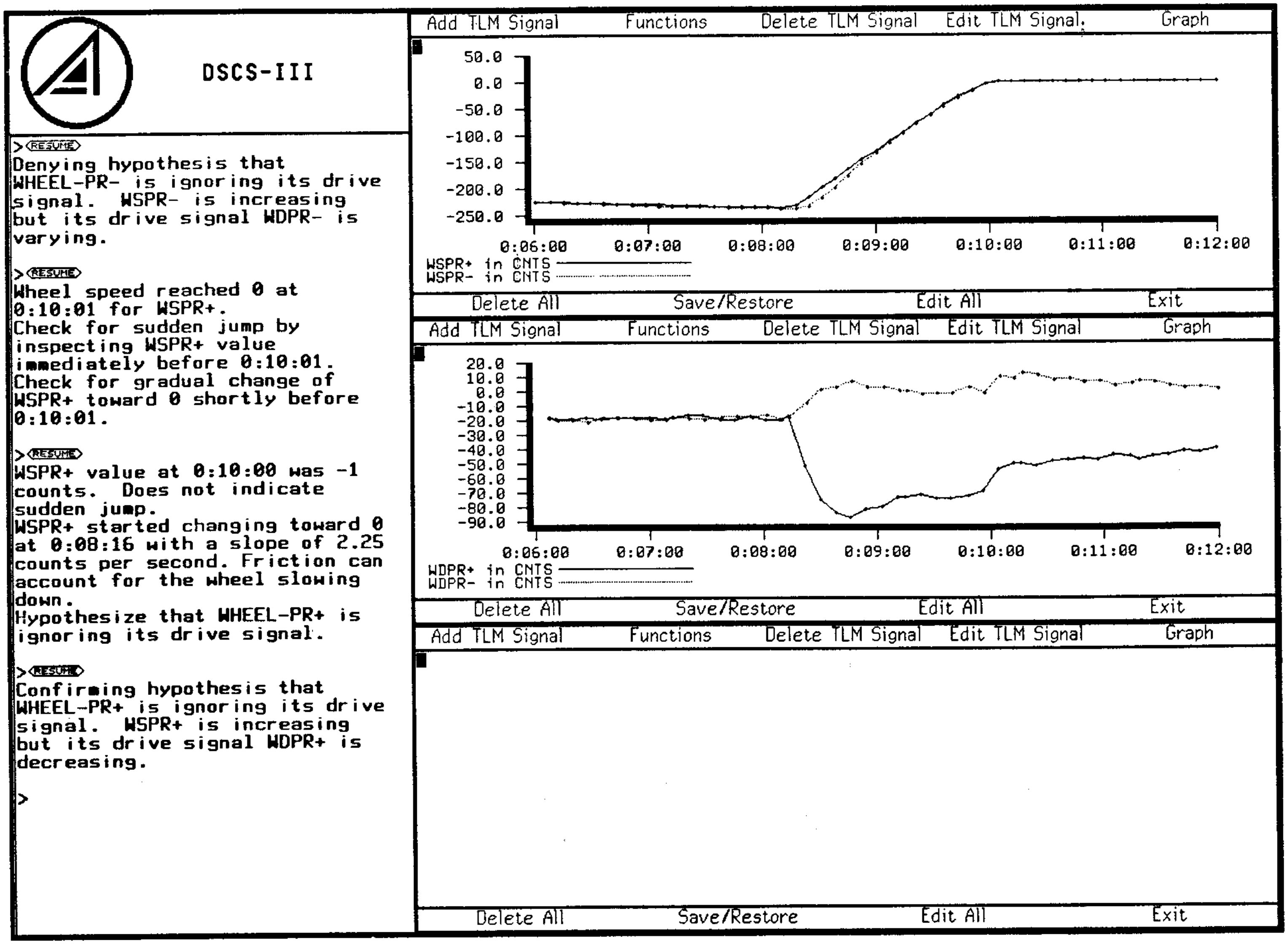

Figure 4. The speed (top graph) and drive (middle graph) signals of wheels PR+ and PR−.

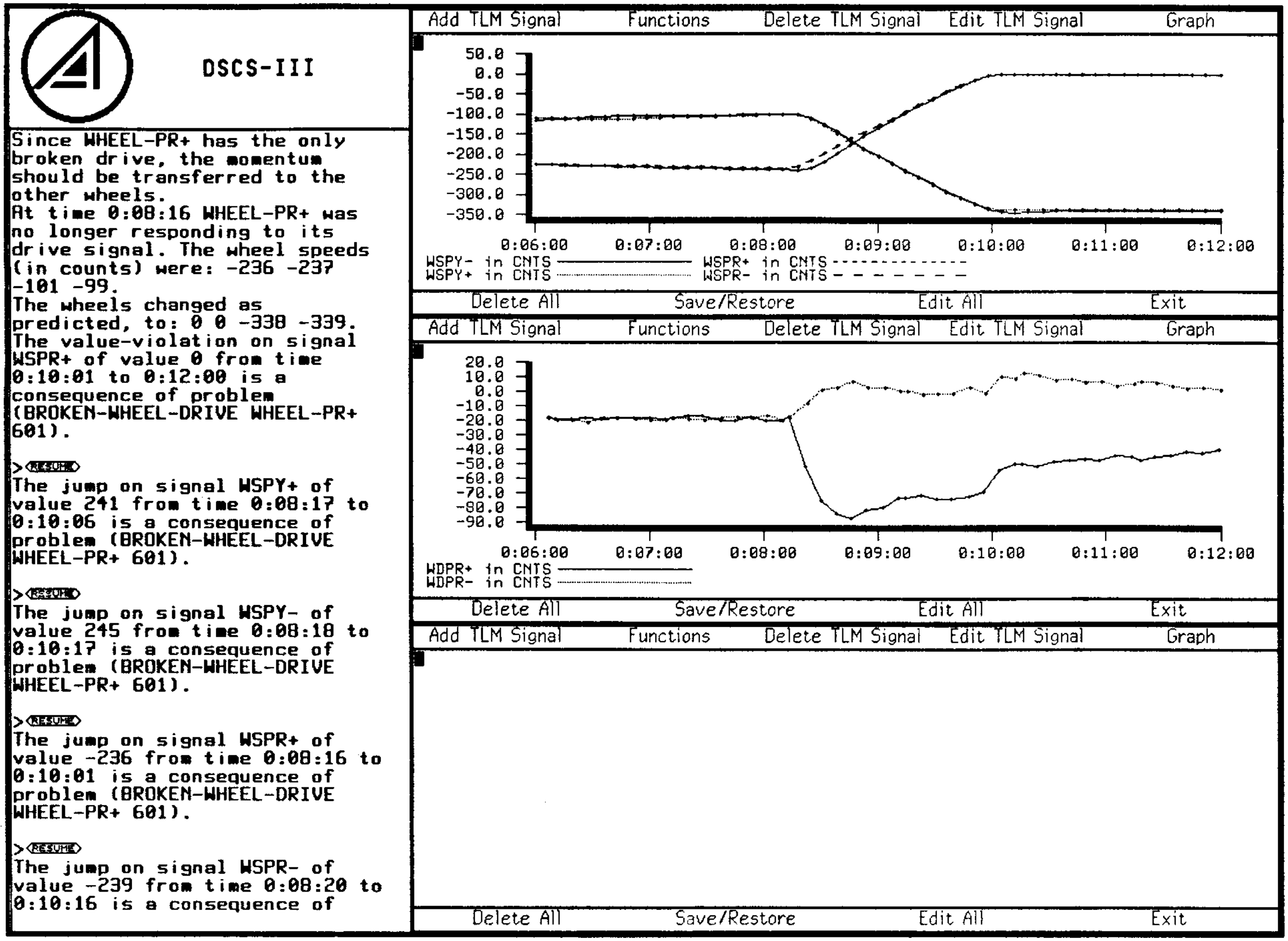

Figure 5. The unusual rates of change on all four wheels (top graph) are explained.

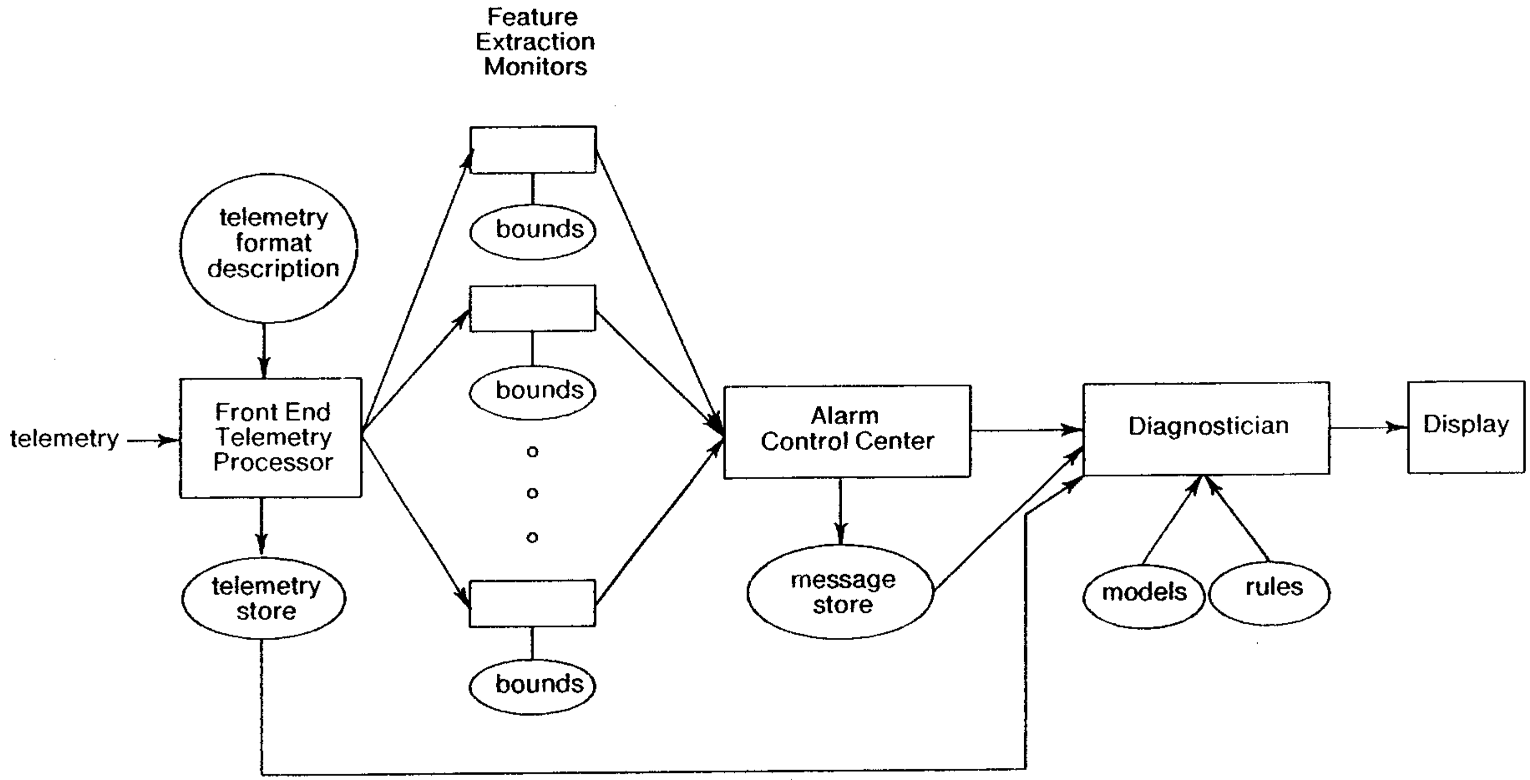

Figure 6. The components of a system to perform passive anomaly diagnosis.