TDRSS Momentum Unload Planning[*]

George R. Cross
Mitchell A. Potter
J. Douglass Whitehead
James T. Smith

Intelligent Systems Laboratory
Contel Technology Center
15000 Conference Center Drive
Chantilly, VA 22021-3808

Abstract

We describe a knowledge-based system which monitors TDRSS telemetry for problems in the momentum unload procedure. The system displays TDRSS telemetry and commands in real time via X-WINDOWS. The system constructs a momentum unload plan which agrees with the preferences of the attitude control specialists and the momentum growth characteristics of the individual spacecraft. During the execution of the plan, the system monitors the progress of the procedure and watches for unexpected problems.

1 Introduction

This paper describes a knowledge-based system to assist in the operation of the White Sands Ground Terminal (WSGT) for the Tracking, Data and Relay Satellite System (TDRSS). We first review knowledge-based systems that have been developed in support of spacecraft operations. We also review the specific systems previously developed in the context of TDRSS operations. Since our system acts as an assistant in the planning and execution of TDRSS momentum unloads, we present some background material on the concepts of momentum unloading in body-centered spacecraft. We then describe the problem and tool selection for the momentum unload system and give the current status of the system.

2 Expert Systems in Spacecraft Operations

2.1 Overview

The applications of expert systems to space systems operations can be broken into three main areas: planning/scheduling, diagnostics, and operations monitoring. We describe example systems in each of these areas.

Planning Systems – Planning and scheduling systems determine times when on-orbit assets can be used without conflict and may operate in real-time or more reflectively off-line. For example,

[*] Published in Telematics and Informatics, 8(4), pp. 253-265, Pergamon Press, 1991

Miller (Miller 1989) reviews expert systems for the Hubble Space Telescope. The obvious problem is to determine and optimize the sequence of observations but there is also an off-line system called the Proposal Entry Processor (Jackson, Johnston, Miller, Lindenmayer, Monger, Vick, Lerner, and Richon 1988) which examines proposals for the use of the telescope and searches for duplication among proposed observations.

Diagnostic Systems – The key issue in on-orbit spacecraft diagnosis and repair is that failing components often cannot be repaired. The few repairs to spacecraft by the Space Shuttle have been accomplished only for those in low earth orbit, not geosynchronous orbits. As a result, repair of satellites in geosynchronous orbit is usually limited to swapping in redundant components. Keller *et al.* (Keller, Baudin, Iwasaki, Nayak, and Tanaka 1989) describe the development of a rule-based expert system directly from a qualitative and quantitative description of the Reaction Wheel Assembly of the Hubble Space Telescope which appears useful in the pre-launch checkout phase. Since spacecraft are so inherently complex, automatic generation of diagnostic systems is necessary from model-based descriptions if there is any hope of breaking the knowledge acquisition bottleneck. The involvement of experts is still needed to refine system characteristics.

Spacecraft Monitoring Systems – Monitoring on-orbit spacecraft requires responsive ground software and a flexible user interface to aid the operational personnel in making decisions. Recent examples of such systems include the Real-Time Data System (Muratore, Heindel, Murphy, Rasmussen, and McFarland 1990) which is being integrated into mission control operations for space shuttle missions and the Spacecraft Health and Reasoning Prototype (SHARP) (Lawson and James 1989) which is being used for monitoring deep space probes like Voyager.

2.2 TDRSS Expert Systems

Previously constructed expert systems for TDRSS can be split into two general categories: those concerned with on-orbit problems and those concerned with problems in the communications network. Tang and Wetzel (Tang and Wetzel 1990) discuss a diagnostic system for the NASA Ground Terminal. The equipment in the ground terminal is concerned with data transport, data quality monitoring and line outages. FIESTA (Lowe, Quillin, Matteson, Wilkinson, and Miksell 1987) diagnoses the end-to-end TDRSS network from WSGT over to the actual site where data is presented to the end user. On-orbit TDRSS systems include ESSOC (Rook and Odubiyi 1986) which assists operators performing a delta-v maneuver on TDRSS Flight 1. ESSOC could use live TDRSS telemetry but was not deployed. MOORE (Howlin, Weissert, and Krantz 1988) and its successor PACES (Erikson and Hooker 1989) deal with the diagnosis of problems in the attitude control subsystem hardware but are primarily demonstration programs not intended for operational use.

3 TDRSS Operations

3.1 Momentum Unload Planning

3.1.1 Yaw Momentum

During nominal TDRSS operations (Group 1985, Volume 1, Book 1, 3-115ff) there is an angular momentum which is *normal* to the orbit plane. The angular momentum vector, through solar radiation pressure torques, picks up a component *in* the orbit plane. This is called the *transverse momentum vector* and is denoted by H_t. We can resolve H_t as two components H_{tx} and H_{tz}. H_{tx}

lies along the projection of the spacecraft roll axis into the orbit plane while H_{tz} lies along the projection of the spacecraft yaw axis onto the orbit plane. When there are no attitude errors, these are both exactly aligned with the roll and yaw axes. The *yaw attitude error* or *yaw angle* is related to H_{tx} by

$$\psi = \arctan(\frac{-H_{tx}}{H_n}) \tag{1}$$

where H_n is the instantaneous angular momentum along the orbit normal and has a nominal value of -32.535 ft-lbs-sec (fps).

The other component H_{tz} is expressible in terms of the reaction wheel speeds ω_1 and ω_2 by the relation in Equation 2 below. The number 366.8 is a conversion between the reaction wheel speed differences and torque expressed in fps (Group 1985, Volume1, Book1, p3-39)

$$\omega_1 - \omega_2 = 366.8 \times H_{tz}. \tag{2}$$

3.1.2 Purpose of Momentum Unloading

The term *secular torques* means the torques caused by slowly-varying phenomena, principally solar disturbances. Momentum unloading is required to reduce the effects of secular torques. In particular, yaw momentum unloading controls the quantities defined in Equations 1 and 2.

The first constraint is that ψ remains under 1.0 degrees (in radians, $\pi/180$). To see the effect of this, we first assume that $|\psi| \leq \pi/180$. After making a substitution for ψ from Equation 1, applying the tangent function to both sides, and accounting for the possibility of both positive and negative deviations, one obtains the relation

$$|H_{tx}| \leq \tan(\pi/180) \times H_n = 0.01745506 \times H_n. \tag{3}$$

More simply, assuming that H_n has its nominal value of -32.535 fps,

$$|H_{tx}| \leq 0.56790054 \approx 0.57 fps. \tag{4}$$

It also is required that the maximum yaw momentum is below the level that can be controlled by the roll/yaw integrator. The maximum that can be controlled is $1.50fps$, and a margin of error of $0.5fps$ is assumed. Thus, the maximum for H_{tz} is set at $1.0fps$. Setting $|H_{tz}| \leq 1.0fps$ and using Equation 2 yields the result

$$\omega_1 - \omega_2 \leq 367rpm. \tag{5}$$

3.2 Momentum Unload Operations

To unload the excess momentum stored in the reaction wheels, commands to perform a series of 50-msec thruster pulses are sent to the spacecraft by the satellite controller. This is procedure is essentially manual, initiated no later than two hours before the first thruster is to be fired ($T_0 - 2Hr$). Currently the satellite controller uses a checklist for performing the procedure which is prepared by an attitude control specialist during the planning phase of the momentum unload.

An abbreviated list of tasks performed by the satellite controller during the momentum unload procedure is as follows:

($T_0 - 2Hr$) – Verify the attitude control specialist has included essential data on the checklist such as the anticipated change in reaction wheel speed, the optimum start time for the dump, and the number and polarity of thruster pulses required. In addition, verify the status of the attitude control subsystem is nominal.

$(T_0 - 1.5Hr)$ – To achieve maximum thruster efficiency, ensure the catalyst bed heaters on the spacecraft are on. Also, ensure a strip chart recorder used to monitor various parameters during the unload operation is configured properly.

$(T_0 - 45Min)$ – Check the configuration of the propulsion system and ensure the valve drive electronics are enabled.

$(T_0 - 15Min)$ – Load the thruster pulse widths into random access memory on-board the spacecraft.

(T_0) – Transmit a command block consisting of a reaction wheel momentum change command (pre-emphasis) and a command to fire the appropriate thrusters.

$(T_0 + 4Min)$ – Verify the results of the thruster firing is as expected and determine if more thruster pulses are required.

4 System Development

4.1 Task Selection

A *task analysis* is a generic name for a set of observational techniques in which the tasks that people have created to perform their jobs are studied by observations and interviews. The tools and information that are used to accomplish the tasks are also noted.

To initiate the TDRSS expert system project, artificial intelligence and human factors specialists from the Intelligent Systems Laboratory conducted a task analysis at WSGT of procedures related to controlling the attitude of TDRSS. The purpose of this task analysis was to identify possible applications of expert system technology for increasing the efficiency of the satellite controllers. The results of this particular task analysis documented:

- Who performs which tasks
- The number and proportion of tasks people perform
- The order in which the tasks are done
- The time it takes to complete the tasks

The approach taken was to conduct interviews with satellite controllers and attitude control engineers in which they were asked to verify and refine 25 flow diagrams of procedures related to the attitude control of the spacecraft. The flow diagrams, initially prepared at the Intelligent Systems Laboratory based on information from the TDRSS Operations Handbook (Systems 1984), describe who is involved in each procedure, what hardware and software is used, what events make it necessary to execute each procedure, and what the inputs and outputs of the procedures are. The flow diagrams were based on a methodology called Structured Analysis and Design Technique (SADT) (Marca and McGowan 1988). The satellite controllers and engineers were also asked to identify the frequency and time required to perform each procedure. In addition to conducting interviews, approximately twelve hours of observations of the satellite controllers were made in the TDRSS Control Center over three days and two shifts. Figure 1 shows the average number of hours per year a satellite controller spends on the 25 attitude control tasks for TDRSS Flight-3.

Three categories of tasks were considered for the application of expert system technology: diagnostic tasks, scheduling tasks, and tasks related to the performance of routine procedures. At

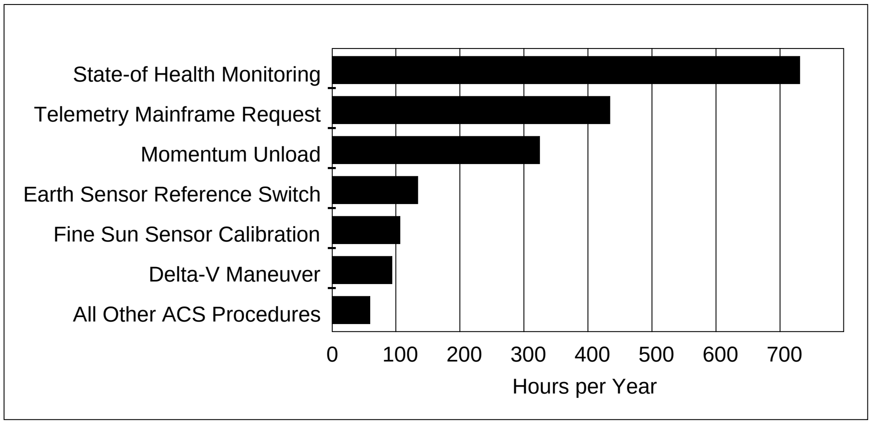

Figure 1: Satellite Controller Time Devoted to Attitude Control of TDRSS

the request of WSGT management, only the attitude control subsystem of TDRSS was considered for initial expert system enhancement.

In the context of TDRSS attitude control, a diagnostic assistant would diagnose existing and pending attitude related spacecraft component failures. These components include the gyro reference assembly, valve drive electronics, control processor electronics, etc. The task analysis revealed that component failure diagnosis in a system as complex as TDRSS ranges from trivial to extremely complex. The complex diagnostic tasks are currently difficult for teams of experienced spacecraft engineers to solve; therefore, it is not realistic to expect this problem to be solved by an expert system given the current level of technology. Progress on expert systems for complex diagnostic tasks is being made using model-based reasoning, but this work is still in the research stage. Two additional reasons revealed by the task analysis for not pursuing a diagnostic assistant include the discovery that diagnosing spacecraft component failures is primarily an engineering and management rather than a satellite controller function and the discovery that TDRSS components seldom fail.

A scheduling system for TDRSS attitude control would schedule routine procedures in a way to optimize satellite controller time. The task analysis showed satellite controller scheduling to be less of a problem than anticipated. The interaction of WSGT personnel performing satellite controller scheduling and the infrequency of scheduled tasks create an environment in which benefits provided by an expert system would be minimal. In addition, much of the information utilized by scheduling personnel is not currently available on-line and would require extensive modifications to existing procedures before being accessible to an expert system.

An expert system to aid the satellite controller with the planning and execution of specific routine procedures would determine when the selected procedures should be executed, automate parts of the procedures such as the verification of parameter values and the performance of calculations currently done by hand, and guide the satellite controller through the areas of the procedures that require commanding the spacecraft. The expert system would initially encompass a few high payoff procedures. Its knowledge could then be incrementally expanded to include more and more procedures with the goal of eventually covering the entire operation of the spacecraft. The task analysis showed the state-of-health monitoring procedure and the momentum unloading procedure to be good initial candidates for such an expert system. Figure 1 shows state-of-health monitoring to be the most time consuming satellite controller procedure, requiring approximately 730 hours of satellite controller time per year for each spacecraft. The primary function of an expert system aid for state-of-health monitoring would be to intelligently filter alarms. By reducing the number of false alarms with an intelligent alarm filter, the time devoted to this task by the satellite controller could be significantly reduced. Figure 1 also shows that the satellite controller spends about 320 hours per year performing momentum unloads for each spacecraft. In addition, each attitude control engineer spends about 170 hours per year doing the planning required for the momentum unload task. An expert system would be able to completely automate the planning done by the engineer and reduce the time required for the satellite controller to perform the task. The expert system would reduce satellite controller time through the monitoring of telemetry throughout the procedure to ensure the spacecraft is configured correctly and is responding as expected to momentum unloading commands, and by leading the satellite controller through the procedure, suggesting the correct commands to be sent to the spacecraft.

In conclusion, the task analysis suggested an expert system that helps with the execution of routine attitude control procedures would provide the most benefit to the satellite controllers. In addition, such an expert system stands the greatest chance of success given the current state of artificial intelligence technology.

4.2 Knowledge Engineering Tools

Knowledge engineering tools for building the TDRSS attitude control expert system consist of a blackboard environment and a production rule extension to the language C. Both of these tools were developed at the Intelligent Systems Laboratory.

4.2.1 Blackboard Environment

The blackboard paradigm (Nii 1986) is appropriate for the solution of problems from domains which, although fairly complex in total, can be partitioned into smaller subproblems and subtasks. It provides an approach to the solution of problems in such a problem domain via the coordination and management of these smaller subproblems and subtasks. The general framework that implements the blackboard paradigm consists of a collection of knowledge sources containing the reasoning and processing facilities required to solve the various subproblems of the problem domain and a blackboard state space that is shared globally by these knowledge sources.

Knowledge sources – Each knowledge source contains those reasoning and processing facilities required to solve one of the subproblems of the problem domain. Each is implemented in some manner, such as a rule-based system, function call, process, or asynchronous task, appropriate to the particulars of that subproblem or subtask and to the environment in which they are used.

Blackboard – The blackboard state space, referred to either as blackboard shared memory, or simply as the blackboard, provides a data structure for representing the problem-solving state. Each knowledge source has access to the current state of the problem's solution by reading this state space, and may communicate requests for assistance as well as report the results of its solution efforts by writing to this state space.

The blackboard system implementation used to build the TDRSS attitude control expert system consists of two layered subsystems: 1) an object system which is the basis of a shared memory database that provides specialized support for the declaration and the run-time management of control-events associated with shared database elements, and 2) a set of functions and structures that support the association of knowledge sources with these control-events, and the customization and activation of a control loop manager which governs the selection and activation of knowledge sources.

The underlying object system on which the blackboard system is implemented is a general-purpose dynamic object system possessing many of the major features and characteristics of LISP-based object systems such as FLAVORS and LOOPS. The basic objects of the system are its classes and their instantiations. Both of these are characterized by their attributes. Blackboard shared memory is implemented as an object-oriented database that is shared by some collection of knowledge sources as their medium of communicating the global status of a common problem's solution with other knowledge sources.

Within this blackboard system implementation, the coordination and cohesion of knowledge sources is provided by a control loop manager which coordinates the activation of knowledge sources. This manager functions as a mini-operating system and database manager as it regulates the interactions of knowledge sources via the blackboard database.

4.2.2 Rule System

Some of the blackboard knowledge sources of the TDRSS expert system are being developed using a simple production rule extension to the language C. Rule systems have proven to be a useful method of modeling the problem solving activity of experts. They are data driven, easily extensible, and understandable by both the knowledge engineer and domain expert. Rule systems are programs consisting of production rules, a global data base, and a control paradigm.

Production memory – Production rules are entities made up of an antecedent (also known as an if-part or left-hand-side) which specifies when the rule is applicable, and a consequent (also referred to as a then-part or right-hand-side) which specifies what actions to perform if the rule is executed. The production rules are collectively referred to as production memory.

Working memory – The global data base of a production system is called working memory. Working memory provides the context in which antecedents are true (satisfied) or false (unsatisfied).

Control paradigm – The control paradigm of a production system is a recognize-act cycle in which rules with satisfied antecedents are identified, a single rule is selected from the satisfied set by a conflict resolution procedure, and its consequent is executed. Identifying rules with satisfied antecedents is referred to as pattern matching.

Production memory in the rule system consists of one or more collections of production rules called rulesets. Rulesets provide a mechanism for grouping rules so the recognize-act cycle can be restricted to a subset of the total collection of rules defined by an application. A production rule antecedent in the rule system consists of a conjunction of clauses. The clauses are partial descriptions of elements of working memory. These partial descriptions are referred to as patterns. An antecedent is satisfied if data exists in working memory precisely matching its patterns. A production rule consequent consists of a C compound statement. This code is executed in the context of the working memory data which satisfied its corresponding antecedent. Specifically, the consequent code has access to variables bound to a user defined subset of the working memory data used in the pattern matching process. Rule consequents consisting of arbitrary C statements is a departure from many other production systems in which both the antecedent and consequent are written in a restricted production system language.

Working memory in the rule system consists of a collection of object instances similar to those mentioned in the blackboard environment description. The blackboard system and the rule system will be integrated such that a region of the blackboard state space may assume the role of working memory.

The rule system recognize-act cycle operates in a forward-chaining mode. In a forward-chaining production system, the process of determining which rules should be in the conflict set is based entirely on the current contents of working memory. The conflict set simply consists of all rules with satisfied antecedents. This is in contrast to a backward-chaining system in which rules are added to the conflict set based on a set of currently active goals. The forward-chaining recognize-act cycle continues until either a solution to the problem is reached, there are no more rules with satisfied antecedents, or a predetermined number of recognize-act cycles have been executed.

A rule system blackboard knowledge source consists minimally of one or more rulesets and C code for initiating forward-chaining on the rulesets. Rule antecedents are compiled by a rule compiler into a discrimination network so that the determination of whether or not they are satisfied may be done with the Rete Match Algorithm, a high performance algorithm for pattern matching (Forgy 1982). Rule consequents are translated by the rule compiler into C functions to be further compiled by a C compiler.

5 System Status and Overview

We are now in a position to explain the architecture of the momentum unload system and how it fits into WSGT operations. We use the blackboard paradigm and encode sections of the monitoring process as knowledge source.

5.1 WSGT LAN

The ground station equipment at WSGT is on an Ethernet LAN. Processors on the LAN communicate via *messages* which have a fixed part and a variable part. Each message identifies the time, the source of the message, and the spacecraft. Processors on the LAN operate in a client-server mode and send and receive messages for status values. Messages are identified by a message code.

Our charter for the development of the expert system precluded the possibility of any changes to the ground station software. Thus, our system operates in *wiretap* mode on the Ethernet and receives all of the messages sent and filters out the irrelevant ones. In the initial version, only messages dealing with changed telemetry values and TDRSS commands sent to the spacecraft are examined. Later, as our system expands to handle other facets of TDRSS operations, we may need to examine other messages such as those dealing with user service requests.

5.2 Outline of System Components

The major components of the system are show in Figure 2. We now discuss their functions.

Wiretap – The Wiretap module examines all incoming packets on the WSGT and reconstructs them into messages. As we have mentioned, messages have a fixed part and a variable part. For example, messages of type 748 contain telemetry values which have changed. The changed values follow the header information and are of variable length.

Router – The Router module uses TCP/IP sockets and sets up connections to processes to receive messages. An arbitrarily large number of processes can sign up as clients of the router. Current clients include one telemetry display module per spacecraft and the expert system itself.

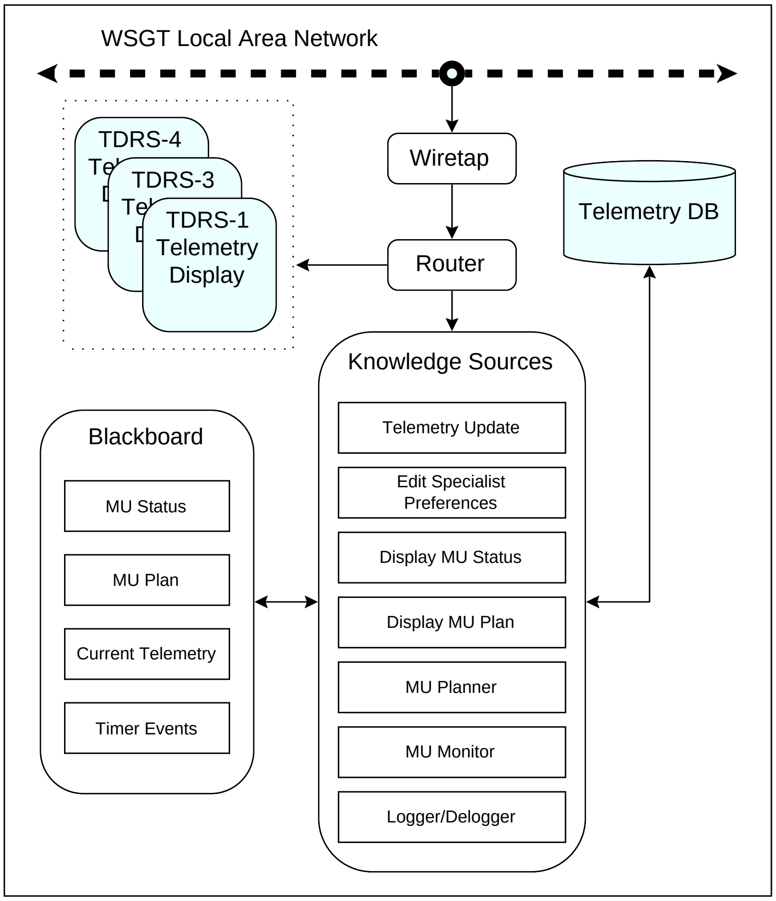

Figure 2: Expert System for TDRSS Momentum Unloading

Telemetry Display – Our telemetry display modules, one for each of the currently on-orbit spacecraft, mirrors the current nominal operating displays now in use at WSGT. After future discussions with the ACS specialists, we expect to add additional display capabilities including trend analysis. The telemetry display was set up outside of the blackboard structure to guard against any performance problems in the display of the telemetry while the blackboard process is running.

Telemetry Database and the Log/Delogger – We have defined a flat file and a simple retrieval system to hold telemetry. Adding items to the database is called *logging* and retrieving them is known as *delogging*. We expect that refinements to the system will have to be made over time, so we have allowed the possibility of saving historical telemetry at various time resolutions. For example, a resolution of about 6 hours is needed to plan a momentum unload under current operational procedures, but anomaly resolution is best done if *every change* in the telemetry is available.

5.3 User Interface and Status Display

The user interface is based on X-Windows with the Sun Microsystems user interface development tool Guide. Within the structure of the blackboard knowledge sources, there are a a number of display and user interface programs:

Telemetry Display – We have mentioned already that display of the telemetry is done outside of the blackboard as shown in Figure 2. The telemetry display shows about 70 telemetry values and has the capability to show another 10 values of the users choosing at any time. We expect that the displays will be modified over time to correspond to specialist preferences.

Edit Specialist Preferences – In current TDRSS operations, there is one ACS specialist assigned to each of the spacecraft. The specialist preferences represent information that the expert system can access as part of its expertise and can be viewed as a collection of parameters to control momentum. For example, it might include the actual shift schedule of the specialist so that no momentum unload would be scheduled while the specialist was off-site.

Display Momentum Unload Plan – At any given time, there is an anticipated time to do the next momentum unload based on the momentum growth profile of the particular satellite. This display gives a graphical representation of the recent momentum growth and presents in tabular form the time to do the next momentum unload and the sequence of appropriate thruster firings needed to accomplish it.

Display Momentum Unload Status – This display identifies each step of the momentum unload and determines which step we are in. We describe this process in more detail in the monitoring section below.

5.4 Expert Momentum Unload Planning

Two currently existing programs are used by WSGT attitude control specialists to plan TDRSS momentum unloads. Both of these programs use the same basic algorithm:

1. Determine the roll and yaw inertial momentum (H_{x_i}, H_{z_i}) at the preceding two 0600 and 1800 local spacecraft time (LST) points.

2. Using linear extrapolation, estimate H_{x_i} and H_{z_i} at the next 0600 and 1800 LST points.

3. Compute an average inertial momentum estimate halfway between the 0600 and 1800 LST estimates.

4. Determine inertial momentum target points (typically by accepting them as input from the specialist).

5. Compute the optimal dump time, appropriate thrusters, and the number of thruster pulses needed to achieve the inertial momentum target points.

To help the attitude control specialist visualize the momentum growth of the spacecraft, the existing programs plot roll and yaw momentum in inertial coordinates in three hour increments beginning at the first data point used in the estimation and continuing through the current momentum of the spacecraft. The estimated points are also shown on the plot.

Although these programs are an improvement over the earlier paper and pencil method of planning momentum unloads, the procedure can be fully automated by capturing the knowledge of an attitude control specialist in an expert system and giving the expert system on-line access to spacecraft telemetry. Access to telemetry is achieved by connecting the expert system to the WSGT LAN as has been previously discussed. Attitude control specialist knowledge is being captured in a collection of production rules developed incrementally through a series of interviews between a knowledge engineer and an attitude control specialist.

Access to telemetry gives the expert system the ability to do momentum growth trending. The momentum growth profiles of the individual spacecraft drift slowly over time. By observing the growth profiles over several weeks, the expert system can estimate future inertial momentum data points with a higher degree of accuracy than is currently possible. This capability will enable the expert system to compute target points that will extend the time between unloads; an advantage for spacecraft such as TDRSS in which momentum unloads are not handled automatically by on-board firmware. Human attitude control specialists currently strive to extend the time between unloads using mental models of momentum growth built from their personal observation of the spacecraft over many months.

Another use of momentum growth trending is to enable the expert system to recognize spacecraft maneuvers and erroneous data points. Both of these conditions are suggested by discontinuities in momentum growth. A maneuver (*e.g.* delta-v) is recognized by a shift in inertial momentum followed by a translated momentum growth with the same slope. If a maneuver is detected, the expert system will still be able to accurately estimate future points through the use of the earlier trend. Erroneous data points often occur at the transition between the portions of the orbit in which the spacecraft yaw angle (one of the parameters used to compute inertial momentum) is computed from telemetered values of the fine sun sensors (observer mode), and portions of the orbit in which the yaw angle must be estimated from earth sensor and wheel speed values (estimator mode). These points are characterized by a few discontinuous points followed by a return to the original growth line. Below a specified threshold, the expert system will simply discard erroneous points.

The attitude control specialist also uses scheduling knowledge in the development of a momentum growth plan. For example, the selection of a target point may be influenced by the current day of the week. If a particular target point would cause the critical planning phase of the next momentum unload to occur on a day in which the specialist is not on-site (*e.g.* weekend or holiday), the target point would be adjusted if it were possible to do so and not exceed normal operating ranges. Another example of the use of scheduling knowledge is the use of solar eclipse information to avoid momentum unloads when the spacecraft is running on battery power. The expert system will access a database of this type of information specific to each spacecraft and use production rules to constrain planning decisions.

Although momentum unload planning will be automatic under normal conditions, the attitude control specialist will be alerted by the expert system when momentum growth does not follow the expected trend.

5.5 Momentum Unload Monitoring

The momentum unload process has a sequence of steps which are always followed and is defined by a combination of telemetry values and spacecraft commands. Since the monitoring occurs without interaction, *i.e.*, no data can be requested or repeated, the monitor waits for the appearance of telemetry value changes and commands that signal the beginning of a momentum unload and indicate its progress.

The first few commands for a yaw momentum unload are shown in Figure 3. TDRSS normal mode means that the attitude control is under control of the control program electronics and that the earth sensor is providing feedback to maintain attitude. A momentum unload is best characterized by a thruster firing in TDRSS normal mode. Other maneuvers like a delta-v (change in momentum) are accomplished in earth mode; thus the presence of normal mode telemetry plus the beginning of a thruster firing sequence indicates a momentum unload. In Figure 3, the first steps to initiate a thruster firing include turning on the heaters for the catalyst bed (required for the hydrazine thrusters), turning on the valve drive electronics, and sending commands to initiate a sequence of thruster firings.

1) Valve Drive Electronics Enabled

2) Proper Catalyst Bed Heater On

3) Thruster Pulse Width Command Sent

4) Proper Thrusters Enabled

5) Proper Pre-emphasis Command Sent

⋮

Figure 3: Momentum Unloading Indications

We observe these steps under the assumption that all data is provided and is accurate. On the other hand, because of ground station LAN failures and RAM hits (single-event upsets in TDRSS on-board memory), some telemetry values may be missing or corrupted and some commands, although issued, may not reach the spacecraft. The other problem we face is that the messages of type 748 sent along the LAN indicate only changes in telemetry values and if we miss a changed value, there is no way to refresh it in our monitor. However, every hour all telemetry values are refreshed, so our database will become correct over time. We support the development of multiple lines of reasoning to reach conclusions about our position in the momentum unload process. For example, if a command setting the thruster pulse width is noted but the catalyst bed heaters were not turned on, we would assume that a momentum unload is in progress and at the third step even though we do not have every required piece of evidence for this conclusion.

The other feature of the monitoring is integration of the momentum unload plan. The current momentum unload plan is available on the blackboard and can be accessed by the monitor. The monitor can thus estimate the number of thruster firings and the expected time of the momentum unload. As these steps unfold, the plan which the momentum unload plan produces is yet another piece of evidence for observing the plan.

The diagnostic aspect of the monitor includes recommendations and conclusions about the status of the plan. For example, each thruster firing is expected to cause a certain decrement in the reaction wheel speed. Since we know the thruster efficiency and the planned number of firings, the monitor can provide an estimate of the status change in wheel speed.

6 Conclusions and Future Work

We have outlined the current status of our system to assist momentum unload operations. It includes a rule-based component which encodes the attitude control specialist knowledge about momentum unloading and a blackboard-based component for monitoring the progress of momentum unload operations. Refinement and integration of the system into WSGT will occur over time.

We expect to evaluate further features of attitude control operations and state-of-health monitoring for inclusion into the system. Furthermore, we will be expanding the coverage of the system to other areas such as power and thermal monitoring.

References

Erikson, C.-L. and P. Hooker (1989). Tracking & Data Relay Satellite Fault Isolation & Correction Using PACES: Power & Attitude Control Expert System. In *Proceedings of the 1989 Goddard Conference on Space Applications of Artificial Intelligence*, pp. 161–168. NASA.

Forgy, C. L. (1982). Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence 19*(1), 17–37.

Group, T. D. S. (1985). *TDRSS Spacecraft Operations - TMO253*, Volume 1. One Space Park, Redondo Beach, CA: TRW.

Howlin, K., J. Weissert, and K. Krantz (1988). MOORE: A Prototype Expert System for Diagnosing Spacecraft Problems. In *Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence*. NASA.

Jackson, R., M. Johnston, G. Miller, K. Lindenmayer, P. Monger, S. Vick, R. Lerner, and J. Richon (1988). The Proposal Entry Processor: Telescience Applications for Hubble Space Telescope Operations. In J. Rash and P. Hughes (Eds.), *Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence, NASA Publication 3009*, pp. 107–124. NASA.

Keller, R. M., C. Baudin, Y. Iwasaki, P. Nayak, and K. Tanaka (1989). Deriving Shallow Rules from Underlying Device Models. In E. Scarl (Ed.), *1989 Workshop on Model Based Reasoning*, pp. 124–128. American Association for Artificial Intelligence.

Lawson, D. L. and M. L. James (1989). SHARP: A Multi-Mission AI System for Spacecraft Telemetry Monitoring and Diagnosis. In *Proceedings of the 1989 Goddard Conference on Space Applications of Artificial Intelligence*, pp. 185–200. NASA.

Lowe, D., B. Quillin, N. Matteson, B. Wilkinson, and S. Miksell (1987). FIESTA: An Operational Decision Aid for Space Network Fault Isolation. In *Proceedings of the 1987 Goddard Conference on Space Applications of Artificial Intelligence*. NASA.

Marca, D. A. and C. L. McGowan (1988). *SADT Structured Analysis and Design Technique*. McGraw-Hill.

Miller, G. (1989). Artificial Intelligence Applications for Hubble Space Telescope Operations. In A. Heck and F. Murtagh (Eds.), *Knowledge-Based Systems in Astronomy*, Volume 329 of *Lecture Notes in Physics*, Chapter 2, pp. 5–31. Berlin: Springer–Verlag.

Muratore, J. F., T. A. Heindel, T. B. Murphy, A. N. Rasmussen, and R. Z. McFarland (1990). Real-Time Data Acquisition at Mission Control. *Communications of the ACM 33*(12), 19–31.

Nii, H. P. (1986, Summer). The Blackboard Model of Problem Solving and the Evolution of Blackboard of Blackboard Architectures. *The AI Magazine 7*(3), 38–53.

Rook, F. W. and J. B. Odubiyi (1986). An Expert System for Satellite Orbit Control (ESSOC). In K. N. Karna, K. Parsaye, and B. G. Silverman (Eds.), *Expert Systems in Government Symposium*, pp. 221–227. The Computer Society of IEEE.

Systems, C. F. (1984, November). *TDRSS Operations Handbook* (Revision A ed.), Volume V. Las Cruces, NM: Contel Federal Systems. WS-0C-0500.

Tang, Y. K. and C. R. Wetzel (1990). NASA Ground Terminal Communication Equipment Automated Fault Isolation Expert Systems. In *Proceedings of the 1990 Goddard Conference on Space Applications of Artificial Intelligence*, pp. 139–145. NASA.