Automated Telescope Monitoring and Diagnosis

David L. Iverson
NASA

Robotic Telescopes
ASP Conference Series, Vol. 79, 1995
Gregory W. Henry and Joel A. Eaton (eds.)

AUTOMATED TELESCOPE MONITORING AND DIAGNOSIS

CHRISTINE M. MONAHAN
Recom Technologies
NASA Ames Research Center
M S 269-4, Moffett Field, CA 94035-1000

F. A. PATTERSON-HINE   DAVID L. IVERSON
NASA Ames Research Center
M S 269-4, Moffett Field, CA 94035-1000

ABSTRACT This paper discusses automated telescope monitoring and diagnosis work currently being performed at NASA Ames Research Center in cooperation with AutoScope Corporation. Topics addressed include information acquisition from the telescope control system, techniques for system monitoring and diagnosis of failures, and the software tools used in the implementation of a monitoring and diagnosis program for a telescope subsystem.

1. INTRODUCTION

Use of untended, fully automatic telescopes bring the need for automated health management for the telescope control system. Automated monitoring of the telescope control system(TCS) allows astronomers to receive data about the TCS without the performance of manual tests by an onsite engineer. Automated fault diagnosis provides the cause of a failure should one occur. Research in this area is currently being performed at NASA Ames Research Center. The primary goal of this research is to provide automated monitoring and fault diagnosis capabilities for remotely operated telescopes to reduce costs and down time. The cooperative effort between Ames and AutoScope Corporation will allow technologies and system models developed at Ames to be transferred to industry. This approach is useful at inhospitable environments such as remote mountain tops or the South Pole, and will be imperative at the planned Lunar Outpost where the luxury of on site engineers is not only impractical but impossible. This approach will allow telescopes to be repaired by technicians, in the case of the Lunar Outpost, astronauts.

2. OVERVIEW

Autoscope's TCS-200 Telescope Control System (TCS) was modeled using digraphs. Digraphs are a type of reliability model that displays the way failures propagate through a system. Closely resembling the schematic, the digraphs of the TCS emulate the functional flow through the system. For example, the failure of the 24 volt power supply is known to cause the failure of the stepper driver function. Therefore, its digraph representation will have the "24 volt power supply" node connected to the "stepper driver" node to indicate this dependency, as depicted by the digraph in figure 1.

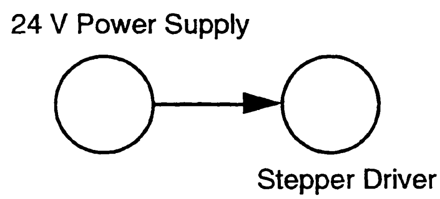

Fig. 1. Simple Digraph Model.

To utilize these digraphs for failure diagnostics, system status will be obtained from AutoScope's Automatic Diagnostic System, the ADS -100. The ADS-100 provides data on 37 key test points within the telescope control system. This enables us to define points in the digraph where component failures could be detected. These points in the digraph are referred to as observation nodes because they represent places in the system being modeled that the user may observe a failure according to test results. The observation nodes will be set as failed according to the findings of the ADS-100. For example, one test point is at the 5 volt power supply. If the ADS-100 finds by sampling the test point that the voltage is not within acceptable limits, the observation node for that test point will be set as failed. This allows the digraphs to be used for near real-time failure diagnostics by propagating the effects of hypothesized failures along the digraph to find component failures which could cause the observed conditions.

In order to utilize these constructed digraphs for automated fault diagnosis, several software and hardware packages must be integrated. Figure 2 displays the data flow through the major components.

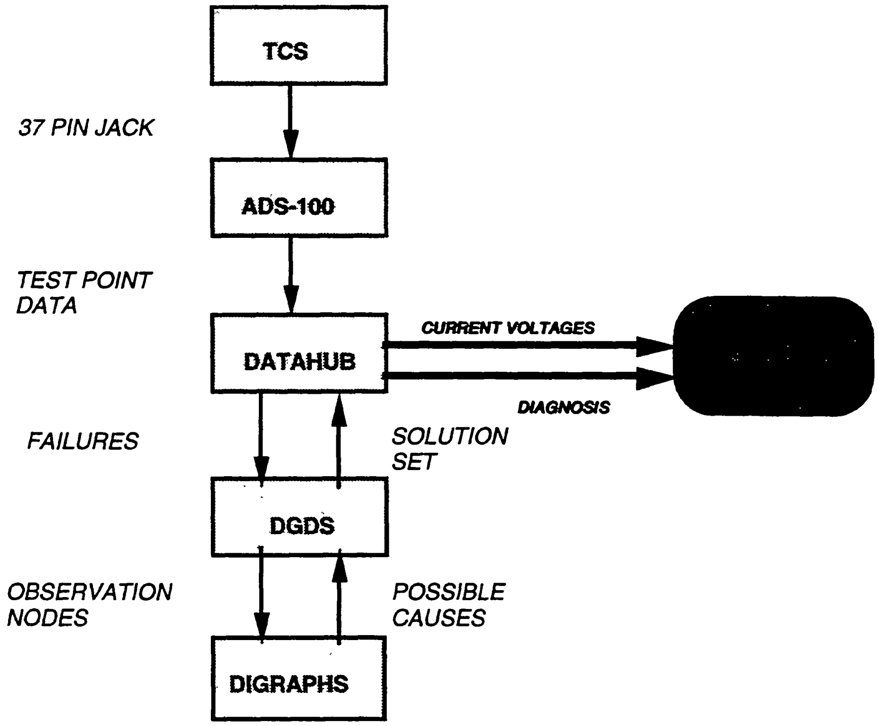

Fig. 2. Data flow through major components.

The following sections discuss the major components involved in the automated monitoring and diagnosis application, with specific emphasis on the digraph models.

3. DIGRAPH MODELS

Digraph models display the way failures propagate through a system. The models consist of nodes and AND gates connected by directed edges. Each digraph node represents a failure in the modeled system. The digraph edges show how the occurrence of a failure can flow through the system to cause other failures. If a node is marked as failed, the failure will propagate through all directed edges leading out of that node and mark any connected nodes as failed. Redundancy is modeled with digraph AND gates. Both input nodes of an AND gate must be marked as failed in order for the failure to propagate past the gate and affect the node on the other side. In a graphical digraph depiction, AND gates are drawn as bars and regular nodes are drawn as circles (see fig. 1). Digraph nodes can be in one of two states, true or false. If a node is true, or *marked*, it means the failure that the node represents has occurred. If the node is false, or *not marked*, then the failure has not occurred.

Digraph models can be derived from system schematics in a fairly straightforward manner by associating a digraph node with each component in the schematic, adding directed edges to represent physical connections, and augmenting that basic digraph with knowledge about component failure modes and system design.

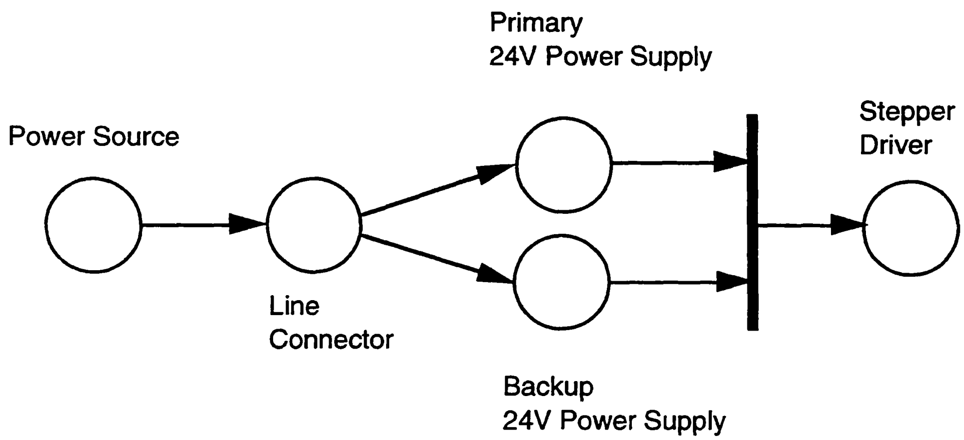

Fig. 3. Stepper Driver Digraph.

Figure 3 shows a simple, hypothetical digraph model of part of a telescope control system. Each node represents a failure of a component of the system and the edges correspond to the flow of power through the system. The stepper driver supplies current to the windings in the stepper motor (linear actuator). Notice that a failure in the power source or the line connector could propagate and cause the power supplies to fail in their function of delivering power to the stepper driver. The bar in the digraph is an AND gate which indicates that both the primary and backup power supplies for the stepper driver must fail to operate before the stepper driver fails due to lack of power. Depending on the desired level of detail, additional nodes (e.g., fuses, switches, connector pins) could be added to this digraph.

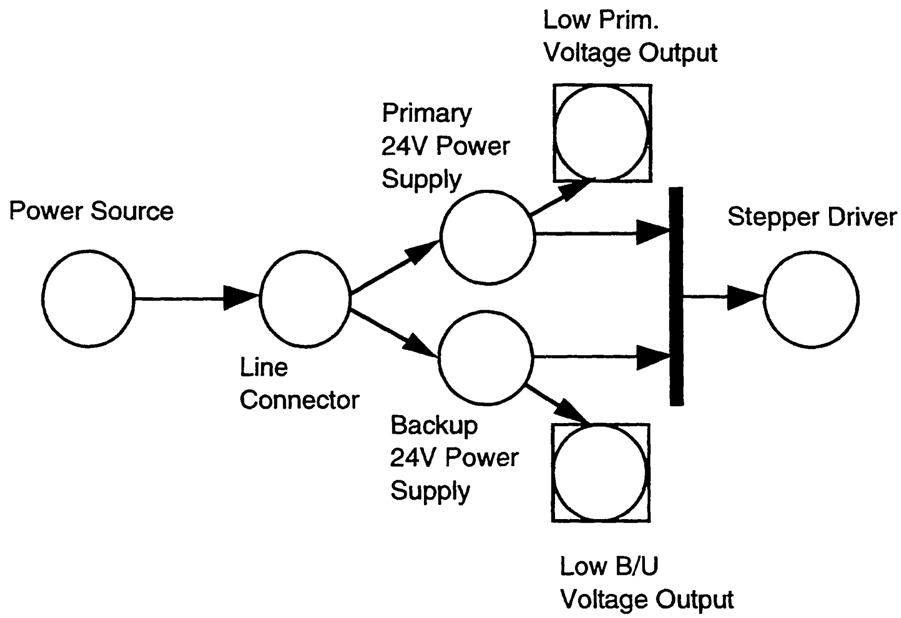

Fig. 4. Stepper Driver Digraph with Observation Nodes (square outline).

In order to use digraph models for fault diagnosis we must include points in the digraph where information regarding the health of the TCS is available. Points where actual system information is available is modeled using observation nodes. Observation nodes correspond to out-of-limits sensor conditions. These observation nodes are added to the digraph as outputs of failures that could cause the sensed condition. In terms of the TCS, these are test points where voltage levels are measured and found to be out of acceptable limits. Figure 4 shows observation nodes that might be added to the stepper driver digraph shown in figure 3. In this example, two observation nodes have been added and are indicated by the nodes with the square outlines. These new nodes correspond to test points with in the TCS where voltage levels for the power supplies can be sampled.

The following sections describe the application that will use the digraph model technique described above for fault diagnosis.

4. TCS DIGRAPH MODELS

The actual digraph models of AutoScope's TCS-200 are much larger and more complex that the previous two digraph examples. Much like the system schematic, the digraphs proceed from the Telescope Control

Computer (TCC) out to the major components. The nodes represent functional failures that can be experienced along the path. For example, as a signal is sent to the linear actuator, there are a number of problems which could be encountered along the way. The stepper driver, its power supply, connector pins, or cables located between the stepper driver and the linear actuator could fail. Hardware failure of one of these components would result in the failure of the linear actuator to perform its task. In the digraphs, this is represented with a node for each of the components. The components are linked together in a string from the TCC through the stepper driver and power supply out to the linear actuator. Thus the functional flow and dependencies are included in the model.

Information about AutoScope's TCS-200 telescope control system was obtained from the system schematic, electrical diagrams, technical manual, and AutoScope's principle engineer. Once sufficient information about the design of the TCS-200 was obtained, the TCS digraph models were constructed using the Digraph Editor, a software package developed for NASA by Lockheed Engineering and Sciences Company. Each piece of the digraph model was checked by the principle engineer to ensure the correct functional flow and dependencies were captured. The final digraph model of the TCS-200 telescope control system consists of hundreds of nodes. Each node is represented by a circle with a text block identifying it. The text block consists of a 16 character mnemonic, a component description, and a schematic link. The 16 character mnemonic is used by the digraph processor in its analysis. The first character in the mnemonic represents the node type. "H" stands for hardware node, "F" for function node, and "O" for observation node. The next twelve characters represent the component. For example, "TCS_J106_P25" stands for jack 106 pin 25. The last three characters in the mnemonic represent the failure mode. "_00" stands for "fails out of limit," "_01" stands for "fails," and "_90" stands for "fails to function." The component description part of the text block simply makes the digraphs easier to read; it plays no functional role in solving the digraphs. The schematic link provides a link between a node and its corresponding component in the system's schematic.

The digraph in figure 5 represents the possible TCS failures that affect the instrument selector function; it is a small portion of the actual TCS-200 digraph. The failure of the instrument selector to perform its task is represented by the node on the far right. The nodes representing the failure of the 5V and 24V power supplies are represented in by the nodes with triangular outlines. This outline is simply an indication to the modeler that this node is repeated elsewhere within the digraphs. As seen in the digraph, a signal is sent from the 24V power supply to the instrument selector and out to the output modules for the four ports of the selector. The level of detail with which the TCS-200 digraphs were constructed can also been seen in this example. Between the 24V power supply and the instrument selector, the model includes hardware failures on the connector pin level, this level of detail is maintained between the instrument selector and each of the four output modules. Each output module is additionally dependent on the 5V power supply. Therefore, as indicated by the digraph, any of the four output modules would fail to perform its task if either the 5V or the 24V power supply failed.

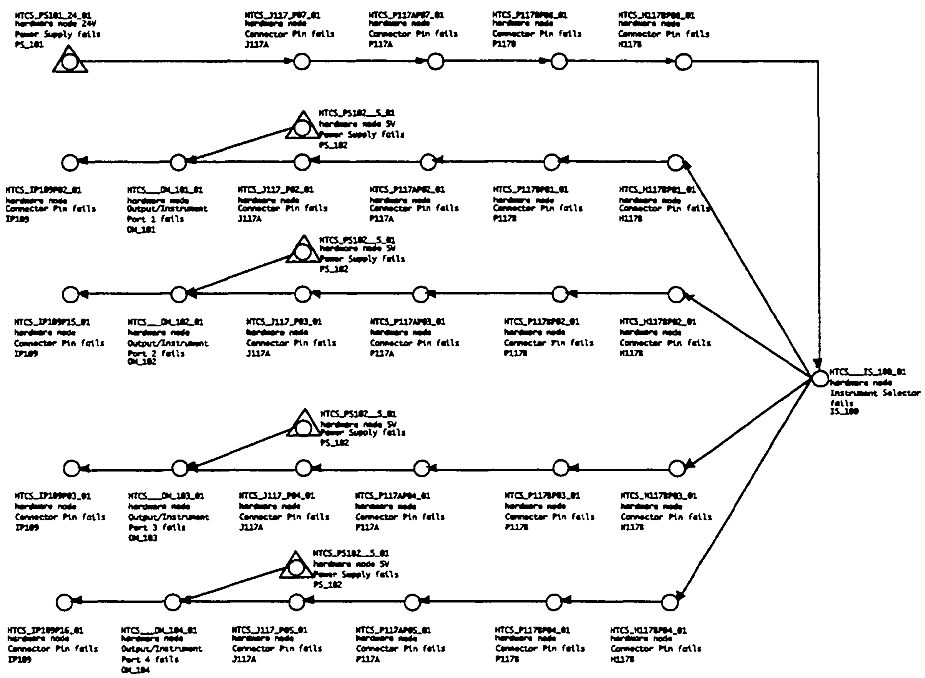

Fig. 5. Digraph Model of the TCS Instrument Selector Function.

5. DATAHUB

Once the ADS-100 has sampled the test points within the TCS, DataHub, a software package developed at Ames, can determine and display which limits have been maintained and which have been exceeded. DataHub is a UNIX based C program running with X-Windows. The TCSView portion of DataHub provides the user with the telescope control system schematic. Red buttons at key points in the schematic depict the test points. Clicking on one of these buttons creates a new window which displays the acceptable voltage range and the current voltage level for that particular test point as obtained from the ADS-100. DataHub also allows the status of all the test points to be viewed simultaneously. This allows the astronomer to continuously monitor the

TCS without an on site engineer manually performing tests. Viewing the voltage levels straight from the TCS without operation interruption saves time and money. Furthermore, the display of the test point data over time allows trends to be detected and possibly a fix before a failure occurs.

6. DIGRAPH DIAGNOSIS SYSTEM

Those test points which have failed out of limit will be marked as failed in the digraphs via the Digraph Diagnosis System (DGDS). DGDS is a UNIX based C program also developed at Ames. This digraph solution algorithm marks the failed test points in terms of observation nodes within the digraphs. It then solves the digraphs by backward propagation of the failure to determine the possible causes for the observed set of failures. To see how this is useful, look back at stepper driver digraph in figure 4. Suppose we know from the system data that the primary 24V power supply has failed to be within acceptable limits. By backward propagation it can be determined that this failure could be due to the failure of the 24V power supply, the line connector, or the power source. However, if the test point for the backup 24V power supply yields an acceptable voltage then we know the observation node is considered good. Therefore, also by backward propagation it can be determined that the back up 24V power supply is functioning correctly as well as the line connector and power source. Thus, the fault has been isolated to the primary 24V power supply.

After fault detection and isolation, the nodes which may have caused the observed failure symptoms will be displayed to the astronomer. This failure diagnosis will be viewed by the astronomer through the DataHub package.

7. CONCLUSION

The digraph model is a graphical representation of a system using nodes, edges, and-gates, and or-gates. The failure of system components are represented by nodes, which are linked together in logical failure sequence by lines and arrows(edges). Digraph models of Autoscope's TCS-200 will be used as part of a automated monitoring and diagnosis system. Automated information acquisition will provide researchers with the status of the control system health without performing manual tests. Furthermore, the integration of several software programs together with the TCS digraphs will enable automatic fault diagnosis.

8. FUTURE WORK

Interfaces between the ADS-100, DataHub and DGDS are yet to be developed. Integration of DGDS and DataHub is planned for the end of fiscal year 94. Integration of the health monitoring software and the ADS-100 is planned for the following year, and will be followed shortly there after by testing on site at AutoScope Corporation. Once data from ADS-100 is available via Internet/ATIS93, the monitoring and diagnosis system will be an optional part of the Associate Principle Astronomer (APA) currently being developed at NASA Ames. The future goal of the integration with the APA is to provide a reactive planning system that can reschedule telescope activities to enable the telescope to continue being used in a degraded mode even when component failures occur.

ACKNOWLEDGMENTS

We wish to thank Ken Valentine for his continued support and many hours of helping us to capture some of his engineering knowledge in the form of digraphs.

REFERENCES

Iverson, D. L. and Patterson-Hine, F. A. 1990 in *A Diagnosis System Using Object-Oriented Fault Tree Models* , The Fifth Conference on Artificial Intelligence for Space Applications Proceedings, Huntsville, AL.

Iverson, D. L. and Patterson-Hine, F. A. 1993 in *Digraph Reliability Model Processsing Advances and Applications*, AIAA Computing in Aerospace 10 Proceedings, San Diego, CA.

Patterson-Hine, F. A., Boyd, M. A., and Iverson, D. L. 1992 in *Automated Monitoring and Diagnosis of Telescope Control System*, SALUTE Workshop Proceedings, NASA ARC.

Valentine, K. M. 1994 in *Self Diagnostics of Automated Telescopes,* Optical Astronomy from the Earth and Moon, ed. D. Pyper Smith and R. Angione, ASP Conference Publication Volume 55.

Drummond, M., Bresina, J., Swanson, K., Edington, W., Drasher, E. and Henry, G. 1994 in *The Associate Principle Astronomer*,, The 106th Annual Meeting of the Astronomical Society of the Pacific Proceedings, Flagstaff, AZ.