USING A VISUAL PROGRAMMING LANGUAGE TO PROCESS SPACE SHUTTLE TELEMETRY DATA

Francisco Razo and James McGregor
Measurement Technology Center,
Jet Propulsion Laboratory
California institute of Technology

Abstract

In October 1994, the Measurement Technology Center (MTC) was asked to provide ground support equipment (GSE) to process telemetry data from the Astro Space Shuttle mission (an advanced star and target optical reference sensor) that was to occur in February 1995, a mere four months away. The project had only enough funds to pay for approximately six work-months of engineering and two PC computers, This paper describes the design and implementation of this task with emphasis on the time-saving use of visual programming, off the shelf hardware, and inexpensive PC computers to complete the task on time and within budget. The mission was a great success and the GSE was a major contributing factor. The success was also due to the close interaction among the customer, the user, and the engineer, and the ease with which the end-user can make his own modifications to the software.

Introduction

The MTC evaluates commercial data acquisition, analysis, display and control hardware and software products that are then made available to experimenters at the Jet Propulsion Laboratory. The MTC specifically configures and delivers turn-key measurement systems that include software, user interface, sensors (e.g., thermocouples, pressure transducers) and signal conditioning, plus data acquisition, analysis, display, simulation and control capabilities.[1]

Visual programming tools are frequently used to simplify development (compared to text-based programming) of such systems, specifically LabVIEW and HP VEE. Employment of visual programming tools that control off-the-shelf interface cards has been the most important factor in reducing time and cost of configuring these systems. The MTC consistently achieves a reduction in software/system development time compared to text-based software tools tailored specifically to our environment.[2,3] Others in industry are reporting similar increases in productivity and reduction in software/system development time and cost.[4]

LabVIEW is a visual programming language developed by National instruments, inc. "Programming" in LabVIEW is accomplished using icons, similar in function to the subroutines of conventional, text based programming languages. Data flow among the icons is provided by connecting these icons with lines called wires.[5] The result is much like a wiring or schematic diagram (Figure 1).

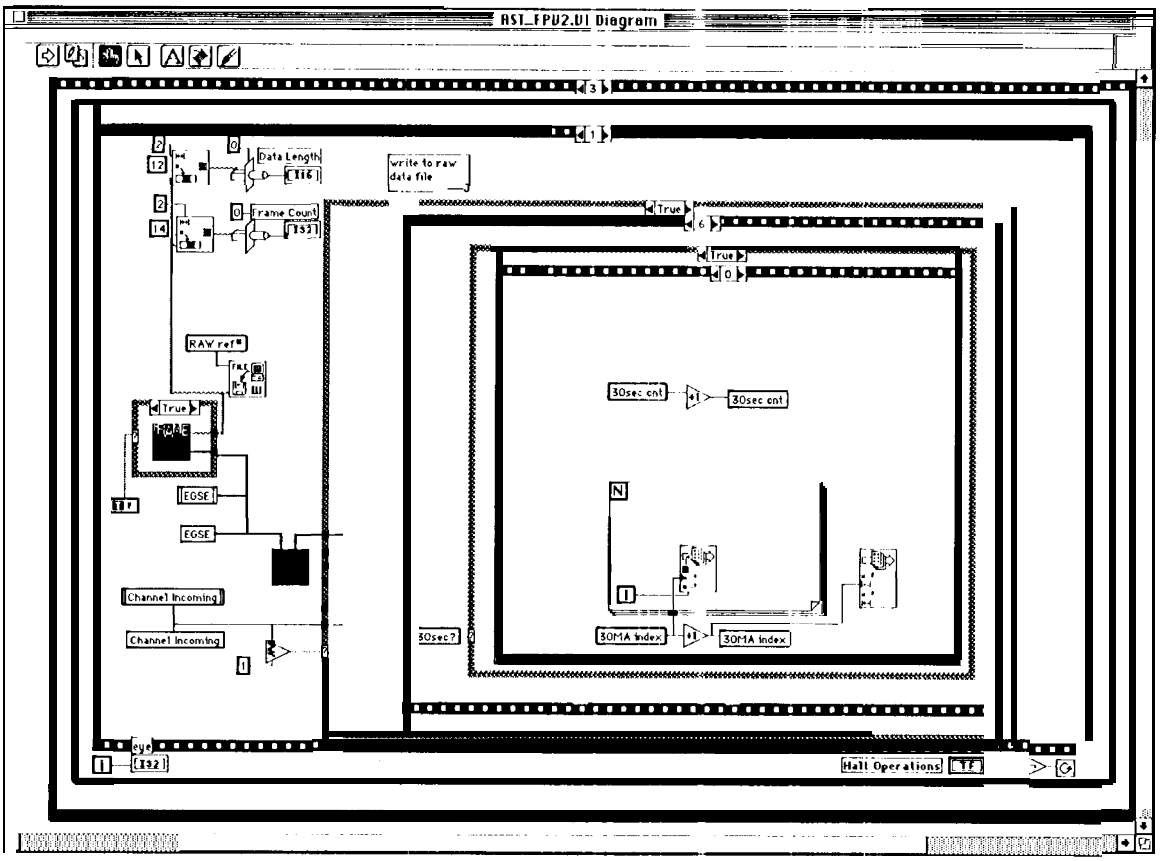

Figure 1. LabVIEW visual programming diagram. This is the actual program.

Once the visual language is mastered, complex programs can be created much faster than with conventional text-based languages.[6] Also, with a little knowledge of the language, one can understand enough of the diagram to make minor, and at times even major, modifications without complete knowledge of the language required to create a complete program. The gains in productivity are attributed to the communication among the customer, developer, and computer that are facilitated by the visual syntax of the tools.

Background

Astro-1 was an observatory equipped with instruments sensitive to ultraviolet light and X-rays. It was flown on the space shuttle Columbia i n December of 1990 to gather data on stars, galaxies and quasars.[7,8] Due to the success of the mission, a follow up mission, Astro-2, was planned. Astro-2 was to continue the exploration of the invisible universe of ultraviolet. It was to capture a large number of images, including globular clusters (using an Ultraviolet imaging Telescope).

in October of 1995, the MTC was asked to provide Astro-2 mission team with ground support equipment for processing the telemetry flight data and sending commands to the flight equipment. The telemetry data was to be provided in near real-time by the Payload Operations Control Center (POCC) at Marshall Space Center in Huntsville, Alabama using IEEE RS-422 communication lines.[9] The data stream (Figure 2) consisted of a header containing sync words, time, etc. and 28 Astro-2 data parameters --14 High Rate Multiplexer (HRM) parameters and 14 Experiment Computer input/Output (ECIO) parameters. The data was to be displayed as it arrived, as well as archived in data files for later analysis.

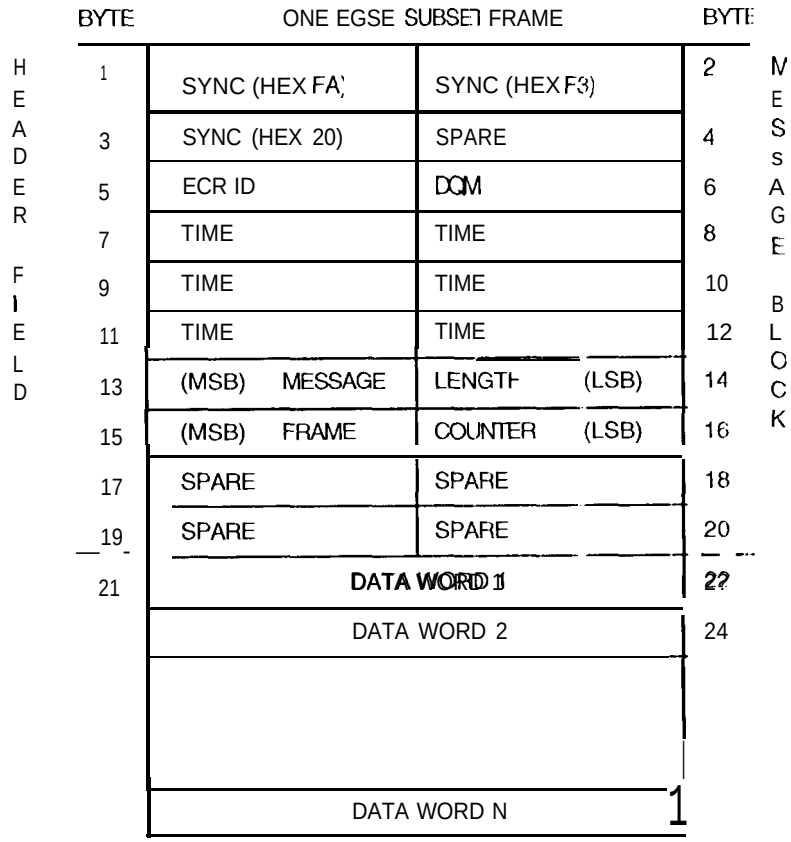

Figure 2. GSE Subset Frame

The Equipment

The equipment (Figure 3) consisted of a Pentium/60 computer with 32 megabytes of RAM memory, a 500 megabyte hard disk drive, a super video graphics adapter, and an IEEE RS-422 input/output card (required by the POCC). The computer display was a **14** inch, high resolution monitor. A second Pentium/60 computer was used as a backup system and for examining archived data.

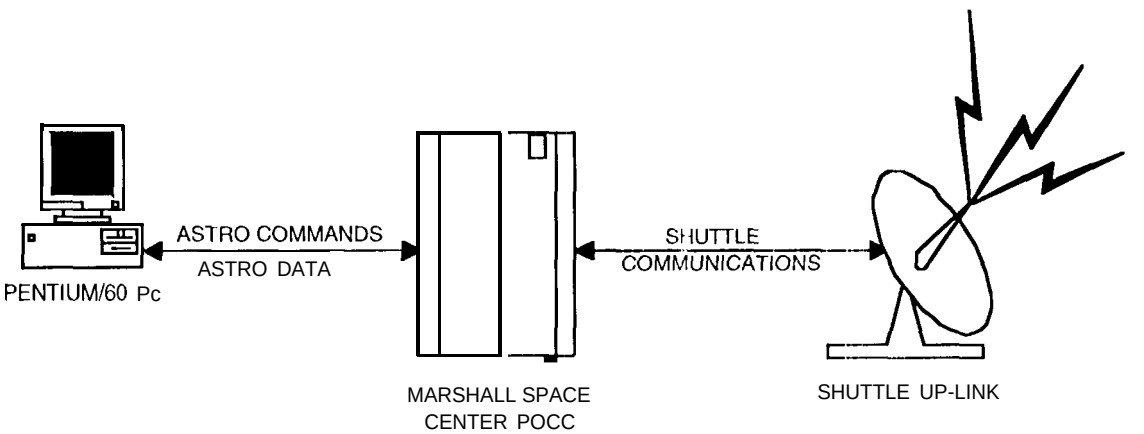

Figure 3. Ground support equipment schematic.

The User Interface and the Program

Due to programming time-constraints, the user interface (Figure 4) was to be rather simple. The LabVIEW programmer worked very closely with both the customer and end-user in developing the computer interface, meeting with one or the other several times per week.

A routine for generating simulated telemetry data was created for testing the user interface. This was accomplished using the pseudo-multitasking ability of LabVIEW. The simulated data was output on the 1< S-422 port with a loop-back connector installed (Figure 5) and read back with the user interface program.

As data arrived, the program checked for the beginning of a data frame by looking for predefine sync patterns. Once a new data frame was located, the data was demultiplexed, sorted, displayed in tables (Figure 6) and strip charts (Figure 7), and archived in disk files. Periodically the archived disk files were written to tape.

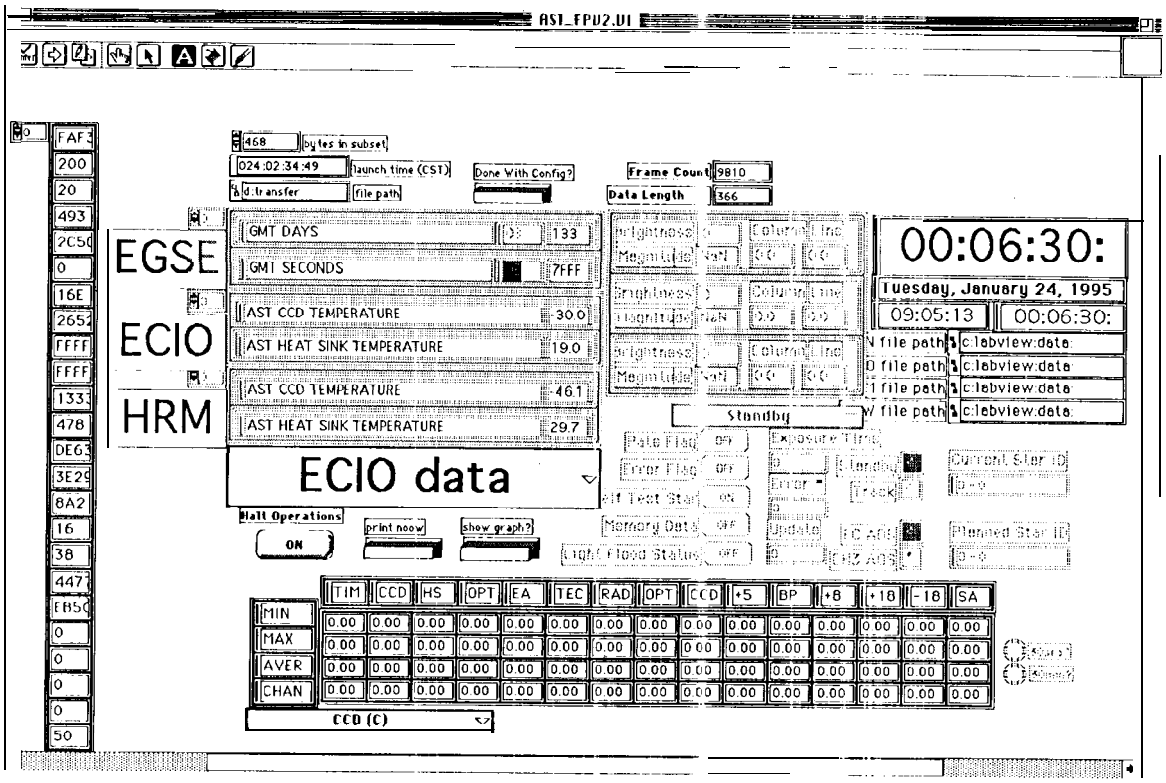

Figure 4. User interface (LabVIEW front pane])

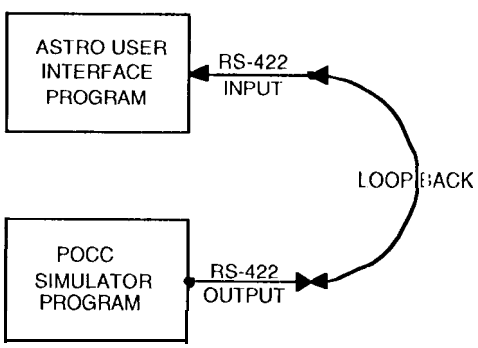

Figure 5. Testing the user interface.

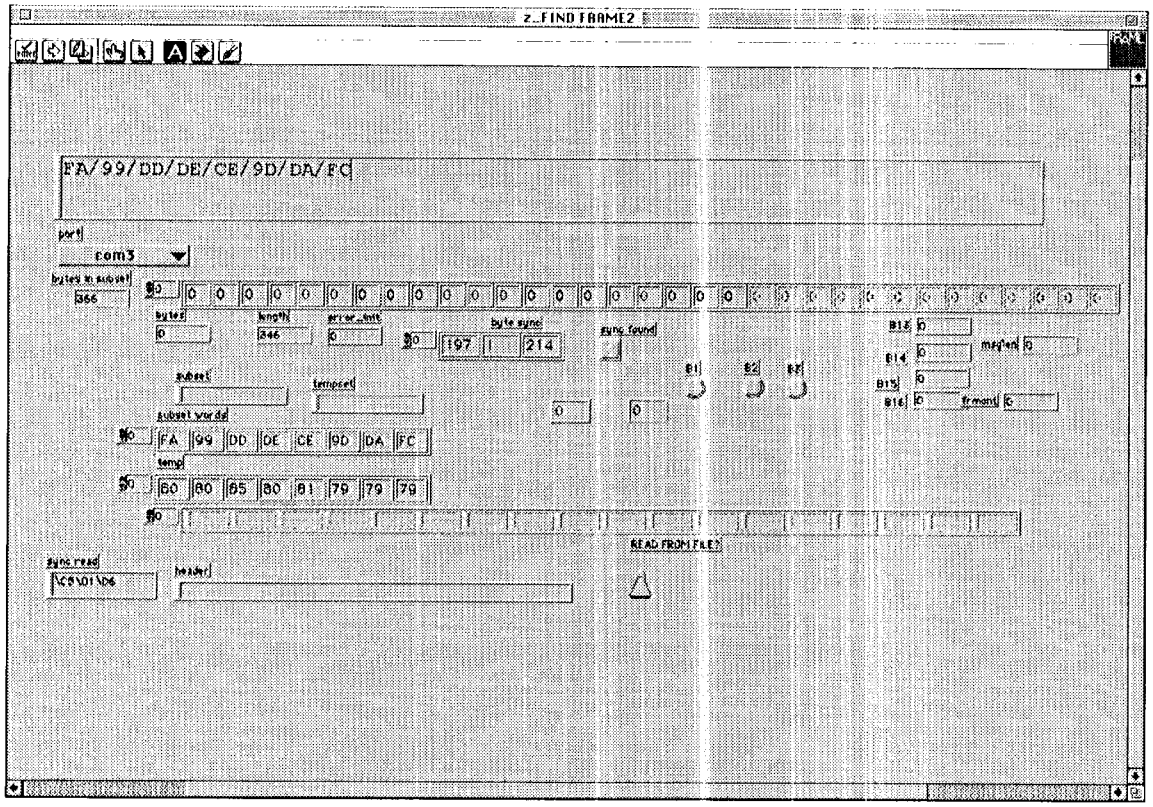

Figure 6. Data Displayed in tabular format.

To allow the viewing of both HRM and ECIO data, a control was placed on the panel to allow switching between the two data sets. As the switching occurred, both the values and the labels changed to reflect the new data set. This was true for both the tabular and the graphical display of data.

The time and frame count from the header were also displayed to check for missed data frames. It was crucial that no data be missed. A frame checker was constructed to verify that all of the data from the frame had been acquired by extracting the message length from the telemetry frame and comparing it to the length of the obtained data. Considerable of time was devoted to checking program operation since it would be running continuously for 17- 21 days.

Testing at Marshall Space Flight Center

During initial testing by the user at the Marshall Space Center it was found that the order of the 28 parameters was different from what had been originally assumed, With the help of a few phone calls to the LabVIEW programmer at JPL the software was quickly modified to correct the problem.

The data acquisition, display, and storage were successful during the initial testing once the these changes were made.

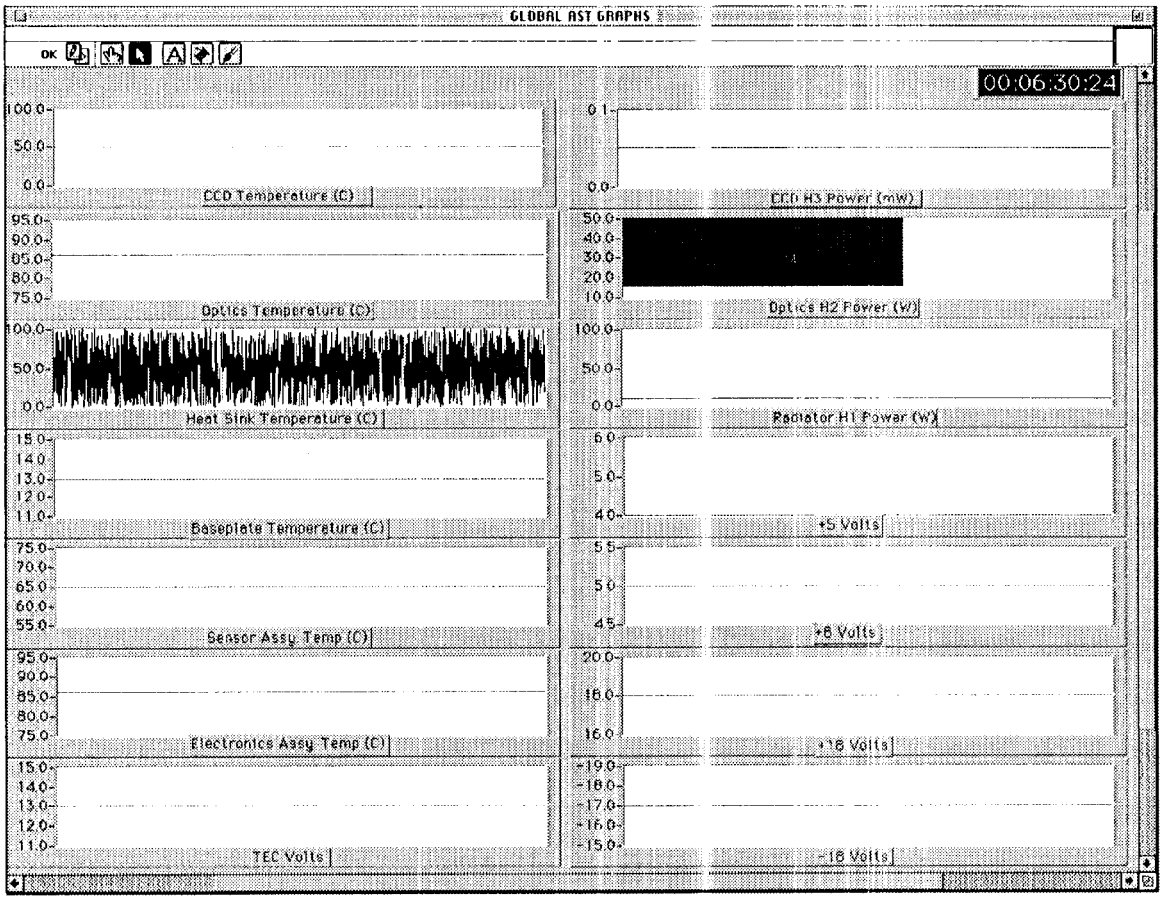

Figure 7. Graphical display of data.

The Actual Flight

Just before flight, the user found that the programs subset size was too small and needed to be increased. This was due to the fact that the shuttle's telemetry stream contained more data than was previously expected. The program was quickly modified by the user to read the larger data subset.

During the actual flight the customer requested to see other data that was not being displayed. During the times of LOS (Loss of Signal) the system was halted without any adverse effects or loss of data and the modifications were quickly made by the user to display this other data.

Conclusion

The task was completed on schedule and within budget. It was accomplished by using a visual programming language and having the customer and user work closely with the programmer. The resulting reduction in time-consuming and costly modifications to the software was one of the primary reasons for the successful completion of this project.

The customer was pleased to have the capability to see the real-time data as it was being received. The additional feature of being able to make modifications during the flight without any data loss was an unexpected bonus for the customer.

Acknowledgments

The research described in this paper was carried out by the Jet Propulsion Laboratory, California institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. E. C. Baroth, D. J. Clark, and R. W. Losey, "Acquisition, Analysis, Control, and Visualization of Data Using Personal Computers and a Graphical-Based Programming Language," *Conference Proceedings of American Society of Engineering Educators (A SEE)*, Toledo, Ohio, June 21-25, 1992, pp. 1447-1453.
2. E. C. Baroth, C. Hartsough, L. Johnsen, J. McGregor, M. Powell-Meeks, A. Walsh, G. Wells, S. Chazanoff, and T. Brunzie, "A Survey of Data Acquisition and Analysis Software Tools, Part l," *evaluation Engineering Magazine*, October, 1993, pp. *54-66.*
3. E. C. Baroth, C. Hartsough, L. Johnsen, J. McGregor, M. Powell-Meeks, A. Walsh, G. Wells, S. Chazanoff, and T. Brunzie, "A Survey of Data Acquisition and Analysis Software Tools, Part 2," *Evaluation Engineering Magazine*, November, 1993, pp. 128-140.
4. *Proceedings of National Instruments User Symposium,* Austin, Texas, March 28-30, 1993.
5. J. Kodosky, J. MacCrisken, and G. Rymar, "Visual Programming Using Structured Data Flow," *Proceedings of the 1991 IEEE Workshop on Visual Languages*, Kobe, Japan, October 8-11, 1991, pp. 34-39.
6. G. Wells and E. C. Baroth, "Telemetry Monitoring and Display using LabVIEW," *Proceedings of National Instrumen is User Symposium*, Austin, Texas, March 28-30, 1993.
7. "Astro-2 Continuing Exploration of the Invisible Universe", National Aeronautics and Space Administration, Astro Physics Division, Washington D. C..

8. Dennison, E. W., Stanton, R.H. and Shimada, K., "Astros: High Performance CCD Tracker for Spacecraft," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.
9. POCC Capabilities Document, (Rev A, PCN-1) Space Shuttle Program Office, National Aeronautics and Space Administration, Washington D.C.. October, 1991.