MARSHALL
GRANT
IN-61-CR
296996
P. 32

UAH Report Number 90-30

Intelligent Data Reduction
(IDARE)
NAG8-642
Final Report

Prepared by:

D. Michael Brady
Donnie R. Ford
Kenneth E. Johnson Research Center
University of Alabama in Huntsville
Huntsville, Alabama 35899

Prepared for:

Bryan K. Walls
David J. Weeks
Electrical Power Branch
Information and Electrical Systems Laboratory
George C. Marshall Space Flight Center

March 19, 1990

(NASA-CR-186322) INTELLIGENT DATA REDUCTION (IDARE) Final Report, 1 Jun. 1988 - 31 May 1989 (Alabama Univ.) 32 p CSCL 09B

N91-11393

Unclas
G3/61 0296996

ABSTRACT

This report contains a compilation of the activities and accomplishments of the Johnson Research Center on the Intelligent Data Reduction System. The period of performance documented is from June 1, 1988 to May 31, 1989.

TABLE OF CONTENTS

LIST OF APPENDICES

1 Project Background

1.1 Data Reduction

Data reduction becomes necessary (or at least valuable) whenever we are interested analyzing a data set whose volume and/or complexity is so great that it prohibits (or inhibits) the desired level or speed of analysis. As such, it is any attempt to condense a data set down into a more manageable volume or more understandable form. Reducing the volume of a data set may be accomplished by simply eliminating unnecessary portions or by expressing the pertinent information in one data subset in terms of a smaller subset. Statistical reductions, such as averages, maximums, and standard deviations, are all examples of the latter type of volume reduction. Complexity reduction is, generally speaking, a more difficult task, in that it implies an ability to better convey information for human understanding. This is made difficult by the fact that different people, with different backgrounds, may be predisposed to better understand different forms of expression. A doctor discussing a case with a colleague, for example, is likely to use a completely different vocabulary than he/she would use with the patient. We can, however, make a few general statements about complexity reduction, like that people are better able to understand information expressed in a graphical form than a tabular one.

Data reduction is performed in a variety of contexts: people who write newspaper headlines, financial reports, "Cliff" notes, and even the author of this report are all engaged in data reduction. The performance of all these activities is measured in terms of accuracy and effectiveness. The reduced data is accurate if it captures the meaning of the original data set, and it is effective if it is easier to understand. These two goals are somewhat at odds with one another, however. The challenge is to present the data set in an easy to understand manner while maintaining its complete meaning.

The success of a data reduction effort is also highly dependent on the nature of the data set and the data source. The reduced data can, of course, only describe the data source as well as does the original data set.

So, it is imperative that all important information be included in the data set. Typically, if we have control over the formation of the data set, we err on the side volume, by extracting *everything* from the data source that we think *might* be important. This is often done under the pretense that some volume reduction techniques will be applied before analysis. The nature of the data source can have a more serious impact on the success of the reduction effort. If the data source and/or its analysis are poorly understood or developed, it may be impossible to accurately reduce the associated data. In this case the dichotomy between the accuracy and the effectiveness of the reduction is complicated by the experimental nature of the data source. It is very difficult to concisely describe a data source, if even the structure of that description is unknown.

1.1.1 Traditional Approaches

Various expert systems (ES's) have been developed to monitor physical data sources and summarize their health. Space and strategic systems have been prime candidates for this research in that the typically emit huge quantities of data that must be analyzed and acted upon very quickly. An ES, in this situation, might be responsible for generating notifications in real-time that describe the health of the data source and, perhaps, make recommendations regarding its maintenance. In recent years, many systems have been developed to utilize Artificial Intelligence (AI) techniques to reduce large quantities of data down to a relatively small qualitative description. The most common approach to this problem is to write an ES that can mimic the logical thought processes involved in reducing the data. Symbolic programming languages, such as Lisp and Prolog, are very good at representing and manipulating high-level, abstract entities, so they are used most frequently for this type of problem. Generally speaking, these systems contain production rules which can be applied to either raw or statistically reduced data to generate notifications describing the current state of the data source. In particular, they have proven useful for the detection of anomalies [7] [9], data classification [3], and even for recommending corrective measures based on anomalies and some knowledge of their probable cause [2] [9].

As these systems become more complicated, however, the number and diversity of the notifications generated may become overwhelming. Questions have been raised as to whether enough attention has been paid to the dynamic patterns of usage these systems might enjoy in the field [5]. Expert systems tend to be designed to generate notifications based solely on the fluctuations of the data source, without regard to the user's current interests. In a sense, then, these systems are static in that they can only be used to interpret the data according to one model, the one embodied by the production rules. Any experimentation or customization of this model would typically require the modification of the production rules themselves [1]. This is not generally thought to be good practice since those changes could damage the integrity of the ES for future users. For this reason, most ES's don't provide the end-user with this capability.

1.1.2 Intelligent Data Reduction

There is an apparent need for a method of filtering notifications based on environmental context, i.e. the user's current focus and the relative importance of the notification. In order to provide greater flexibility and more sensitivity to the ever-changing runtime environment of the ES, we have proposed the addition of a user profile that will describe alternate, temporal models of the system. These profiles should be easy to create and modify so as to promote experimentation, but they should be user-specific so that they will not interfere with the overall integrity of the system. We have termed a data reduction system with the addition of a user profile infrastructure as an intelligent data reduction system (IDARE).

On a simplistic level, the user should be allowed to specify which, if any, notifications types he/she would like to have filtered out. The default, however, is that no notifications are suppressed. Using this facility, the user can, for example, filter out any communications notifications if they are unimportant currently. Furthermore, the user should be able to construct more complicated, conditional filters. Conditional filters are, in effect, meta-rules that further reduce the data form what is merely aberrant (i.e., the notifications form the ES) to what is merely aberrant and interestingly so (i.e., the notifications that make it through the user profile filter). It should be stressed, however, that these meta-rules should be thought of as temporal. The user profile should be used to experiment with different models and to customize the system to one's likings, not to make modifications to the ES's rule base.

1.1.2.1 Applicability

A user profile system, as described above, is applicable in situations where some experimentation with the model of the data source encapsulated by the ES's production rules might be of value of where it is known that different users might be interested in different aspects of the data source. If the ES is designed to do a single very specific task, as is the case with many ES's today, there is probably no need for such a facility.

There has been a great deal of interest, also, in autonomous or closed-loop ES's [8] [9]. In these system, the ES is responsible for not only monitoring, but also affection the data source-- the advantage being that corrective action could be initiated within second of detection. Since this implies that the model of the system is very well known and static, a user profile system does not apply. In fact, the role of the ES as an advisor is often thought to be a first step towards autonomy. In this case, the utility of the user profile system will become less as autonomy increases.

1.1.2.2 Generality

The intention of this research has been to provide a generic user profile system that can be used in a variety of applications. It must accept as input, therefore, some of the specifics of the ES in question. Among these are the vocabulary of the ES, which includes the names of all of the physical and conceptual objects in the system, the available measures pertaining to these objects, and the relationships that can exist between them. For example, the vocabulary of the present system includes names like battery, overcharging, and SPA's; measures like cell pressures, and SPA currents; and relationships like $\leq$, increasing, and =. This sort of information is most easily encapsulated using object oriented programming techniques. In this system, there are classes for names, measures, and relationships. The input to the user profile system form the ES, then is a list of instances of these classes.

Abstracting the workings of the user profile system form the specifics of this input, while maintaining the flexibility of the conditional filter interface, is a difficult problem that has not been completely solved at this time. This type of interface is something like a natural language (NL) interface, except that the user must choose his/her phraseology from the selections provided by the computer rather than from his/her imagination. This bounds the problem, which has traditionally been the major obstacle in NL systems [6]. If one wishes to preserve the expressiveness of NL, however, the bound are still quite large.

1.2 The HST Electrical Power System Testbed

The HST Electrical Power System (EPS) testbed located at the Marshall Space Flight Center, has been used as the data source for this system. The schematic for the testbed is shown in Figure 1 as it appears in the system.

ORIGINAL PAGE IS
OF POOR QUALITY

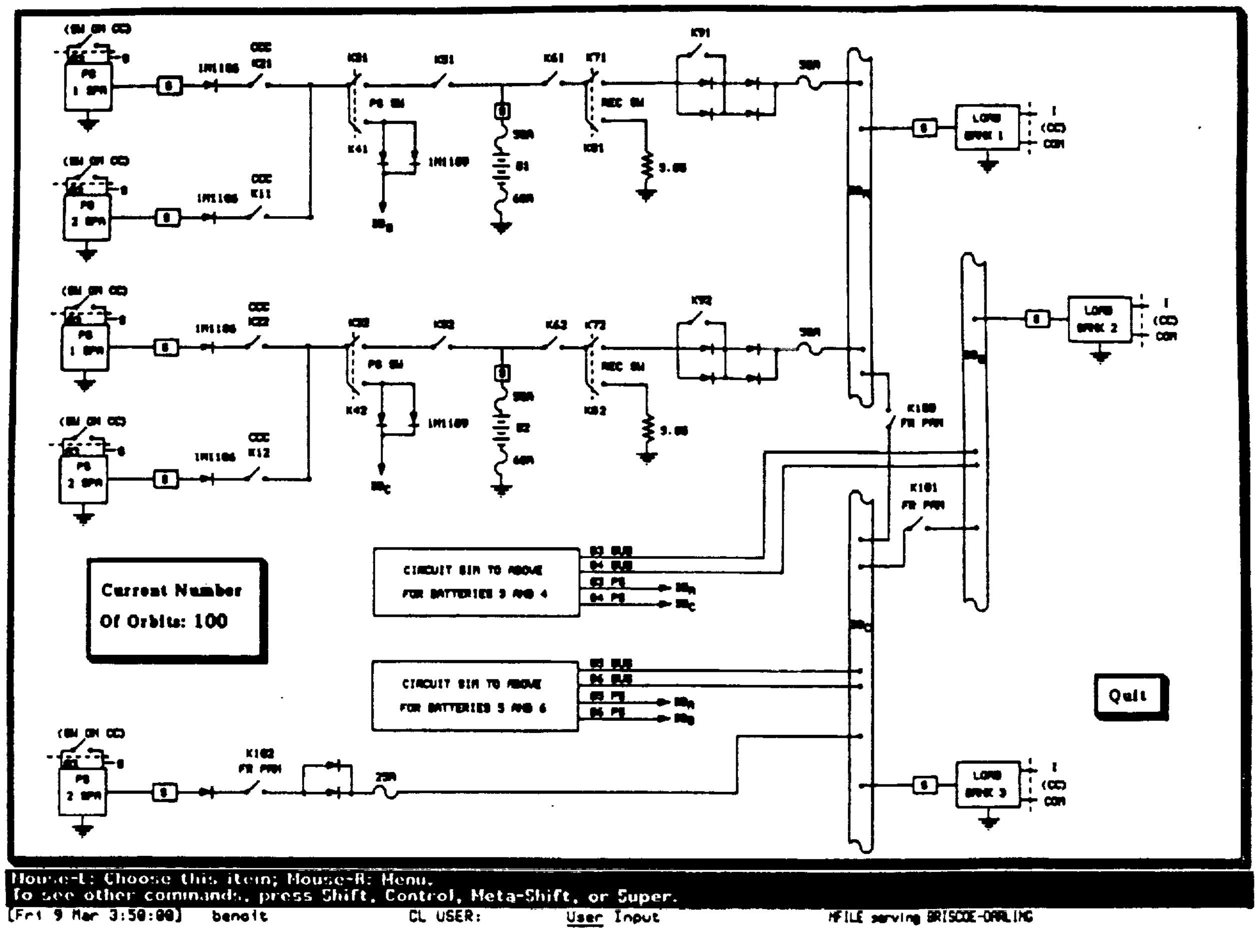


Figure 1 HST EPS Testbed Schematic

The particular area of interest is the health and performance of the six Nickel-Cadmium batteries as they undergo the charge and load fluctuations of the simulated orbiting. In 1986 Martin Marietta developed the Nickel-Cadmium Battery Expert Systems (NICBES) for health management and diagnosis of these batteries [4]. The knowledge base in present system is derived primarily from the warnings and alarms in NICBES. Both systems get their input form a stream of 370 sensor readings emitting form the testbed every minute. Included in these streams are cell voltages and pressures, total battery voltages and currents, and bus voltages and currents [5] [4].

The testbed has proven to be a valuable tool for studying the characteristics of the often complicated lifecyle of these batteries. Although their advantages as an energy storage medium are well known, there is still much to be learned about how to best manage them [3]. The behavioral model of these batteries,then, is not fixed. This explanatory facet of this application made it a prime candidate for the present research.

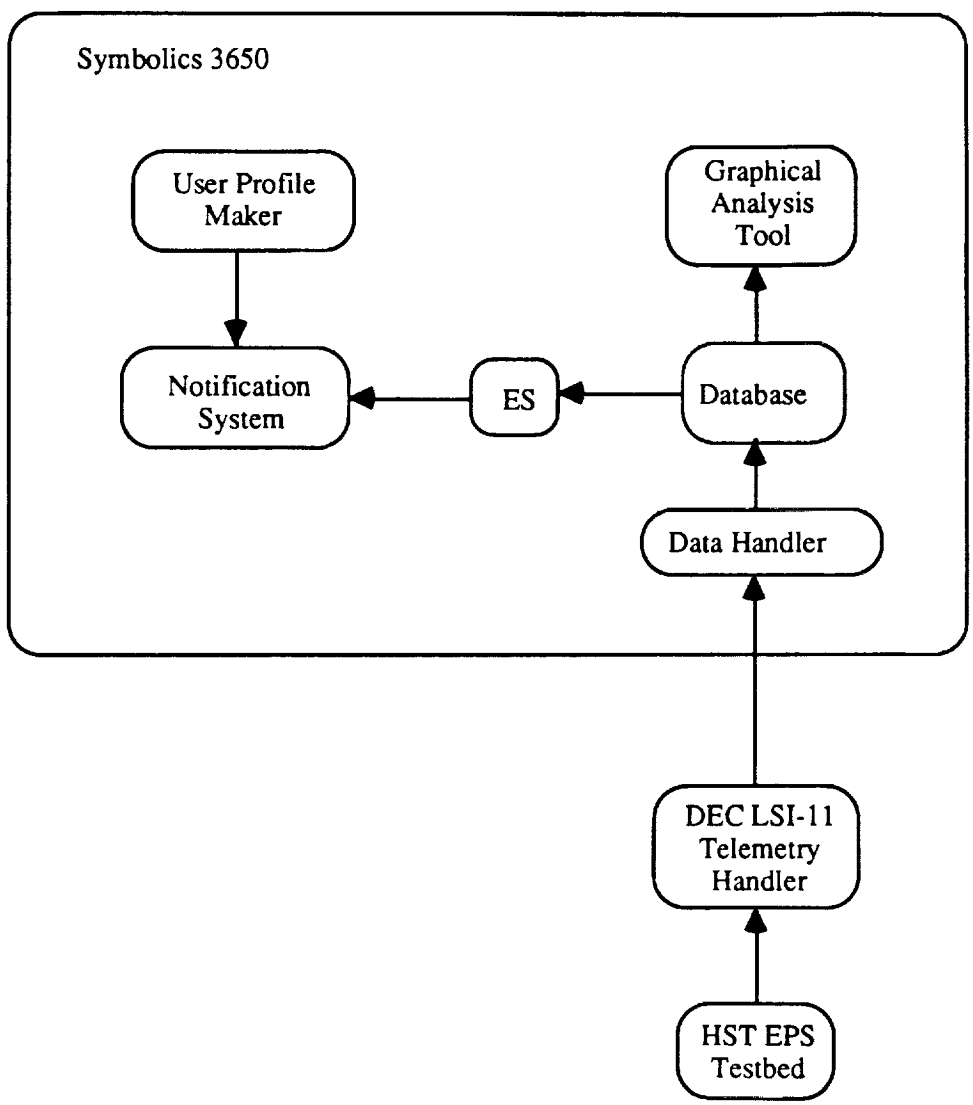


Figure 2 IDARE Functional Schematic

2 System Description

Figure 2 shows a schematic representation of the current IDARE system. The two external components, the HST EPS testbed and DEC LSI-11, were in place prior to this project's inception. The workings of these components have been abstracted from the IDARE system proper. The input for IDARE comes from an RS232 line originating from the DEC LSI-11 located at the HST EPS testbed. IDARE is implemented on a Symbolics 3650 lisp machine located in the Power Systems Lab at MSFC.

2.1 Data Handler

The data handler module runs in its own process, which is launched when the machine is cold booted. Within reasonable limits, the data handler is responsible for ensuring accurate and current information. This process is an infinite loop which reads an incoming burst of 370 data items every minute. After each burst is read, the handler process sleeps for approximately 45 seconds and then wakes up to read the next burst. This idle time helps reduce the number of errors incurred by random noise on the communications line.

2.1.1 Telemetry Burst Format

Data bursts arrive every minute and contain 370 data entries. Figure 3 shows the format of these bursts. Each burst is preceded by the character #\A, used to help insure communications synchronization.

Header Information (Integer):

0	**year**	198x
1	**day of year**	1 to 365
2	**hour**	0 to 24
3	**minute**	0 to 60
4	**second**	0 to 60
5	**orbit**	positive integer
6	**phase**	0 for discharge 1 for charge
7	**day" (minutes in charge)**	0 to 70
8	**night" (minutes in discharge)**	0 to 37

Battery Information (57 entries for each of six batteries):

9 *(66;123;180;237;294)*	**battery number**	integer 1-6
10-32 *(67-89;124-146;181-203;238-260;295-317)*	**cell voltage**	23 reals -2 to +2 volts
33-55 *(90-112;147-169;204-226;261-283;318-340)*	**cell pressure**	23 integers 0 to 150 psi
56 *(113;170;227;284;341)*	**battery voltage**	real 0 to 40 volts
57 *(114;171;228;285;342)*	**battery current**	real -30 to +25 amps (neg. for discharge phase) (pos. for charge phase)
58 *(115;172;229;286;343)*	**BPRC current**	real 0 to 5 amps
59-64 *(116-121;173-178;230-235;287-292;344-349)*	**temperature sensors**	6 reals -15 to +30 deg. C
65 *(122;179;236;293;350)*	**battery recond.**	integer - 0 no, 1 yes

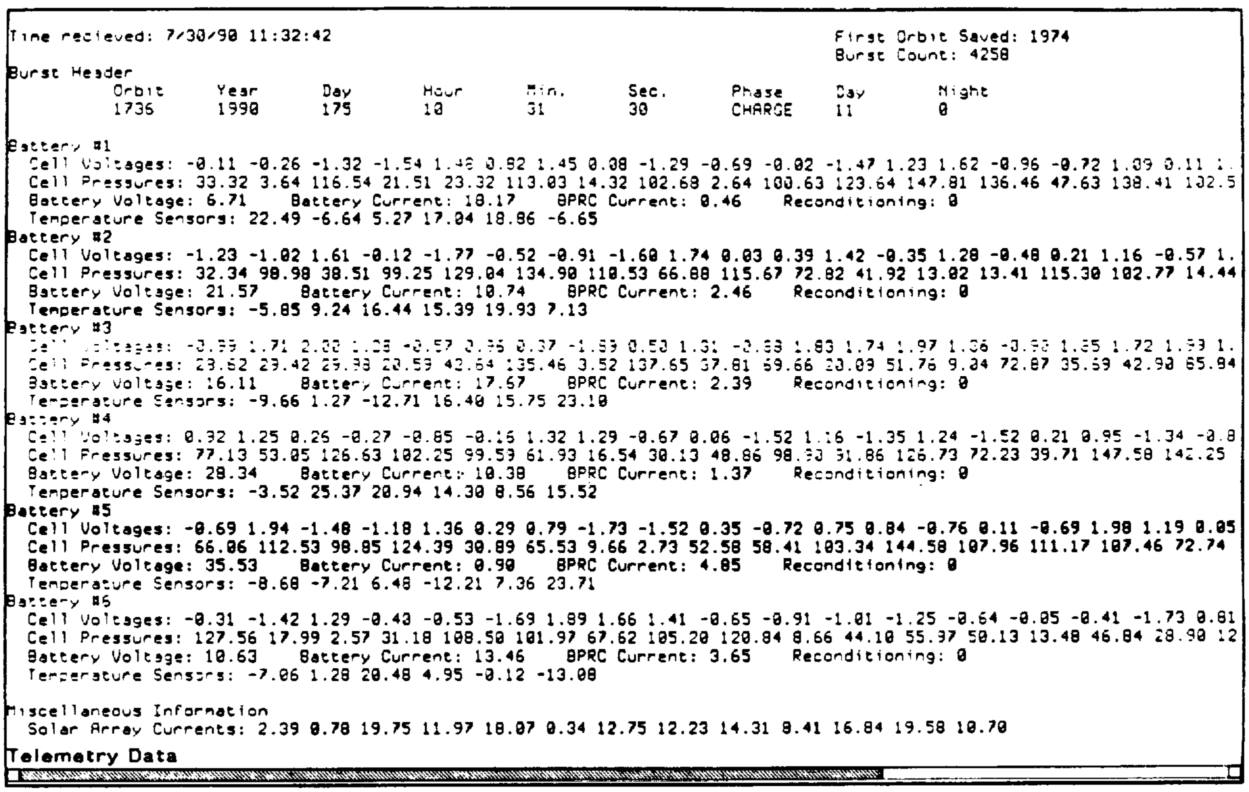
Miscellaneous Information:

351-363	**solar array current**	13 reals 0 to 20 amps
364-366	**bus voltage**	3 reals 0 to 40 volts
367-369	**bus current**	3 reals 0 to 90 amps

Figure 3 Telemetry Burst Format

2.1.2 The Show Telemetry Data Facility

The raw bursts may be examined in real-time by selecting the Show Telemetry Data facility, which is accessible via SELECT ▲. This display is updated once every minute to show the current battery information, broken down on a per battery basis. Figure 4 shows the Show Telemetry Data facility in operation.



```
Time recieved: 7/30/90 11:32:42                                         First Orbit Saved: 1974
                                                                        Burst Count: 4250
Burst Header
        Orbit     Year      Day       Hour      Min.      Sec.      Phase     Day       Night
        1736      1990      175       10        31        30        CHARGE    11        0

Battery #1
  Cell Voltages: -0.11 -0.26 -1.32 -1.54 1.48 0.82 1.45 0.08 -1.29 -0.69 -0.02 -1.47 1.23 1.62 -0.96 -0.72 1.09 0.11 1.
  Cell Pressures: 33.32 3.64 116.54 21.51 23.32 113.03 14.32 102.68 2.64 100.63 123.64 147.81 136.46 47.63 138.41 102.5
  Battery Voltage: 6.71    Battery Current: 10.17    BPRC Current: 0.46    Reconditioning: 0
  Temperature Sensors: 22.49 -6.64 5.27 17.04 18.86 -6.65
Battery #2
  Cell Voltages: -1.23 -1.02 1.61 -0.12 -1.77 -0.52 -0.91 -1.60 1.74 0.03 0.39 1.42 -0.35 1.28 -0.48 0.21 1.16 -0.57 1.
  Cell Pressures: 32.34 98.98 38.51 99.25 129.04 134.90 110.53 66.88 115.67 72.82 41.92 13.02 13.41 115.30 102.77 14.44
  Battery Voltage: 21.57    Battery Current: 10.74    BPRC Current: 2.46    Reconditioning: 0
  Temperature Sensors: -5.85 9.24 16.44 15.39 19.93 7.13
Battery #3
  Cell Voltages: -0.99 1.71 2.02 1.05 -0.57 0.95 0.37 -1.59 0.50 1.31 -0.83 1.83 1.74 1.97 1.36 -0.98 1.35 1.72 1.99 1.
  Cell Pressures: 23.62 29.42 29.98 20.59 42.64 135.46 3.52 137.65 37.81 69.66 20.09 51.76 9.04 72.87 35.69 42.90 85.94
  Battery Voltage: 16.11    Battery Current: 17.67    BPRC Current: 2.39    Reconditioning: 0
  Temperature Sensors: -9.66 1.27 -12.71 16.40 15.75 23.10
Battery #4
  Cell Voltages: 0.32 1.25 0.26 -0.27 -0.85 -0.16 1.32 1.29 -0.67 0.06 -1.52 1.16 -1.35 1.24 -1.52 0.21 0.95 -1.34 -0.8
  Cell Pressures: 77.13 53.05 126.63 102.25 99.59 61.93 16.54 30.13 48.86 98.90 31.86 126.73 72.23 39.71 147.58 142.25
  Battery Voltage: 28.34    Battery Current: 10.38    BPRC Current: 1.37    Reconditioning: 0
  Temperature Sensors: -3.52 25.37 20.94 14.30 8.56 15.52
Battery #5
  Cell Voltages: -0.69 1.94 -1.48 -1.18 1.36 0.29 0.79 -1.73 -1.52 0.35 -0.72 0.75 0.84 -0.76 0.11 -0.69 1.98 1.19 0.05
  Cell Pressures: 66.06 112.53 98.85 124.39 30.89 65.53 9.66 2.73 52.58 58.41 103.34 144.58 107.96 111.17 107.46 72.74
  Battery Voltage: 35.53    Battery Current: 0.90    BPRC Current: 4.85    Reconditioning: 0
  Temperature Sensors: -8.68 -7.21 6.48 -12.21 7.36 23.71
Battery #6
  Cell Voltages: -0.31 -1.42 1.29 -0.43 -0.53 -1.69 1.89 1.66 1.41 -0.65 -0.91 -1.01 -1.25 -0.64 -0.05 -0.41 -1.73 0.81
  Cell Pressures: 127.56 17.99 2.57 31.18 108.50 101.97 67.62 105.20 120.84 8.66 44.10 55.37 50.13 13.48 46.84 28.90 12
  Battery Voltage: 10.63    Battery Current: 13.46    BPRC Current: 3.65    Reconditioning: 0
  Temperature Sensors: -7.06 1.28 20.48 4.95 -0.12 -13.08

Miscellaneous Information
  Solar Array Currents: 2.39 0.78 19.75 11.97 10.07 0.34 12.75 12.23 14.31 8.41 16.84 19.58 10.70
Telemetry Data
```

Figure 4 Show Telemetry Data Facility

ORIGINAL PAGE IS OF POOR QUALITY

2.1.3 Data Structures

The primary data structure used by the data handler is the flavor *burst*. For storage efficiency reasons, there is only one instance of the flavor, whose instance variables are updated once every minute. The important instance variables of the flavor *burst* are time-in, validated-p, raw-data, and error-during-read. time-in is a time stamp indicating the wall clock time when the burst was received, validated-p is a boolean indicating whether or not the burst has been successfully validated, and raw-data is a 1x370 array where the raw telemetry data is stored. error-during-read is discussed in section 2.1.4.

Methods for the flavor *burst* have been written to extract each type of battery data. The method phase, for example, extracts element 6 of the raw-data array and returns CHARGE or DISCHARGE. The *burst* methods parse-burst and validate are discussed below.

2.1.4 Telemetry Validation

The method validate of the flavor *burst* is responsible for checking each burst to insure that communications are synchronized and that no aberrant data was received while the burst was being read. This is accomplished by checking that each element is, as should always be the case, a number and that a few of them are within the expected range (like that the year is greater than or equal to 1990). The instance variable error-during-read is also checked in the validate method. During the actual read, all error checking has been suspended. This was necessary because of the many types of communications errors caused by the imperfect RS 232 line. If an error occurs while error checking was suspended the error-during-read flag is set to signal the validate method that some type of i./o error occured and that the data is unreliable. If for and reason the burst can not be validated, the validated-p flag is set to nil and a communications error message is issued to the IDARE Log facility.

2.2 Database

One of the original goals of the project was to extend the quantity of data available for analysis from 12 orbits (the amount saved by NICBES) to somewhere around 1000 orbits. There seemed to be no need, however, to change the per orbit measures that were established for NICBES.

2.2.1 Contents

The IDARE database is capable of storing 1000 orbit summaries if 4535K 8-bit bytes (or approximately 998 LMFS records) are available. This figure is based on an orbit summary of 907 single-precision floating point numbers, as defined for NICBES. The specific measures stored are shown in figure 5.

Measure	Size
orbit-number	1
high-cell-voltages	6x1
recharge-ratio	6x1
temperatures-on-odd-minutes	6x48
average-temperatures	6x1
high-cell-voltage-at-high-charge	6x1
average-cell-voltage-at-high-charge	6x1
low-cell-voltage-at-high-charge	6x1
cell-voltages-at-high-charge	6x23
EOC-cell-pressures	6x23
minutes-on-trickle-charge	6x1
EOD-battery-voltage	6x1
EOD-cell-voltages	6x23
EOD-high-cell-voltage	6x1
EOD-average-cell-voltage	6x1
EOD-low-cell-voltage	6x1
EOD-cell-pressures	6x23

ORIGINAL PAGE IS OF POOR QUALITY

Figure 5 Database Contents per Orbit

2.2.2 Space Efficiency

The database values have compacted for maximum storage efficiency. Each value (a single-precision floating point number) is stored in 40 bits of the direct access file. This allows for minimum word segmentation, and thus minimum waisted space. The numbers are stored, not as ASCII numbers, but as uninterpreted character strings, each 5 characters long. The functions single-float-to-string and string-to-single-float must be used convert. From the above discussion, we see that the formula for calculating the storage requirements (in LMFS records) of the IDARE database is

$$\frac{\text{\#-orbits} \ * \ \text{(5 bytes/number)} \ * \ \text{(907 numbers/orbit)}}{\text{4544 bytes/LMFS record}} \ .$$

2.3 Expert System

The expert system portion of IDARE is based on the knowledge encapsulated within the NICBES rules. It consists of an alarm system, which checks for irregularities on a per burst basis and an orbit summary generator, which accumulates per orbit analysis information.

2.3.1 Alarms

The alarms generated by IDARE categorized in figure 6. These rules are applied by the data handler for each validated burst. Any anomalies that are detected are sent to the notification system via the function idare-notify.

ORIGINAL PAGE IS
OF POOR QUALITY

Notification Type	Cause
low-spa-current	SPA current < 5 amps during first 5 minutes of charge phase
high-1-spa-current	1-SPA current ≥ 8 amps
high-2-spa-current	2-SPA current ≥ 16 amps
high-discharge-spa-current	SPA current > 5 amps during discharge phase
low-cell-voltage	Cell voltage ≤ 0 volts
high-cell-voltage	Cell voltage > 1.55 volts
high-bus-currents	Sum of 3 bus currents > 99 amps
low-bus-currents	Load < 5 amps on single bus during discharge phase
temperature-out-of-rang	Average of the 6 temperature sensors > 25 C or < -10 C

Figure 6 Alarm Types

2.3.2 Per Orbit Data

The expert system is also responsible for accumulating and maintaining the orbit summary data listed in figure 5. These calculations constitute the statistical data reduction portion of IDARE. The *orbit* flavor contains instance variables for each of these measures. During an orbit, which is approximately 96 minutes long, running averages, maximums, and minimums are being determined so that they can be written to the database. No information is actually written until the orbit is complete. Missed data bursts, because of communications errors, are ignored so that their impact on these statistics is minimized.

2.4 Graphical Analysis Tool

An original goal of IDARE was to provide extensive graphical analysis capabilities to facilitate the continuing study of the Nickel-Cadmium batteries in the testbed. Because these batteries are not very well understood, however, it was not clear which exact facilities would be most useful. Great care was taken, therefore, to insure that the graphical analysis tool should be very flexible and versatile. The tool can be used to plot the data from the historical database in an infinite variety of forms. The user has control over which measures are plotted and how they should be displayed.

2.4.1 Graph Requests

The user can extract and display any information from the database by formulating a graph request. Typically, these are generated, transparent to the user, by the Plot Request Interface. A graph request consists of the measure or measures to be displayed, along with parameters for controlling the look of the graph on the screen. These include parameters for scaling the data along the X, Y, and Z axes, labelling the data, and orienting the plot in three-space. Either 2-D or 3-D plots can be requested.

The 2-D plotting facility is similar to that found in NICBES. The 3-D plotting facility was included specifically to allow the engineer to view the performance of the batteries relative to one another. In this case the data for each requested battery is aligned along the X axis with orbit number running along the Y. Any aberrations of a particular battery will then manifest themselves as mountains on the data terrain. The Y scale parameter is especially useful for adjusting the relative terrain fluctuations so that these mountains can be made more visible.

2.4.2 Efficiency

The time efficiency of the graphical analysis tool is a problem because of volume of data being displayed and the relatively slow disk access times required to retrieve the data from the database. To help mitigate this problem, the graph request editor is designed to make it easier to get it right the first time, meaning that the user may have to wait for the plot he/she has requested, but that its appearance is specified beforehand so the results will most likely be satisfactory. If, however, the user is not pleased with the "look" of the plot on the screen, he/she can replot it relatively quickly, since the data has already been retrieved from the database. The graphical analysis tool automatically keeps track of whether or not it is necessary to retrieve new information from the database.

2.5 Notification System

The notification system is relatively straight forward. It's job is to notify the user of any problems, provided that they have not been filtered out by the current user profile. It receives its input from the expert system and the user profile maker. When the notification system is invoked, via the idare-notify function, it loops through each of the filters in the current user profile looking for any whose predicate is satisfied by the notification. In other words, it tries to find a reason to suppress the notification from the user. If no filter is successful, the notification is sent to the IDARE Log. The IDARE Log is accessible by SELECT ● . Aside from the alarm information, the IDARE Log also displays messages about database initializations, communications errors, and data handler start-ups and shut-downs.

3. References

[1] Bykat, A., "SIRE - A Rule Editor for an Expert System Evaluation of the Hubble Space Telescope Power System," Proceedings of the Third International Software for Strategic Systems Conference, Huntsville, AL, February 1990, pp. 171-177.

[2] Dolce, J.L., Fayman, K.A., "Automating the U.S. Space Station Electrical Power System," Optical Engineering, November, 1986, Vol.25 No. 11, pp. 1181-1185.

[3] Donovan, R.M., Song, L., "An Expert System for Control and Data Reduction in Cytometry," Proceedings of the Ninth Annual IEEE Engineering in Medicine and Biology Conference, Boston, MA, November 1987, pp. 1553-1554.

[4] "Final Report for the Nickel Cadmium Battery Expert System," Report No. MCRR-85-641, Martin Marietta Aerospace Division, Denver, CO, November 1986.

[5] Floyd, S.A., "Intelligent Data Reduction for Autonomous Power Systems," Final Report: NASA/ASEE Summer Faculty Fellowship Program, September 1988.

[6] Rich, Elaine, Artificial Intelligence, McGraw Hill, New York, 1983.

[7] Utt, W.K., et al., "An Expert System for Data Reduction," Proceedings for the Second Conference on Artificial Intelligence Applications: The Engineering of Knowledge-Systems, Miami Beach FL, 1985, pp. 120-124.

[8] Wang, C., et al,. "Automatic Detection of Electric Power Troubles (ADEPT)," Proceedings: Fourth Conference on Artificial Intelligence for Space Applications, Huntsville, Al, 1988.

[9] Weeks, D.J., "Artificial Intelligence Approaches in Space Power Systems Automation at Marshall Space Flight Center," Proceedings of the First

International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, 1988, pp. 361-366

APPENDIX A

IDARE User's Guide

IDARE User's Guide

Introduction

IDARE has been designed to be easily usable, so most of its features should be intuitive. This guide should, however, answer any questions about how to use the program. This document is divided into two sections: one describing how to maintain IDARE and one describing how to run IDARE. The former should be of interest to a very limited group of people, but does provide information vital to the continued performance of IDARE. The latter section should be read by anyone interested accessing the data contained in IDARE's databases.

Maintaining IDARE

Installation

IDARE has been installed on the Symbolics 3650 Lisp Machine in the Power Systems Lab at MSFC. IDARE has been designed to accept its input form the RS232 cable running from HST EPS testbed to the Power Systems Lab. This cable should be connected to the bulkhead serial port on the back of the 3650's processor.

All IDARE system software has been installed on the 3650's LMFS in the IDARE directory. The program will not run properly if these files are relocated or renamed. IDARE was designed to run under the Genera 8.0 operating system.

Loading IDARE

To load IDARE, boot the 3650 into a Genera 8.0 world. If the color system is not included in the world, it can be loaded by issuing the command processor command, Load System Color. Then, issue the Load File command, supplying local:>idare>load-idare.lisp when prompted. All system files will then be automatically loaded. When all files have been loaded, IDARE will be initialized and launched automatically.

These steps will be required whenever the 3650 has been turned off. As IDARE is intended to run continually, collecting data from the testbed, it is recommended that the 3650 be kept on as much as possible. Down-time, however, will only result in lost data. IDARE is able to continue functioning in spite of any lost data. Note also that IDARE continue collecting data as a background process, so that the machine can be used for other purposes.

Maintaining the IDARE Database

The IDARE database, containing battery health information on a per orbit basis, is located in the file, local:>idare>database.data. The primary interface through which to control this database is the Initialize Data Handler command processor command. This command should be used to initiate the data handler if it has been interupted for any reason. Note that the data handler is initiated automatically when IDARE is loaded, so this facility is not generally necessary.

The second, more common, use of this interface is to reset the database. When one issues the command, the system will prompt, "Shall I reinitialize the database file?" Answering yes to this question will enable one to change either the file containing the database or its size. The most probable use of this feature is to increase or decrease the number of orbits stored in the database. This may become necessary as the memory requirements of the 3650 change. The system will lead the user through a dialog to determine the size of the database. Note that the current memory capacity of the machine will only allow for the storage of 500 orbits.

Answering no to the reinitialize database question will cause the system to prompt, "Shall I overwrite the existing database?" Answering yes begins data collection and storage anew with the same database file, thus destroying any information currently stored in the database. Answering no will cause IDARE to continue collecting data, leaving any previous orbital information in tact.

Support

Any inquiries regarding the maintenance of the IDARE system should be directed to the following:

Johnson Research Center
Cognitive Systems Lab
RI PO Box 212
University of Alabama in Huntsville
Huntsville, AL 35899
(205) 895-6217

The IDARE User Interface

The IDARE User Interface can be brought up by pressing the SELECT key and then pressing the square key (SELECT ■). If IDARE has not been properly loaded, the console will beep, indicating that the IDARE program could not be found (see section "Loading IDARE" above, for details on how to load the system). If IDARE has been loaded, the initial window will appear, in color, as shown in figure 1 (note that the figures shown here are monochrome approximations of the actual color windows).

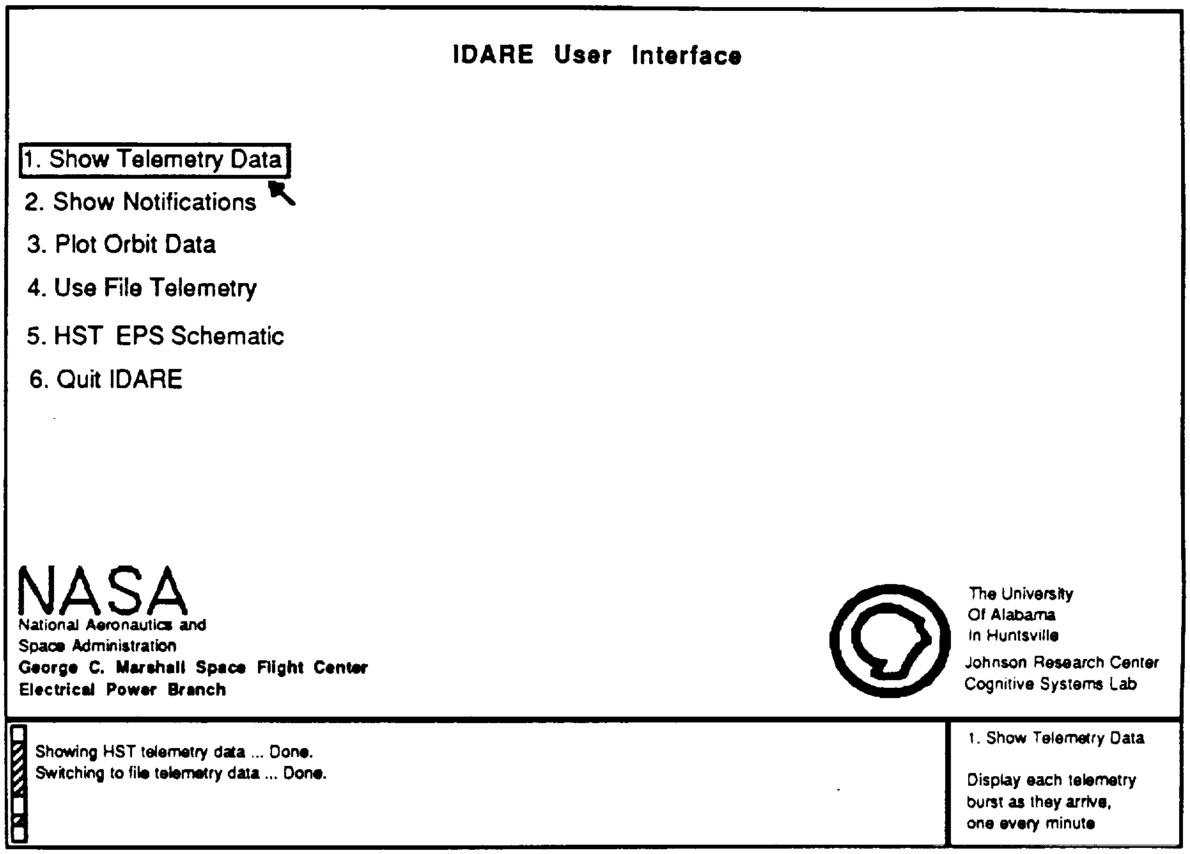


Figure 1
IDARE's initial screen

The contents of the upper window in figure 1 are essentially static, but the lower-left status window and the lower-right help window are dynamic.

The status window continually displays the current task the IDARE interface is executing. The help window provides context-sensitive help about each of the user's current choices. In figure 1 the mouse has been positioned over the Show Telemetry Data menu item. The box around the item indicates that clicking left at this point is a defined action. The help window displays a brief description of what that action entails. Note that each of the menu items can, alternatively, be chosen from the keyboard by simply pressing the appropriate number (1 through 6).

Choosing the Show Telemetry Data menu item, either by clicking the left mouse button on it or pressing 1 on the keyboard, brings up the Battery Telemetry Data window shown in figure 2.

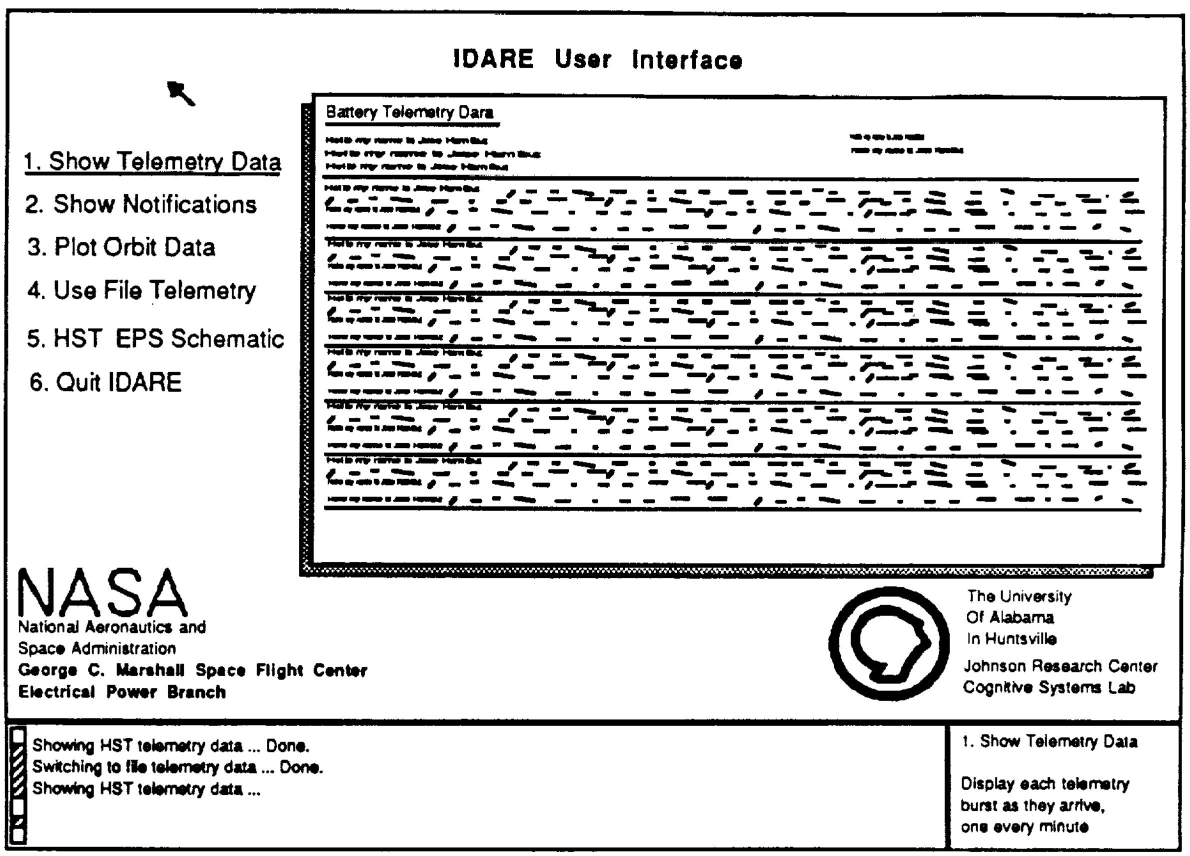


Figure 2
The Show Telemetry Data Option

ORIGINAL PAGE IS
OF POOR QUALITY

The Battery Telemetry Data window is similar to the Show Telemetry Data facility available via SELECT ▲ . This display is updated every minute to show the current telemetry information arriving from the testbed (or from file data as described below). As with all of the IDARE menu options, clicking a mouse button or pressing a key will make the Battery Telemetry Data window to go away.

The second menu option, Show Notifications, has been selected in figure 3.

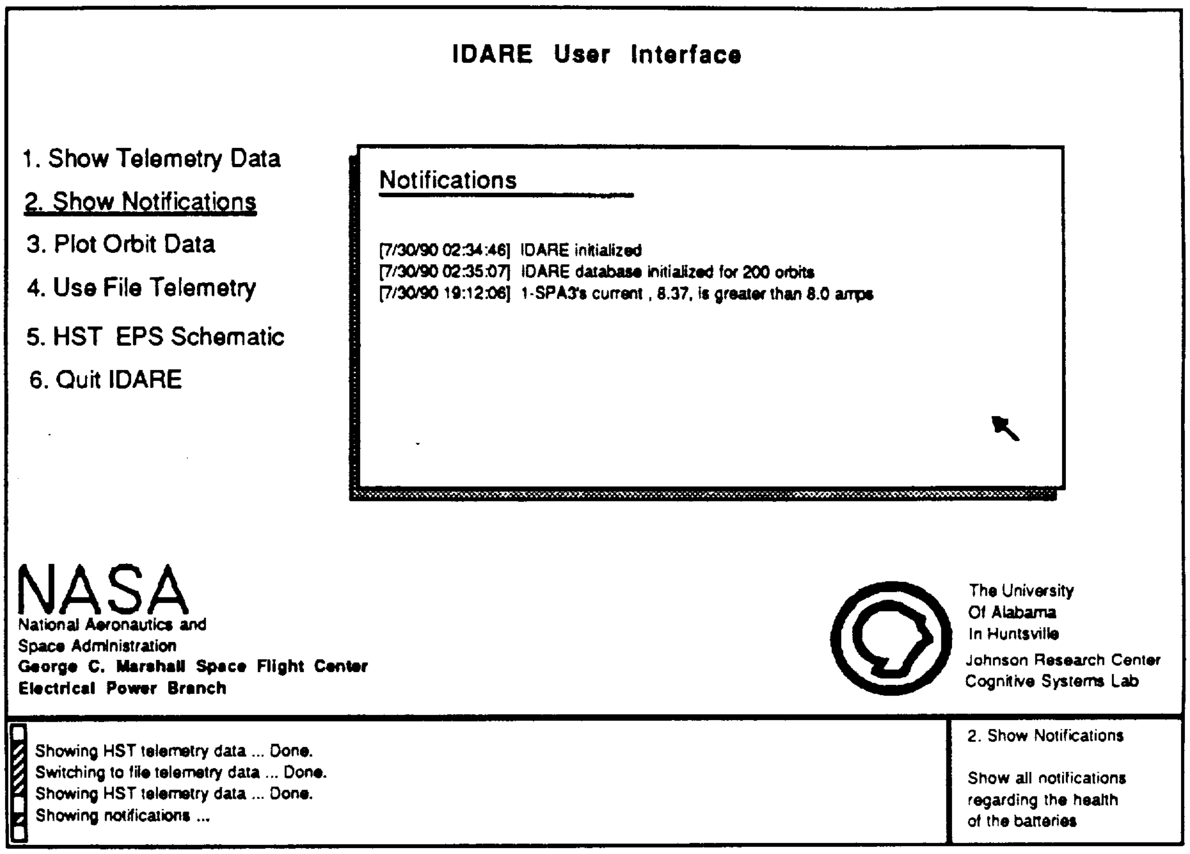


Figure 3
The Show Notifications Option

This option is similar to the IDARE Log facility, accessible via SELECT ●. It shows all notifications concerning the health of the batteries and of IDARE itself. Each notification is timestamped for identification purposes.

ORIGINAL PAGE IS
OF POOR QUALITY

Option number 3 is the Plot Orbit Data facility. This is the primary method for analyzing the orbital data stored in the database. Any information from the database can be accessed via this facility and plotted in 3D in color. When the option is chosen, the Choose Orientation window is brought up as in figure 4.

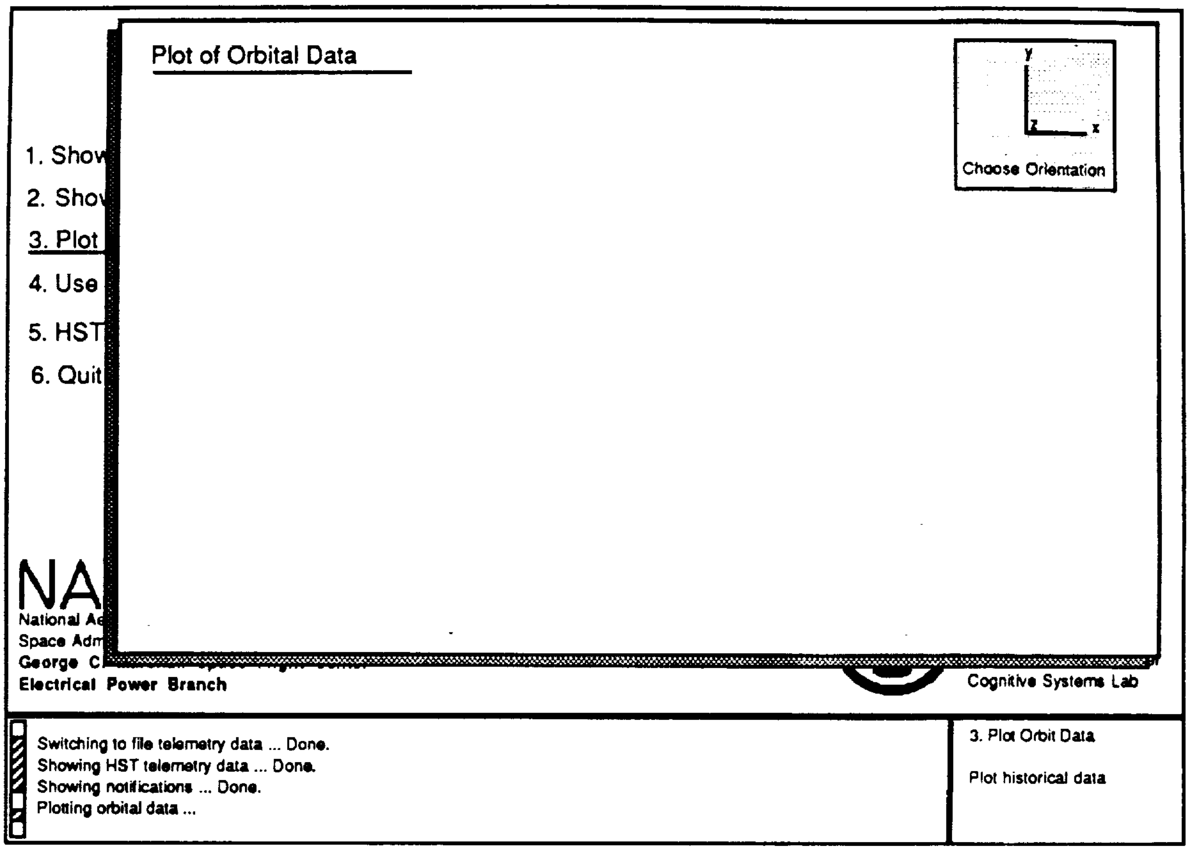


Figure 4
The Choose Orientation Facility

This interface allows the user to interactively specify the orientation of the plot. As indicated in the mouse documentation line at the bottom of the screen, the angle of the x axis can be changed by holding down the left mouse button and dragging mouse. The y and z axes can be set using the middle and right mouse buttons respectively. Once you have positioned the axes as desired, press the END key to proceed.

Next the Choose Plot Parameters menu will appear as in figure 5.

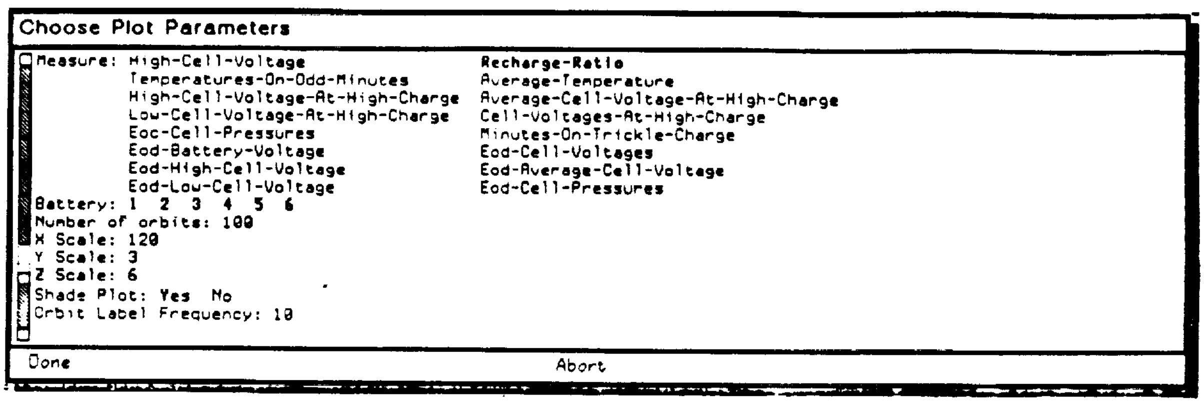


Figure 5
The Choose Plot Parameters Menu

Using this menu the user can customize what is to be plotted and how it is to be displayed. Specifically, you can select the database measure to plot and for which batteries, the number of orbits, the scale factors for the axes, whether or not the plot should be shaded, and how often the orbits should be labelled. Again the mouse documentation line should indicate how to select and modify these various options. Click left on Done when you are satisfied with the plot parameters. The plot will then be drawn as in figure 6. To proceed simply click any mouse button.

ORIGINAL PAGE IS
OF POOR QUALITY

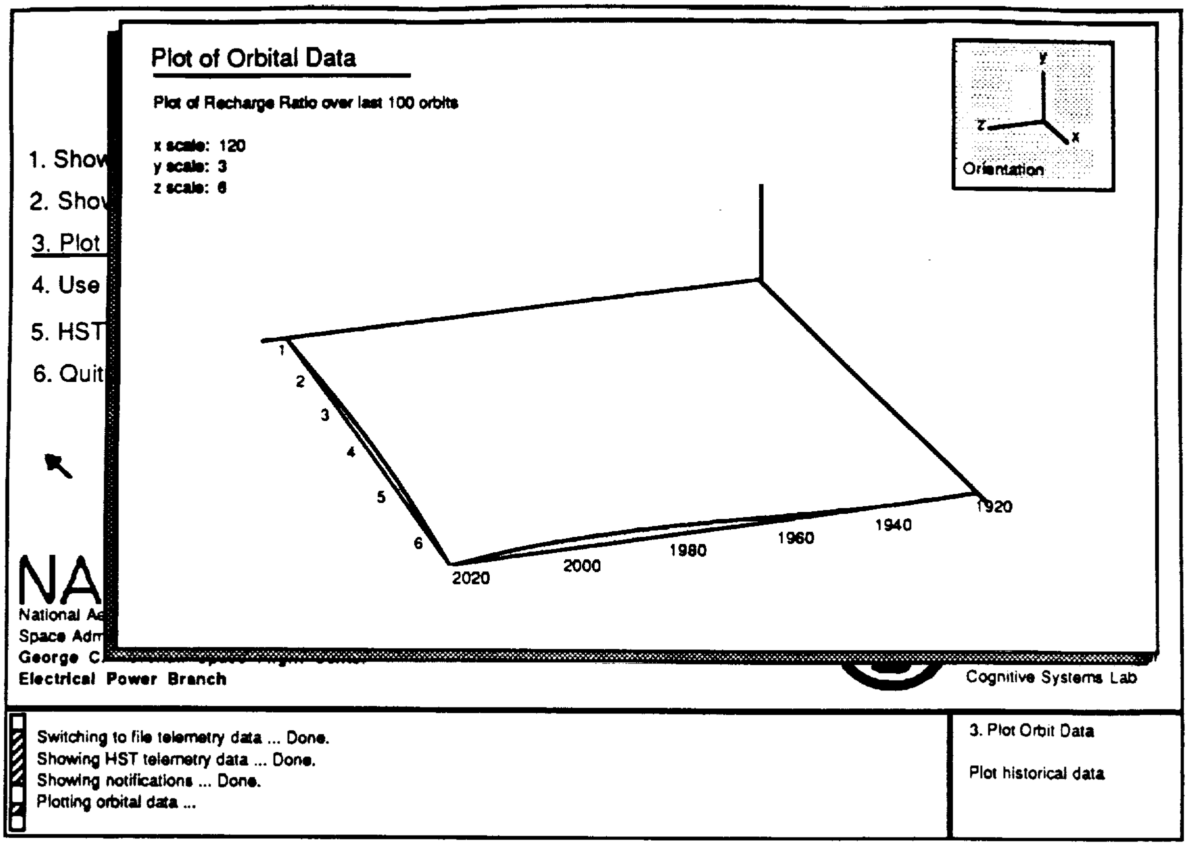


Figure 6
The Plot Orbit Data Option

The remaining three menu options are simple. The Use File Telemetry option has been provided so that the functionality of IDARE can be demonstrated more easily. Telemetry data arrives only once every minute and the chances of one actually observing an aberrant datum at any given time are quite small. So the IDARE system has the capability to accept its input from a file rather than from the HST testbed itself. This file contains unusually bad data, so that IDARE's alarm systems can be demonstrated (and tested) more easily. The Use File Telemetry option does not actually effect the IDARE data handler, which will continue to collect the actual data from the testbed on the background, but only the interface, so it can be used freely without degrading the system's overall performance.

The HST EPS Schematic option has been provided, again largely for reference and demonstration purposes, so that the user can view a schematic of the entire testbed including the Nickel-Cadmium batteries. As with the other options, the schematic window can be deexposed by simply clicking any mouse button or pressing any key.

The last option, Quit IDARE, simply causes the IDARE User Interface window to deexpose. It is recommended that this option be chosen when one is through using the interface, so that the other IDARE subsystems, most notably the data handler, can have complete access to the machine's resources.

Support

Any inquiries regarding the IDARE User Interface should be directed to the following:

Johnson Research Center
Cognitive Systems Lab
RI PO Box 212
University of Alabama in Huntsville
Huntsville, AL 35899
(205) 895-6217