

Autonomous Power System Intelligent Diagnosis and Control

Mark J. Ringer
Todd M. Quinn
Anthony Merolla
Sverdrup Technology Inc.
NASA Lewis Research Center Group
Brook Park, Ohio 44142

Abstract

The Autonomous Power System (APS) project at NASA Lewis Research Center is designed to demonstrate the abilities of integrated intelligent diagnosis, control and scheduling techniques to space power distribution hardware. Knowledge-based software provides a robust method of control for highly complex space-based power systems that conventional methods do not allow. The project consists of three elements: the Autonomous Power Expert System (APEX) for fault diagnosis and control, the Autonomous Intelligent Power Scheduler (AIPS) to determine system configuration, and power hardware (Brassboard) to simulate a space based power system.

The Autonomous Power Expert (APEX) is a software system that emulates human expert reasoning processes to detect, isolate, and reconfigure in the case of a power system distribution fault. The APEX system continuously monitors the operating status of the Brassboard and reports any anomaly (either static or incipient) as a fault condition. APEX functions as a diagnostic advisor aiding the user in isolating the probable cause of the fault. Upon isolating the probable cause, APEX automatically reconfigures the Brassboard based upon internal knowledge as well as information from the scheduler. APEX provides a natural language justification of its reasoning processes and a multi-level graphical display to depict the status of the Brassboard.

The Autonomous Intelligent Power Scheduler (AIPS) is an intelligent scheduler used to control the efficient operation of the Brassboard. A database is kept of the power demand of each load on the Brassboard and its specified duration and priority. AIPS uses a set of heuristic rules in order to assign start and end times to each load based on priorities as well as temporal and resource constraints. When a fault condition occurs AIPS assists APEX in reconfiguring the system.

The APS Brassboard is a prototype of a space-based power distribution system and includes a set of smart switchgear, power supplies and loads. Faults can be introduced into the Brassboard and, in turn, be diagnosed and corrected by APEX and AIPS. The Brassboard also serves as a learning tool for continuously adding knowledge to the APEX knowledge base.

This paper describes the operation of the Autonomous Power System as a whole and characterizes the responsibilities of the three elements: APEX, AIPS and Brassboard. A discussion of the methodologies used in each element is provided. Future plans are discussed for the growth of the Autonomous Power System.

Introduction

Our future presence in space will require larger and more sophisticated working and living environments. Such environments

will consist of numerous integrated subsystems that will have to be maintained with a high degree of reliability. The electrical power system on a platform such as the Space Station Freedom, Lunar base, or Mars base will be one such subsystem. The APS project explores intelligent hardware and software architectures for safe and efficient system operation.

On large space platforms, the small number of humans on-board will be overloaded with science and other activities. Normal operational concerns of the power system including the control of a large number of switching devices, routine maintenance checks, and scheduling of loads would overwhelm the crew of any such mission.

Ground based experts are one possibility for control. The operators on the ground would be responsible for the minute-by-minute operation of the power system. These experts will be expensive, and the lifecycle cost of any such system would surely suffer in the long run, if too much work rested on the shoulders of these experts. Also, these experts would have a long problem solution time, if they are immediately available at all. This problem is compounded by communication delays, if the power system is more distant than low earth orbit [Dolce].

Since the control of such a large system is difficult to accomplish by the use of on-board crew or ground based systems, the importance of an automated on-board system is evident. The more responsibilities that an automated system assumes will increase the speed of critical decisions. Consequently, this will improve reliability and safety, and drastically lower the life-cycle costs of the power system. The expertise needed to automate these systems could be contained within a set of expert systems. For this reason, the use of expert systems for control of such a power system will significantly decrease operating costs, increase efficiency, and provide for safer operation.

The proposed distributed design of the power system will be much different than bus-based power systems of earlier spacecraft. The operation of a large distributed space power system will be more efficient and safer than its bus-based counterpart, although its control will be more complex.

Hardware design will be a major element in keeping the system operating as safely as possible. Safing of the power distribution system in the case of a "hard" or catastrophic fault will be accomplished by embedded controllers and "smart" switchgear. To preserve the efficient operation of the power distribution system, however, the fault must be isolated, and appropriate recovery procedures must be performed. This is the job of the fault diagnosis expert system. Potential power disruptions can also be avoided by detecting incipient fault conditions that are not immediately threatening to the power distribution system, but over a period of time will become a fault. Soft faults, which can cause graceful degradations of a system, are also best detected by expert systems.

Intelligent scheduling systems will be an integral part of minute-by-minute operations. The scheduler must know the state of the system at all times and be able to help reconfigure in the case of a fault. The most important responsibility of the scheduler is the efficient operation of the system which is accomplished by assigning as many of the available resources as possible to the proper activities.

The object of the APS project is to demonstrate the use of intelligent control, diagnosis and scheduling technologies to a prototype space power system. Many of the obstacles encountered in the development of software are attributed to the integration of multiple software products, and further integration of these products with hardware. These obstacles as well as aspects of communication, cooperation, and protocol will be addressed. This paper describes many of the potential design and development concerns of an automated space-based electrical power system.

Brassboard

Power System Design

In a distributed power system such as a terrestrial power utility, there are multiple levels of control, including power generation, long range high-voltage transmission, electrical substations, down to commercial and residential load centers. These sub-systems, collectively known as the power grid have controllers ranging from computers operating at regional control centers to humans turning on wall switches. The higher level sub-systems have built-in redundancy to keep the system operating in the event of a local fault. Power can be used at will, as spinning reserve generators are available to generate more power when needed. The complexity of control of such a system is proportional to the relatively small number of power generating facilities. The power utility, in most cases, does not have to worry about which users are demanding power from the grid, only a measure of loading on the grid as a whole.

In a conventional bus-based satellite power system, the operating scheme is much

different. The power system is designed so that it has just enough power to run all the power consuming loads at the peak operating condition. All of this power is supplied to a local bus to which all loads are connected. There may be some redundancy in the form of multiple busses, but the architecture is usually quite simple. In general, there is no need to schedule loads, since there is sufficient power to meet load demand. This architecture does not provide for a robust and reliable control environment when designing a large power system.

The proposed distributed power system for space applications is much like that of a terrestrial power utility with regard to power generation, transmission, and distribution. Unlike a terrestrial system where power is supplied to a load on demand, all required power must be determined in advance by the scheduler. In the space system, the total amount of resources is fixed, and the loads must be controlled. Since the number of loads is much larger than the number of sources, the complexity of this control scheme is much higher. This complexity is offset by the added flexibility, reliability, and safety offered over a

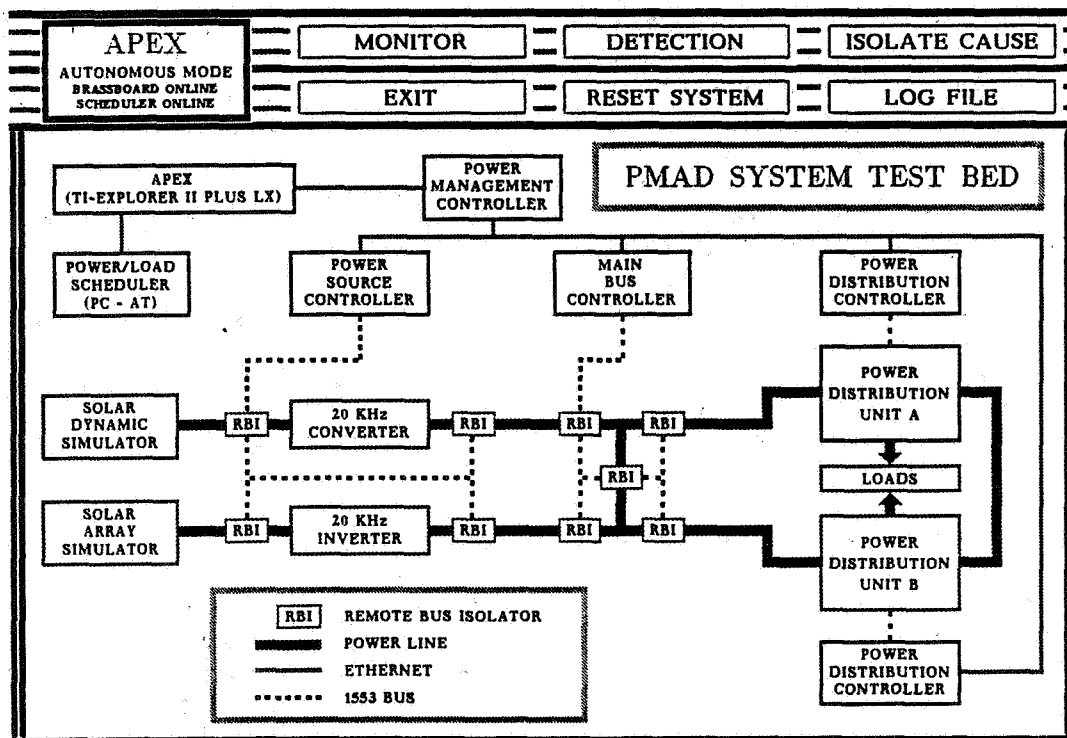


Figure 1. Power System Distribution Architecture

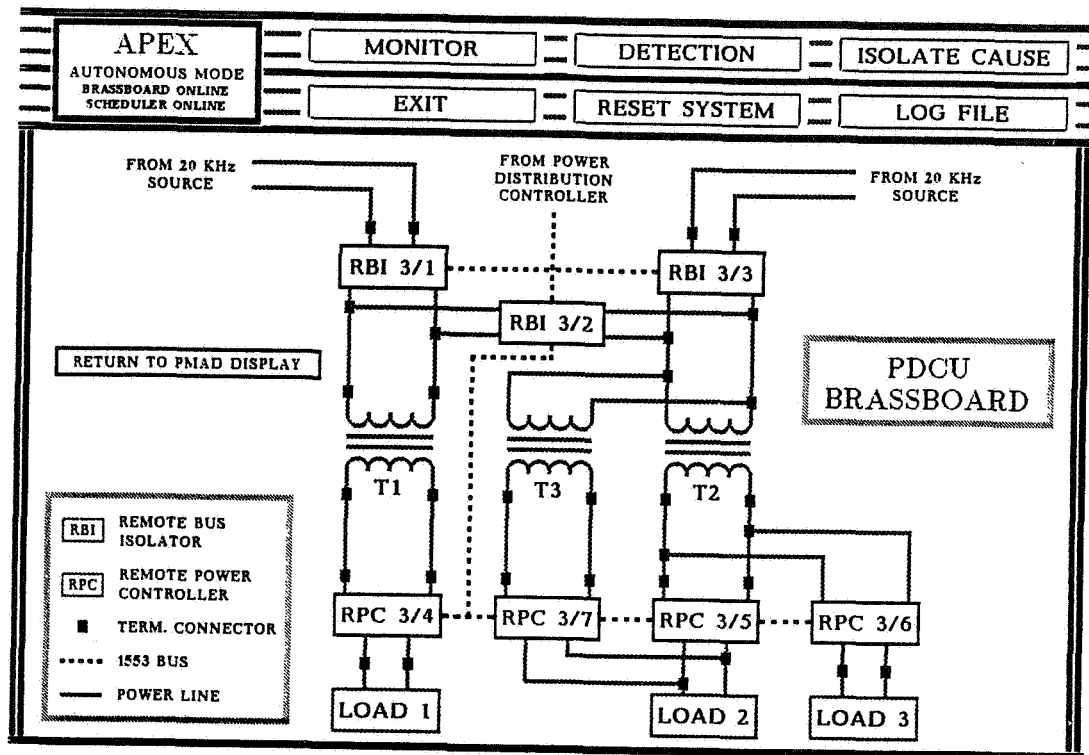


Figure 2. APS Brassboard Configuration.

conventional bus-based system.

The power system switchgear are designed to safe the system in the case of a hard fault as quickly as possible. This is done in a time frame much shorter than possible by any type of software. After the system is safed, a software system can then diagnose and reconfigure the system to return it to a safe and efficient operating condition. Another important design feature in a terrestrial utility as well as the proposed space-based design is the concept of coordination. Coordination prevents faults from propagating up the system. This means that if a fault occurs at some level of the distribution structure, it should be isolated at the next higher level. Without such a design, any fault could disable the entire system. Containment of these faults is accomplished by designing the lower level switches to trip faster and at lower current levels than the higher level switches. This safes the system and keeps a local fault from affecting a large portion of the power distribution structure.

The space power system described in this paper is based on such a distributed

system architecture. This architecture is based on a previous Space Station Freedom design and the architecture is the important aspect of the system [Beach]. The operating frequency does not affect the function of the distributed control scheme, it will function with either an AC or DC system. The entire power system architecture is shown in Figure 1. The APS Brassboard is shown in Figure 2. Figure 3 shows a switch level diagram of an RBI.

Hardware

Each piece of switchgear, contains a set of sensors, analog-to-digital electronics, 1553 bus interface, and a set of power supplies. Two types of switches are used on the Brassboard, Remote Power Controllers, RPCs, and Remote Bus Isolators, RBIs. An RPC contains only a solid state switch. An RBI contains both solid state and mechanical switches which allow for a higher power capacity and increased operating efficiency. The RPC switches on the order of 30 micro-seconds. In an RBI, the solid state switch turns on first, then the mechanical switch engages to lessen the effects of contact

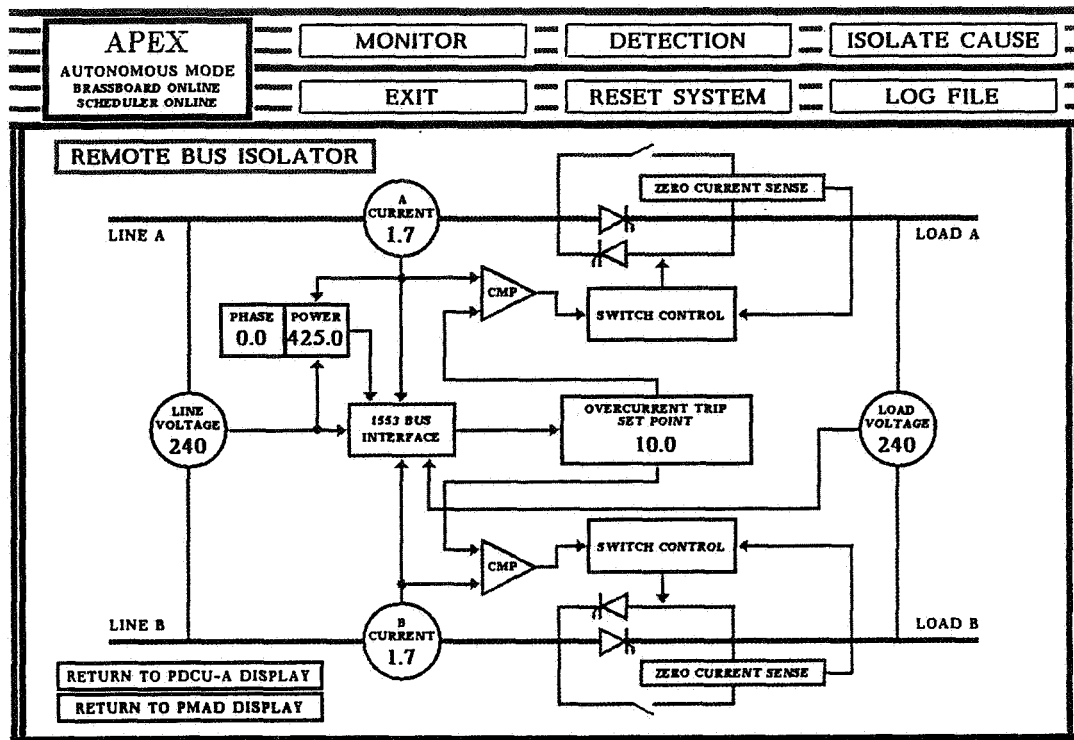


Figure 3. Switch Diagram of an RBI.

depletion and arcing, and to increase the efficiency at high power levels. This takes place within 15 milli-seconds. A set of 20 kHz transformers is also used to step down the voltage between the RBIs and RPCs.

The signal sensing and limiting circuit is the heart of the switches' "smarts". Voltage, current, power, and phase angle are monitored both at the line (power input) and load (power output) side of the switch. This information is then digitized using an 8-bit analog-to-digital signal converter. A circuit is included which stores the over-current set point which is determined by the controlling software. The circuit continuously monitors this limit, and if a problem is encountered, the switch is tripped. Since the switch has both line and load sensors, after a switch has tripped, data can still be taken to see if it is again safe to turn the switch on, or to find out exactly what the problem is.

The Brassboard has two 20 kHz power sources, a 1.5 kW power amplifier with an external oscillator and regulator circuit, and a 12 kW Mapham inverter. Loads on the brassboard are simple incandescent or infrared light bulbs. These loads provide a good

visual indication of what switches are turned on. When current or voltage problems exist, a dimming of the lights is produced.

Brassboard Communications and Software

The brassboard switchgear is controlled by a set of Intel 8086 based single board computers. Each computer has a built in 1553B, IEEE 802.3, and RS-232 communication capability, which allows them to communicate with the switchgear, each other, and outside computers respectively. The Power Management Controller PMC and Power Distribution Controller PDC represent a hierarchy of controllers responsible for the Brassboard's operation.

The control algorithms within the PMC and PDC are implemented using embedded Ada language software [Wright]. APEX sends a request for data to the PMC over the RS-232 link. The PMC contains continuously updates values of voltage, current, power, phase, as well as 12 bits of status information for each piece of switchgear.

Faults

The Brassboard normally operates

with the no faults existing in the system. To develop fault detection, recovery, and reconfiguration schemes, actual hardware faults must be introduced into the system. The APEX fault diagnostic system is capable of detecting hard, soft, and incipient faults.

A hard fault will cause a switch to trip, thus generating an abrupt change in system configuration. This type of fault can be caused by a multitude of possible problems including switch failures, short circuits across power lines, and some types of load failures. These can be simulated by switches in the system to cross the power lines, etc.

A soft fault is a type of fault which does not cause a switching device to trip. It can be manifested as a small leakage path in the wiring between switches, problems within the load itself, leakage paths within the switch itself, or problems with parts of the mechanical switch. This can be simulated by introducing various resistive circuits into the brassboard.

Incipient faults are faults that can be detected before they actually become a problem. Such faults could be small changes in a load power demand, a small degradation in the insulation of wiring, or small resistive faults building up within the switchgear. The incipient faults can be simulated by the introduction of a programmable load to portions of the Brassboard.

Reconfiguration

The Brassboard architecture, although simple by comparison, incorporates many of the design concepts found in a much larger system. The Brassboard includes two power sources to which any load can be attached. If one source needs to be disconnected from the system or fails, the remaining loads can be reconfigured and attached to an alternate power source. In this way, a level of redundancy exists for dealing with source faults. One of the loads on the Brassboard has a redundant path attached to it. This is meant to simulate a fault of one piece of switchgear, and loss of power to the load. This fault can be cleared by switching the

load over to the redundant path.

APEX

Introduction to Operation

The APEX system is designed to take the place of a human expert in the control and diagnosis areas that require intricate thought patterns, repetitious operations, rapid response time, or where human experts are not available. The Autonomous Power Expert (APEX) system has been developed to emulate human expert reasoning processes used in the diagnoses of fault conditions in the domain of space power distribution. The knowledge that exists in the minds of the power system experts and their thought processes represent a sound method to diagnose faults in such a power system. In order to capture this method, isolation, diagnosis, and reconfiguration knowledge is represented in APEX as a set of rules and data concerning the operation and configuration of the power system. APEX is connected to the Brassboard and the scheduler in order to implement an entire power system control architecture.

APEX receives data from the PMC and then initiates fault detection. APEX detects faults by comparing expected values obtained from AIPS to the measured operating values obtained from the Brassboard. If no deviations from the expected operating state of the Brassboard are found, APEX will again request data from the PMC and re-initiate the fault detection activity with the new data.

Upon detection of a fault condition, APEX accesses information and rules contained in its knowledge base, reaches a conclusion, and displays the probable cause for the detected fault to the user. Actions are then taken to correct the fault and return the Brassboard to a safe and efficient operating condition. The user can also ask APEX to justify its conclusion. APEX can operate as either an advisor or an autonomous controller of the APS Brassboard.

Implementation Overview

APEX is currently implemented on a Texas Instruments Explorer II workstation in LISP and employs the Knowledge Engineering Environment (KEE) expert system shell. APEX consists of a knowledge base, a database, an inference engine, and various support and interface software. The knowledge base comprises facts and rules that correspond to knowledge acquired from the human expert during problem solving. The database is the basic working area where storage and calculations of sensory data for incipient fault detection occurs. The inference engine is the reasoning mechanism that draws conclusions from information stored within the knowledge base. In choosing the appropriate recovery procedures for the isolated fault, APEX also relies on the reasoning capabilities of the inference engine. Conventional software provides the user with an interactive interface to communicate with APEX and to obtain data from the Brassboard and AIPS.

User Interface and Operation

The goal of the user interface is to provide access to APEX which requires a minimal amount of training. Communication between APEX and the user is accomplished by easy-to-use mouse-selectable menus, color graphics and text displays. The user interface screen presents a color display that is divided into three areas as shown in Figure 1. The top portion of the screen is the control menu that allows the user to select the desired APEX function. When a function is selected, mouse-selectable options for that function appear in the options menu located in the lower portion of the screen. Located on the left side of the control menu is the APEX mode/interface menu. Fault detection and fault isolation results are shown within the main display area by means of color diagrams and text explanations.

The graphical displays in the main display area consist of a set of hierarchical diagrams that represent three different levels of the power distribution system. The diagram in the main display area shown in Figure 1 represents the overall power

distribution system. When an active fault is detected, the area of detection is outlined in red. For an incipient fault condition, the area is outlined in yellow. The yellow indicates that a parametric value is probably going to go out of tolerance if preventive action is not taken. The user can get a more detailed diagram of an area by choosing the particular area of interest and clicking the mouse. Figure 2 shows the Brassboard top level diagram. Figure 3 shows the switch level diagram which is one level below the Brassboard diagram. Each switch level diagram displays the actual measured data values enabling the user to see which parametric attribute is out of tolerance.

The control menu in Figure 1 contains the following six mouse-selectable functions: *monitor*, *detection*, *isolate cause*, *reset system*, *log file*, and *exit*. The *monitor* selection causes APEX to continuously acquire and check parametric values from the power distribution system. When either an active or incipient fault is detected, APEX displays a "fault detected" message in the upper left corner of the user interface screen. Once alerted, the user can display the fault detection analysis by selecting *detection* in the control menu. When *isolate cause* is selected from the menu, APEX will access the fault isolation rules to determine the probable cause of the detected fault.

Recall that when a function is selected, the options menu provides the user with available options for that function. For example, when the user selects *isolate cause* from the control menu, APEX will display the probable cause of a detected fault and the options menu will contain *continue*, *why?*, *recommend*. This is shown in Figure 4. If the user selects *why?*, APEX will display the reasoning process leading to the probable cause conclusion as shown in Figure 5. The *recommend* option allows the user to request recommended action procedures for correcting the fault.

The mode/interface menu provides controls for selecting the operational mode of APEX as well as changing the online/offline status of the data acquisition and scheduler interfaces. APEX can operate in manual

Incipient Fault Detected	MONITOR	DETECTION	ISOLATE CAUSE
	EXIT	RESET SYSTEM	LOG FILE

Fault Isolation Analysis

--- Fault #1 of 1 ---

The probable cause for the problem detected at RBI.3/3 is:

1. A leakage path from the high to low side is increasing. The path is within the transmission line between the RBI.3/3 load side and the transformer primary.

Click the mouse on
CONTINUE below to
close this display.

CONTINUE	WHY?	RECOMMEND
----------	------	-----------

Figure 4. Isolation.

Incipient Fault Detected	MONITOR	DETECTION	ISOLATE CAUSE
	EXIT	RESET SYSTEM	LOG FILE

A leakage path from the high to low side is increasing. The path is within the transmission line between the RBI.3/3 load side and the transformer primary.

JUSTIFICATION

1. RBI.3/3 is a Remote Bus Isolator.
2. The A current of RBI.3/3 is increasing.
3. A and B currents for RBI.3/3 are equal.
4. The B current of RBI.3/3 is increasing.
5. All power for RPCs connected to RBI.3/3 have no increasing incipient fault characteristics.

Figure 5. Justification.

mode where the user selects appropriate commands from the control menu. In autonomous mode, APEX will monitor the power distribution system, detect faults, isolate the probable cause and provide appropriate fault recovery automatically without input from the user. The user can select whether APEX is to acquire data from the Brassboard or the Brassboard simulator. In scheduler online mode, APEX can request and receive live scheduling information from AIPS. When the scheduler interface is offline, APEX reads pre-saved scheduling information.

Brassboard Simulation, Display and Control

Part of the user interface, is the Brassboard data simulator/display/control interface. In simulation mode the main display area contains a diagram of the simulated Brassboard which can be set by the user. Along with simulated switching device data, the simulated data for each load on the brassboard is also displayed. With the use of the Brassboard simulator, the fault diagnosis capabilities of APEX can be used without the Brassboard. In display mode, actual sensor data from the brassboard can be displayed and recorded for the user's observation as shown in Figure 6. This function is used as an interface for our experts and operators to verify the reasoning of APEX. In control mode the user has the capability of issuing commands to the Brassboard in order to turn switching devices on/off and set trip limits.

Rule Base Operation

Representation of knowledge within the APEX knowledge base consists mainly of frames, semantic triples and production rules. Frames are structures which describe objects or classes of objects and the relationships between them. Objects are composed of slots which specify the different attributes belonging to each object. Individual slots of an object can contain either declarative or procedural information. Declarative information expresses facts about the object, whereas procedural information is in the form of a program or a set of procedural steps. Frames themselves are considered to be declarative information. Within APEX,

declarative information is also represented by semantic triples which state information in the form of object/attribute/value (i.e. attribute of object = value). Production rules are 'If-Then' statements which infer either declarative or procedural facts when the conditions contained in the premises of the rule are found to be true. Again the facts are represented as a semantic triple (declarative) or a program (procedural).

The inference engine is the heart of the expert system, determining how knowledge is represented and processed. By operating on rules within the knowledge base, the inference engine can reason and make inferences about the state of the power system. The ways the inference engine processes the rules and data are commonly referred to as forward and backward chaining. Forward chaining (also known as data driven) works from the given data to a conclusion. Backward chaining (also known as goal driven) works from a particular goal and tries to either confirm or refute its truth. In the APEX system, fault detection is implemented with forward chaining and fault isolation is accomplished with backward chaining [Ringer].

Fault Detection

When a fault occurs in the power distribution system it appears as a deviation on the expected values of the currents flowing through the switching devices. Therefore, faults are detected by comparing the measured operating current values to the expected current values calculated by APEX and AIPS. A fault is detected when an operational value and the corresponding expected value are in disagreement within a given tolerance. With this mode of fault detection, the set of detection rules can be kept small, greatly reducing the time for fault monitoring.

Fault Isolation

The primary function of fault isolation is probable cause determination for a given fault condition. APEX uses the knowledge contained within the fault isolation rules and the backward chaining capabilities of the KEE inference engine to determine the most

probable cause. Figure 4 shows a display of fault isolation analysis. In this case, there are three possibilities listed as the probable cause. Based upon the present knowledge in the knowledge base and the sensor data obtained from the power distribution system, the exact cause cannot be further isolated without more information.

Figure 5 shows a typical display of probable cause justification. At the top of the main display area the probable cause, which is the backward chaining goal, is displayed. Below the stated probable cause are the premises which support the truth of the probable cause statement. The unhighlighted numbers (1 - 4) are primitive statements of fact contained within the knowledge base. Numbers that are highlighted represent statements of facts that were inferred as subgoals. The user can see the premises used to prove the truth of each subgoal.

Fault Recovery

After the probable cause of a detected fault or incipient fault condition has been isolated, APEX will analyze available information about the current operating conditions and take appropriate actions. These actions pertain to both short and long-term recovery. Short-term recovery determines if the power distribution system can be reconfigured, or if load shedding is necessary. For long term recovery, the repair procedures needed to correct the fault are determined after short term actions have been implemented.

Short-term recovery analysis is based on a set of "recommended action" rules for the particular fault condition. Information about available power sources, current configuration of the power distribution system, the scheduled run times of the loads, and the effects of the fault on the system are all considered during the analysis. If the fault is seriously affecting the amount of power reaching a particular load and an alternate path for power distribution exists, then the system can be reconfigured automatically, to allow the load to run to completion. When the fault cannot be tolerated and alternate power distribution

paths are unavailable, the schedule for the loads is revised by the scheduler possibly resulting in load shedding.

After short-term recovery, the cause of the fault in the power distribution system needs to be repaired. The appropriate procedures needed to repair the problem are determined by long term recovery, based on a set of recommended action rules. In some cases, the cause of the fault is traceable to a group of possibilities, and additional troubleshooting procedures are displayed to intelligently guide the user to further isolate the exact location and to make repairs.

Incipient Fault Detection

If, during conventional fault detection, the rules have been exhausted with no faults detected, APEX checks for incipient faults. Incipient detection is based on statistical linear regression and correlation analysis of the historical data. Measured and expected parametric values of the power system are saved in a historical database. The expert system analyzes the historical data looking for any indication of a parametric attribute that has maintained either an upward or downward trend in the data values over a period of time.

Since the power system is dynamic and the measured value fluctuates over a period of time during normal operation, a parametric ratio of the measured-to-expected value is used to identify any increasing or a decreasing trends in the parametric data. Once the data has been stored in the database, correlation coefficients are calculated for each parametric attribute of all switching devices. A high correlation coefficient indicates that an incipient fault exists in the system.

Once an incipient fault condition has been detected, the user can view the results of the statistical analysis. The results are shown graphically to the user showing trends in the ratio between measured values and expected values. The output screen also shows in which switching device the trend is located. Along with the plot of the linear regression results, the

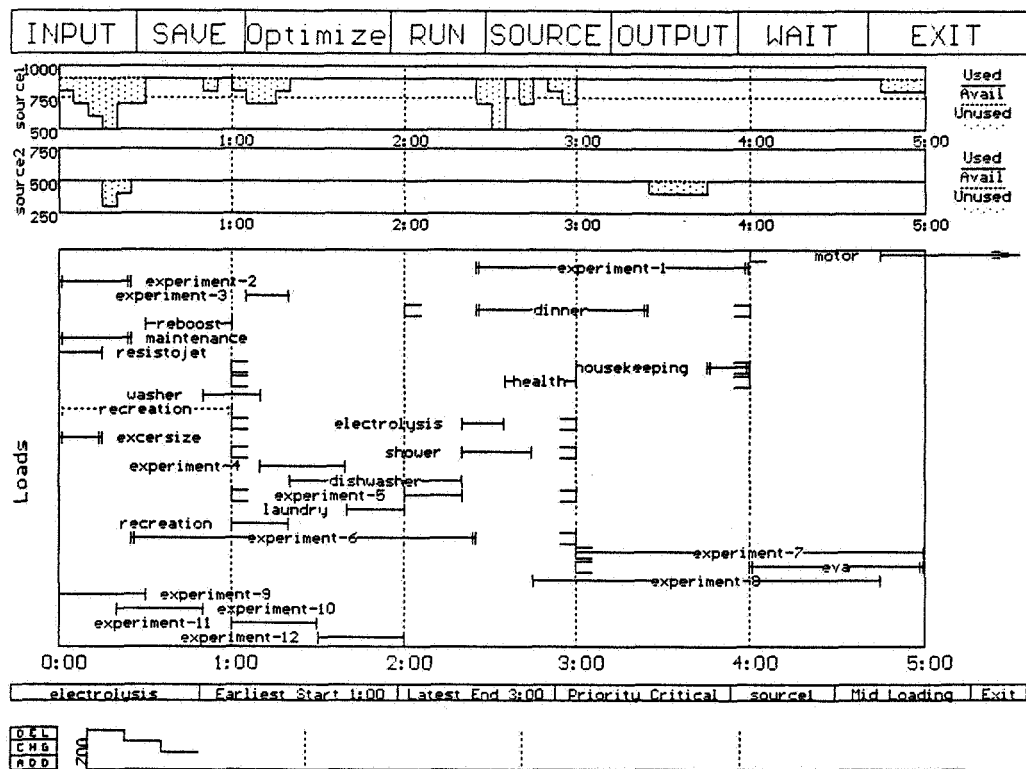


Figure 7. Scheduler Output.

activities must be developed in a day or less. If a schedule takes longer than this definition of real-time, the scheduler will always be backlogged. For this reason, other methods of schedule generation must be employed, that will then give a solution to the problem within our definition of real-time.

Many scheduling methods, referred to as engines, capable of producing acceptable solutions in a reasonable amount of time exist. Constraint propagation, simulated annealing, and various types of heuristic based scheduling approaches all produce good results. AIPS uses a set of heuristics to construct a schedule. This method takes a set of loads and, based on a set of rules, places each of these loads on the schedule. This schedule can be generated very quickly with one depth-first path through the search space.

This heuristic engine can produce a good schedule very quickly, in fact, sometimes even faster than needed. In order to take advantage of this extra time, the one-pass engine can be augmented by an optimization engine. This optimization engine can be employed until the time that the schedule must be implemented. It is also

important that the scheduler be able to stop at anytime during the optimization process and produce a feasible schedule. This is considered an anytime engine. A scheduling engine is considered "anytime" if at anytime during the solution, it can be stopped and a feasible schedule exists. This concept is important if the scheduler needs to operate in conjunction with other software and hardware that need the information as quickly as possible. The AIPS scheduler is always considered "anytime" since the scheduling engine does not allow constraint violations to exist at any point during the scheduling process [Zweben].

Implementation

The Autonomous Intelligent Power Scheduler (AIPS) is designed to schedule a subset of the conditions that will exist on a space-based system. This will demonstrate the feasibility of integrating intelligent software systems for the control of a distributed space power system. AIPS determines the Brassboard configuration and assists APEX in reconfiguration when faults are detected. AIPS is completely autonomous needing no user input to completely schedule

any set of activities received from APEX or when interactively operated. AIPS is written in the C language on a PC platform and is connected to APEX via an ethernet link.

In order to adequately model the interaction between APEX and AIPS, a set of protocols was developed to communicate different scheduling and rescheduling procedures. Protocols were developed to generate an initial schedule and modify existing schedules. The initial schedule generation takes a set of sources and loads and generates a schedule for APEX to follow. Five modification protocols exist: activity change, resource change, activity add, activity delete, and resource delete. During the execution of a schedule, the priority of an activity may change, the power demand of a load may change, the load may need to be dropped from the schedule, or a load may need to be added. Resources also may be changed during the execution of a schedule, or deleted altogether. These protocols enable APEX and AIPS to communicate system configuration information as well as reconfigure the system in the case of a fault.

Internal Structure

Since the problem of scheduling the vast array of loads is so great, one easy way of reducing the complexity of the problem is to break the problem up into smaller sub-problems. In fact this is the only way to solve such a problem. Even if all of the loads that needed to be scheduled were known ahead of time, which they aren't, it would be an impossible task. The problem is currently broken up into a set of planning horizons. This necessitates the carry-over of information from one planning horizon to the next because some loads will carry over from the previous horizon.

Many attributes of constraints on activities exist in a scheduling environment. Some of these attributes have been used in the development of the AIPS scheduler [Britt]. These attributes are stored as structured data objects for each load. A time varying profile of power requested by the load is stored as well as the length. The earliest start time and latest end time can be

specified for loads that wish to start and finish in a certain time window. The scheduling engine also take into account the priority of a load. Priorities are critical, normal, and low. A set of functions also exists to allow for loads to continue in from the last planning horizon, this is done by using the priority immediate. Each load is attached to a certain power source. It is not feasible that the activity can be assigned to more than one source given the architecture of the system. The last attribute specified for each load is a loading preference, declaring whether the would rather be scheduled as early as possible, as late as possible, or at the middle of its available start time window.

Evaluating the "goodness" of a schedule can be an ambiguous task. It is important to both schedule the high priority loads, as well as use as much available power as possible. What if it is possible to delete a higher priority load and replace it with a lower priority load which makes for an overall higher power use? Is this a better schedule? A balance between these and many other rating schemes must be decided. The AIPS scheduler takes into account many of these elements but, coming from a power system perspective, the AIPS scheduler gives preference to a schedule that uses more power than one that schedules more loads.

Scheduling Engine

The main scheduling engine of AIPS is a heuristic one-pass engine. The information used to position each activity includes projected resource demand and total resource use, as well as each activities power demand, temporal constraints, requested loading preferences, and priority. Using this method, a schedule is generated in one-pass using information of how a human would build a "good" schedule. This may not be the mathematical optimum but this schedule can be generated in a very short time.

Since this is a one-pass engine it must be decided in which order to place the loads onto the schedule. This decision influences the final outcome of the schedule. In a one-pass engine, an activity has a much better choice of available positions, as well as a

better chance of being scheduled if it is put on the schedule before most other loads have been put on. It is also advantageous to place larger higher power consuming loads on the schedule as early (in the scheduling process) as possible, because they may not fit if they are put on later, when much of the available power has already been subscribed. Also, if a load is very constrained in the length of its possible start times, it should be placed on the schedule earlier. This ranking is done before any load are actually placed on the schedule.

Since the amount of power demanded differs over time (temporal constraints of loads causes this), a projection of the loading on each power source at each time period can be made. This projection can then point out projected bottleneck areas for each source. The important point is that this is a projected demand, no loads have actually been scheduled yet [Biefeld]. Next, each load is placed on the schedule.

Each feasible start time where a load can be scheduled is determined and evaluated based on the previously stated factors. The projected demand here is used to influence the decision of where to place the load. Other factors are also used in the temporal placement of the load. A front loaded schedule will usually make for more resource usage (because it tends to pack the schedule more tightly), but this must be balanced with the conflicts of projected bottleneck regions, as well as each load's preference for a certain position on the timeline (front, mid, end loading). Based on these criteria, a start time is decided upon. This process is repeated until all loads have been attempted to be placed on the schedule.

Explanation of Output

The scheduler also has an interactive graphical user interface that can be used to test the scheduler as shown in Figure 7. This interface is fully mouse controlled and allows the user to edit activity information, resource information, as well as making changes to the schedule after the scheduling engine has given its solution. The user may test to see if they can manipulate the schedule in such a way to

build a "better" schedule than the scheduling engine itself. This user interface also makes it easy to see how the scheduling engine works and makes it easier to test the scheduler.

The upper two graphs show the two sources of electrical power, with the power on the y-axis and time on the x-axis. Available power is shown as a dotted line, while scheduled power is shown as a solid line. This difference between available power and scheduled power is the unused (unscheduled) power which is shown as the dotted fill area between the available and scheduled power. This display can also show the amount of power that is oversubscribed. The user can introduce oversubscriptions in the schedule but this makes the schedule invalid.

The gantt chart in the middle of the screen shows each load that was scheduled. The length of the load corresponds to the length of the load and the scale is shown on the x-axis of the gantt chart. The earliest start and latest end points (if they are specified) are shown by brackets at each side of the load. On the color screen, each load is color coded by its priority. If a load continues into the next planning horizon, this is shown as an arrow at the end of the load.

The edit window at the bottom of the screen shows a more specific description of the load that the mouse is currently pointing to. Values such as power demand, length, start and end constraints, priority, source, and loading preference can be specified within this window.

Future Developments

Rules need to be added to enhance the performance and increase the scope of APEX's knowledge in the areas of long-term recommended actions, autonomous mode operation, and recovery scenarios. Enhancements are needed in the areas of updating the log file, updating the user interface, and allowing time variant load operations. Additional software implementation for testbed control and

schedule display are also needed. The application of model-based reasoning to the system is also being studied to relieve the overhead of a large number of rules and provide deeper reasoning for fault isolation.

The Brassboard currently contains seven pieces of switchgear, and two controllers. Future enhancements will greatly increase the number of switchgear and controllers available. This will allow more complex control and fault scenarios to be tested. Time varying loads will also be added to the Brassboard load structure to increase the complexity and provide APEX and AIPS with a more realistic load set.

The scheduler optimization engine needs to be upgraded, since the current engine is very simple. This will allow for a better schedule to be generated. Also, a more robust interface between APEX and the scheduler is needed.

Summary

With the advent of larger and more complex space environments, more reliable and efficient subsystems must be designed. The APS project represents a power system automation scheme implemented through the use of intelligent software systems. Many of the problems faced in the design of a complete system were encountered in the APS project. The integration of a power system, fault diagnosis software, and scheduling software proved the feasibility of implementing such a control scheme. Of course, more complex hardware and software scenarios still need to be tested to further explore the workings of such a system.

Acknowledgements

The authors would like to thank the Space Station Electrical Systems Division at the NASA Lewis Research Center for their hardware and software support. We would also like to thank Gene Lieberman for his work on the Ada software, and Walt Krawczonek for being our power system "expert". And, Jim Kish for his work in

project management and coordination.

References

- [Beach] R.F. Beach, G.L. Kimnach, T.A. Jett, and L.M. Trash, "Evaluation of Power Control Concepts Using the PMAD Systems Testbed", In Proceedings 24th IECEC, 1989.
- [Biefeld] E. Biefeld, L. Cooper, "Operation Mission Planner: Final Report", JPL Publication 90-16, March 15, 1990.
- [Britt] D.L. Britt, A.L. Geoffroy, J.R. Gohring, "Managing Temporal Relations", Proceedings of the Goddard Conference on Space Applications of Artificial Intelligence, 1990, NASA CP 3068.
- [Dolce] J.L. Dolce, J.A. Kish, and P.A. Mellor, "Automated Electric Power Management and Control for Space Station Freedom", In Proceedings 25th IECEC, AICHE 1990.
- [Ringer] M.J. Ringer and T.M. Quinn, "Autonomous Power Expert System", In Proceedings 25th Intersociety Energy Conversion Engineering Conference IECEC, AICHE 1990.
- [Wright] T. Wright, M. Mackin, and D. Gantose, "Development of Ada Language Control Software for the NASA PMAD Testbed", In Proceedings 24th IECEC, 1989.
- [Zweben] M. Zweben, M. Deale, and M. Eskey, "Anytime Rescheduling", NASA Ames Artificial Intelligence Research Branch Technical Report.