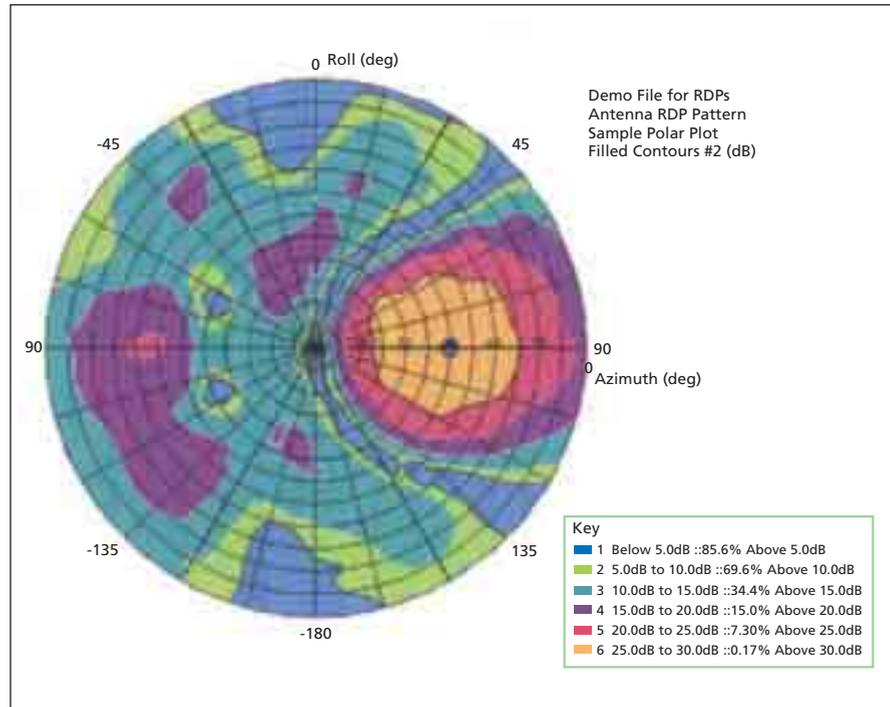


Displaying and Analyzing Antenna Radiation Patterns

Lyndon B. Johnson Space Center, Houston, Texas

Radiant Energy Display and Analysis Software Package (REDAP) is a computer program for processing antenna-radiation-pattern data that have been pre-processed by a data-collection program and stored in a spherical-coordinate format. REDAP is designed specifically for application to data generated in testing of antennas in an anechoic chamber at Johnson Space Center; parts of REDAP may be reusable for processing antenna-test data collected elsewhere. REDAP provides a graphical user interface (GUI) and executes mathematical and plotting routines on a personal computer. The routines include statistical calculations (e.g., maximum, minimum, 3-dB-falloff points, and percent coverage); addition and subtraction of offsets; multiplication and division by scaling factors; and computation of circular-polarization characteristics from linear-polarization measurement data.

REDAP can display radiation-pattern data in a variety of formats that include, for example, lines plotted on rectangular coordinates, images of radiation patterns in three-dimensional space, and contour plots on polar coordinates (see figure). All characteristics of a display, including contents of plots, coordinate axes, marks, colors, and the destination



A Sample Polar Plot illustrates one of the formats that can be generated from REDAP.

(e.g., printer or file) to which the display data are to be sent are controlled through a series of GUI dialog boxes.

This work was done by Jim Siekierski of

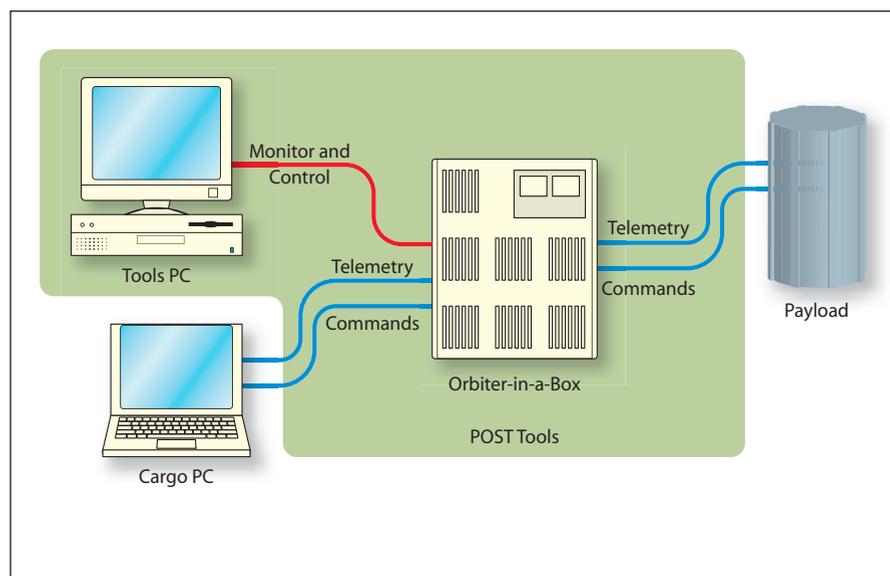
Johnson Space Center and Ash Henson of Lockheed Martin Corp. Further information is contained in a TSP (see page 1). MSC-23210-1

Payload Operations Support Team Tools

Lyndon B. Johnson Space Center, Houston, Texas

Payload Operations Support Team Tools is a software system that assists in (1) development and testing of software for payloads to be flown aboard the space shuttles and (2) training of payload customers, flight controllers, and flight crews in payload operations. POST Tools includes the following subsystems:

- The Orbiter-in-a-Box Tool is an embedded real-time operating-system model of the space-shuttle orbiter avionics. In conjunction with the Command and Data Tool described below, the Orbiter-in-a-Box Tool (see figure) enables testing of the application software of the Cargo PC, which is a computer through which a space-shuttle crew can monitor and control payloads.
- The Shuttle Mission Simulator (SMS) Model Tool is a suite of software tools that payload customers can use to cre-



An Orbiter-in-a-Box can be used to test the orbiter interface between the cargo PC and the payload.

ate mathematical models of payloads. These models will be used in the SMS to train flight controllers and flight crews on payload operations.

- The Mission Operations Tool provides a streamlined and simplified way to collect the payload-operations data to support the development of mission-operations documentation.
- The Command and Data Tool is a client-server database application program to be used to collect command and telemetry data for the Cargo PC.

This program was written by Bill Askew, Matthew Barry, Gary Burrows, Mike Casey, Joe Charles, Nicholas Downing, Monika Jain, Rebecca Leopold, Roger Luty, David McDill, Scott Mermelstein, Jon Morsics, Richard Osborne, Cindy Owens, Thomas Price, Ayman Quaddumi, Jim Thompson, and Patrick Walter of United Space Alliance, LLC; Melanie Vail of Raytheon Co.; and Richard Campbell and Mark Kelly of Systems Interface Software, Ltd. for Johnson Space Center. Further information is contained in a TSP (see page 1).

Title to this invention has been waived under the provisions of the National Aeronautics and Space Act (42 U.S.C. 2457(f)), to United Space Alliance, LLC. Inquiries concerning licenses for its commercial development should be addressed to:

*United Space Alliance, LLC
Flight Operations
600 Gemini
Houston, TX 77058-2777*

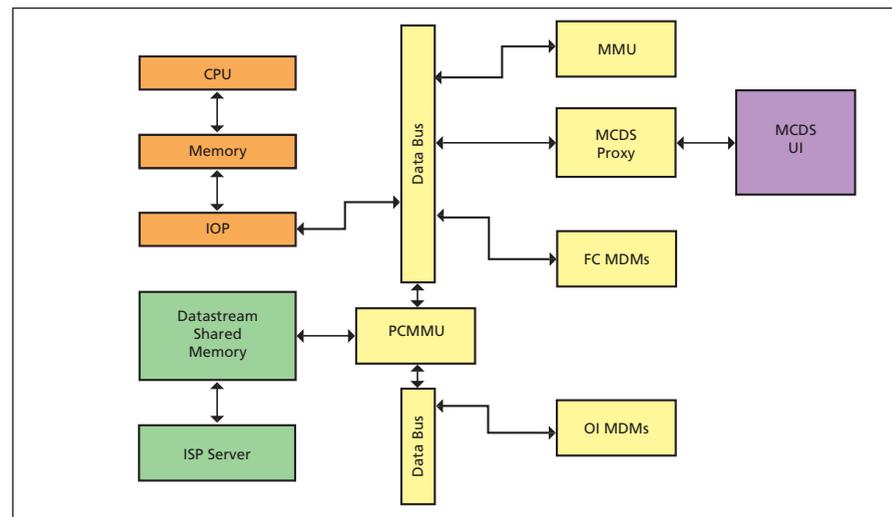
Refer to MSC-23419-1/20-1/1-1/2-1, volume and number of this NASA Tech Briefs issue, and the page number.

Space-Shuttle Emulator Software

Lyndon B. Johnson Space Center, Houston, Texas

A package of software has been developed to execute a raw binary image of the space shuttle flight software for simulation of the computational effects of operation of space shuttle avionics. This software can be run on inexpensive computer workstations. Heretofore, it was necessary to use real flight computers to perform such tests and simulations.

The package includes a program that emulates the space shuttle orbiter general-purpose computer (see figure) [consisting of a central processing unit (CPU), input/output processor (IOP), master sequence controller, and bus-control elements]; an emulator of the orbiter display electronics unit and models of the associated cathode-ray tubes, keyboards, and switch controls; computational models of the data-bus network; computational models of the multiplexer-demultiplexer components; an emulation of the pulse-code modulation master unit; an emulation of the payload data interleaver; a model of the master timing unit; a model of the mass memory unit; and a software component that ensures compatibility of telemetry and command services between the simulated space shuttle



A High-Level Schematic indicates the arrangement. (Note: CPU = central processing unit, IOP = input/output processor, ISP= information sharing protocol, PCMMU = pulse code modulation master unit, MCDS = multifunction CRT display system, MDMs = multiplexer/demultiplexers, FC = flight critical, OC = operational instrumentation, and UI = user interface.)

avionics and a mission control center. The software package is portable to several host platforms.

This program was written by Scott Arnold, Bill Askew, Matthew R. Barry, Agnes Leigh, Scott Mermelstein, James Owens, Dan Payne, Jim Pemble, John Sollinger, Hiram Thomp-

son, James C. Thompson, Patrick Walter, David Brummel and Steven P. Weismuller of United Space Alliance, LLC; and Ron Aadsen, Keith Hurley, and Chris Ruhle of Raytheon Co. for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-23289-1

Soft Real-Time PID Control on a VME Computer

NASA's Jet Propulsion Laboratory, Pasadena, California

microPID (μ PID) is a computer program for real-time proportional + integral + derivative (PID) control of a translation stage in a Fourier-transform ultraviolet spectrometer. μ PID implements a PID control loop over a position profile at sampling rate of 8 kHz (sampling period

125 μ s). The software runs in a stripped-down Linux operating system on a VersaModule Eurocard (VME) computer operating in real-time priority queue using an embedded controller, a 16-bit digital-to-analog converter (D/A) board, and a laser-positioning board (LPB).

μ PID consists of three main parts: (1) VME device-driver routines, (2) software that administers a custom protocol for serial communication with a control computer, and (3) a loop section that obtains the current position from an LPB-driver routine, calculates