processed by other software for viewing and analyzing the trajectory.

DEBRIS supplants a prior debris-tracking code that took ≈15 minutes to calculate a single particle trajectory: DEBRIS can calculate 1,000 trajectories in ≈20 seconds on a desktop computer. Other improvements over the prior code include adaptive time-stepping to ensure accuracy, forcing at least one step per grid cell to ensure resolution of all CFD-resolved flow features, ability to simulate rebound of debris from surfaces, extensive error checking, a built-in suite of test cases, and dynamic allocation of memory.

This program was written by Phillip C. Stuart of Johnson Space Center and Stuart E. Rogers of Ames Research Center. Further information is contained in a TSP (see page 1). *MSC-23945-1*

Estimating Thruster Impulses From IMU and Doppler Data

NASA's Jet Propulsion Laboratory, Pasadena, California

A computer program implements a thrust impulse measurement (TIM) filter, which processes data on changes in velocity and attitude of a spacecraft to estimate the small impulsive forces and torques exerted by the thrusters of the spacecraft reaction control system (RCS). The velocity-change data are obtained from line-of-sight-velocity data from Doppler measurements made from the Earth. The attitude-change data are the telemetered from an inertial measurement unit (IMU) aboard the spacecraft.

The TIM filter estimates the three-axis thrust vector for each RCS thruster, thereby enabling reduction of cumulative navigation error attributable to inaccurate prediction of thrust vectors. The filter has been augmented with a simple mathematical model to compensate for large temperature fluctuations in the spacecraft thruster catalyst bed in order to estimate thrust more accurately at deadbanding "cold-firing" levels. Also, rigorous consider-covariance estimation is applied in the TIM to account for the expected uncertainty in the moment of inertia and the location of the center of gravity of the spacecraft. The TIM filter was built with, and depends upon, a sigma-point consider-filter algorithm implemented in a Python-language computer program.

This program was written by Michael E. Lisano and Gerhard L. Kruizinga of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-45825.

Oxygen Generation System Laptop Bus Controller Flight Software

Lyndon B. Johnson Space Center, Houston, Texas

The Oxygen Generation System Laptop Bus Controller Flight Software was developed to allow the International Space Station (ISS) program to activate specific components of the Oxygen Generation System (OGS) to perform a checkout of key hardware operation in a microgravity environment, as well as to perform preventative maintenance operations of system valves during a long period of what would otherwise be hardware dormancy. The software provides direct connectivity to the OGS Firmware Controller with pre-programmed tasks operated by on-orbit astronauts to exercise OGS valves and motors. The software is used to manipulate the pump, separator, and valves to alleviate the concerns of hardware problems due to long-term inactivity and to allow for operational verification of microgravity-sensitive components early enough so that, if problems are found, they can be addressed before the hardware is required for operation on-orbit.

The decision was made to use existing on-orbit IBM ThinkPad A31p laptops and MIL-STD-1553B interface cards as the hardware configuration. The software at the time of this reporting was developed and tested for use under the Windows 2000 Professional operating system to ensure compatibility with the existing on-orbit computer systems.

This program was written by Chad Rowe and Donna Panter for Johnson Space Center. Further information is contained in a TSP (see page 1). *MSC-24316-1*

Port-O-Sim Object Simulation Application

Goddard Space Flight Center, Greenbelt, Maryland

Port-O-Sim is a software application that supports engineering modeling and simulation of launch-range systems and subsystems, as well as the vehicles that operate on them. It is flexible, distributed, object-oriented, and real-time. A scripting language is used to configure an array of simulation objects and link them together. The script is contained in a text file, but executed and controlled using a graphical user interface.

A set of modules is defined, each with input variables, output variables, and settings. These engineering models can be either linked to each other or run as standalone. The settings can be modified during execution.

Since 2001, this application has been used for pre-mission failure mode training for many Range Safety Scenarios. It contains range asset link analysis, develops look-angle data, supports sky-screen site selection, drives GPS (Global Positioning System) and IMU (Inertial Measurement Unit) simulators, and can support conceptual design efforts for multiple flight programs with its capacity for rapid six-degrees-of-freedom model development. Due to the assembly of various object types into one application, the application is applicable across a wide variety of launch range problem domains.

This work was done by Raymond J Lanzi of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-15571-1

Monitoring and Controlling an Underwater Robotic Arm

Lyndon B. Johnson Space Center, Houston, Texas

The SSRMS Module 1 software is part of a system for monitoring an adaptive, closed-loop control of the motions of a robotic arm in NASA's Neutral Buoyancy Laboratory, where buoyancy in a pool of water is used to simulate the weightlessness of outer space. This software is so named because the robot arm is a replica of the Space Shuttle Remote Manipulator System (SSRMS).

This software is distributed, running on remote joint processors (RJPs), each of which is mounted in a hydraulic actuator comprising the joint of the robotic arm and communicating with a poolside processor denoted the Direct Control Rack (DCR). Each RJP executes the feedback joint-motion control algorithm for its joint and communicates with the DCR. The DCR receives joint-angular-velocity commands either locally from an operator or remotely from computers that simulate the flight like SSRMS and perform coordinated motion calculations based on hand-controller inputs. The received commands are checked for validity before they are transmitted to the RJPs. The DCR software generates a display of the statuses of the RJPs for the DCR operator and can shut down the hydraulic pump when excessive joint-angle error or failure of a RJP is detected.

This work was done by John Haas and Brian Keith Todd of Johnson Space Center, Larry Woodcock and Fred M. Robinson of Oceaneering Space Systems, and Thomas (Jay) Costales of Raytheon Co. Further information is contained in a TSP (see page 1). MSC-24165-1

Digital Camera Control for Faster Inspection

Lyndon B. Johnson Space Center, Houston, Texas

Digital Camera Control Software (DCCS) is a computer program for controlling a boom and a boom-mounted camera used to inspect the external surface of a space shuttle in orbit around the Earth. Running in a laptop computer in the space-shuttle crew cabin, DCCS commands integrated displays and controls. By means of a simple one-button command, a crewmember can view low- resolution images to quickly spot problem areas and can then cause a rapid transition to high- resolution images. The crewmember can command that camera settings apply to a specific small area of interest within the field of view of the camera so as to maximize image quality within that area.

DCCS also provides critical high-resolution images to a ground screening team, which analyzes the images to assess damage (if any); in so doing, DCCS enables the team to clear initially suspect areas more quickly than would otherwise be possible and further saves time by minimizing the probability of re-imaging of areas already inspected. On the basis of experience with a previous version (2.0) of the software, the present version (3.0) incorporates a number of advanced imaging features that optimize crewmember capability and efficiency.

This program was written by Katharine Brown, James D. Siekierski, Mark L. Mangieri, Kent Dekome, John Cobarruvias, and Perry J. Piplani of Johnson Space Center and Joel Busa of the Draper Laboratory. Further information is contained in a TSP (see page 1). MSC-24319-1/168-1

Reaction Wheel Disturbance Model Extraction Software — RWDMES

Goddard Space Flight Center, Greenbelt, Maryland

The RMDMES is a tool for modeling the disturbances imparted on spacecraft by spinning reaction wheels. Reaction wheels are usually the largest disturbance source on a precision pointing spacecraft, and can be the dominating source of pointing error. Accurate knowledge of the disturbance environment is critical to accurate prediction of the pointing performance. In the past, it has been difficult to extract an accurate wheel disturbance model since the forcing mechanisms are difficult to model physically, and the forcing amplitudes are filtered by the dynamics of the reaction wheel. RDMES captures the wheel-induced disturbances using a hybrid physical/empirical model that is extracted directly from measured forcing data.

The empirical models capture the tonal forces that occur at harmonics of the spin rate, and the broadband forces that arise from random effects. The em-