

uses COTS (commercial off-the-shelf) software where possible.

This program was written by Francisco Delgado of Johnson Space Center. For further information, contact the Johnson Technology Transfer Office at (281) 483-3809. MSC-23490

Software for Engineering Simulations of a Spacecraft

Spacecraft Engineering Simulation II (SES II) is a C-language computer program for simulating diverse aspects of operation of a spacecraft characterized by either three or six degrees of freedom. A functional model in SES can include a trajectory flight plan; a submodel of a flight computer running navigational and flight-control software; and submodels of the environment, the dynamics of the spacecraft, and sensor inputs and outputs. SES II features a modular, object-oriented programming style. SES II supports event-based simulations, which, in turn, create an easily adaptable simulation environment in which many different types of trajectories can be simulated by use of the same software. The simulation output consists largely of flight data. SES II can be used to perform optimization and Monte Carlo dispersion simulations. It can also be used to perform simulations for multiple spacecraft. In addition to its generic simulation capabilities, SES offers special capabilities for space-shuttle simulations: for this purpose, it incorporates submodels of the space-shuttle dynamics and a C-language version of the guidance, navigation, and control components of the space-shuttle flight software.

This program was written by Kirk Shireman and Gene McSwain of Johnson Space Center, and Bernell McCormick and Panayiotis Fardelos of the Boeing Co. For further information, contact the Johnson Technology Transfer Office at (281) 483-3809. MSC-23537

LabVIEW Interface for PCI-SpaceWire Interface Card

This software provides a LabView interface to the NT drivers for the PCI-SpaceWire card, which is a peripheral component interface (PCI) bus interface that conforms to the IEEE-1355/SpaceWire standard. As SpaceWire grows in popularity, the ability to use SpaceWire links within LabVIEW will be important to electronic ground support equipment vendors. In addition, there is a need for a high-level LabVIEW interface to the low-level device-driver software supplied with the card.

The LabVIEW virtual instrument (VI) provides graphical interfaces to support all (1) SpaceWire link functions, including message handling and routing; (2) monitoring as a passive “tap” using specialized hardware; and (3) low-level access to satellite mission-control subsystem functions. The software is supplied in a zip file that contains LabVIEW VI files, which provide various functions of the PCI-SpaceWire card, as well as higher-link-level functions. The VIs are suitably named according to the matching function names in the driver manual. A number of test programs also are provided to exercise various functions.

This work was done by James Lux, Frank Loya, and Alex Bachmann of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-35207.

Path Following With Slip Compensation for a Mars Rover

A software system for autonomous operation of a Mars rover is composed of several key algorithms that enable the rover to accurately follow a designated path, compensate for slippage of its wheels on terrain, and reach intended goals. The techniques implemented by the algorithms are visual odometry, full vehicle kinematics, a Kalman filter, and path following with slip compensation. The visual-odometry algorithm tracks distinctive scene features in stereo imagery to estimate rover motion between successively acquired stereo image pairs, by use of a maximum-likelihood motion-estimation algorithm. The full-vehicle kinematics algorithm estimates motion, with a no-slip assumption, from measured wheel rates, steering angles, and angles of rockers and bogies in the rover suspension system. The Kalman filter merges data from an inertial measurement unit (IMU) and the visual-odometry algorithm. The merged estimate is then compared to the kinematic estimate to determine whether and how much slippage has occurred. The kinematic estimate is used to complement the Kalman-filter estimate if no statistically significant slippage has occurred. If slippage has occurred, then a slip vector is calculated by subtracting the current Kalman filter estimate from the kinematic estimate. This slip vector is then used, in conjunction with the inverse

kinematics, to determine the wheel velocities and steering angles needed to compensate for slip and follow the desired path.

This work was done by Daniel Helmick, Yang Cheng, Daniel Clouse, Larry Matthies, and Stergios Roumeliotis of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40703.

International Space Station Electric Power System Performance Code — SPACE

The System Power Analysis for Capability Evaluation (SPACE) software analyzes and predicts the minute-by-minute state of the International Space Station (ISS) electrical power system (EPS) for upcoming missions as well as EPS power generation capacity as a function of ISS configuration and orbital conditions. In order to complete the Certification of Flight Readiness (CoFR) process — in which the mission is certified for flight — each ISS System must thoroughly assess every proposed mission to verify that the system will support the planned mission operations; SPACE is the sole tool used to conduct these assessments for the power system capability. SPACE is an integrated power system model that incorporates a variety of modules tied together with integration routines and graphical output. The modules include orbit mechanics, solar array pointing/shadowing/thermal and electrical, battery performance, and power management and distribution performance. These modules are tightly integrated within a flexible architecture featuring data-file-driven configurations, source- or load-driven operation, and event scripting. SPACE also predicts the amount of power available for a given system configuration, spacecraft orientation, solar-array-pointing conditions, orbit, and the like. In the source-driven mode, the model must assure that energy balance is achieved, meaning that energy removed from the batteries must be restored (or balanced) each and every orbit. This entails an optimization scheme to ensure that energy balance is maintained without violating any other constraints. In the load-driven mode, SPACE determines whether a given distributed, time-varying electrical load profile can be supported by the power system and will determine whether the system stays in energy balance. Load-driven

mode also is used to assess each space shuttle mission to ISS, in support of the CoFR process.

This work was done by Jeffrey Hojnicky, David McKissock, James Fincannon, Robert Green, Thomas Kerlake, Ann Delleur, Jeffrey Follo, Jeffrey Trudell, David J. Hoffman, Anthony Jannette, and Carlos Rodriguez for Glenn Research Center. Further information is contained in a TSP (see page 1).

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17067.

2 Software for Automation of Real-Time Agents, Version 2

Version 2 of Closed Loop Execution and Recovery (CLEaR) has been developed. The previous version was reported in "Software for Automation of Real-Time Agents" (NPO-21040), *NASA Tech Briefs*, Vol. 26, No. 7 (July 2002), page 34. To recapitulate: CLEaR is an artificial intelligence computer program for use in planning and execution of actions of autonomous agents, including, for example, Deep Space Network (DSN) antenna ground stations, robotic exploratory ground vehicles (rovers), robotic aircraft (UAVs), and robotic spacecraft. CLEaR automates the generation and execution of command sequences, monitoring the sequence execution, and modifying the command sequence in response to execution deviations and failures as well as new goals for the agent to achieve. The development of CLEaR has focused on the unification of planning and execution to increase the ability of the autonomous agent to perform under tight resource and time constraints coupled with uncertainty in how much of resources and time will be required to perform a task. This unification is realized by extending the traditional three-tier robotic control architecture by increasing the interaction between the software components that perform deliberation and reactive functions. The increase in interaction reduces the need to replan, enables earlier detection of the need to replan, and enables replanning to occur before an agent enters a state of failure.

This program was written by Forest Fisher, Tara Estlin, Daniel Gaines, Steve Schaffer, Caroline Chouinard, Barbara Engelhardt, Colette Wilklow, Darren Mutz, Russell Knight, Gregg Rabideau, and Steve Chien of Caltech, and Reid Simmons and David Apfelbaum of CMU for NASA's Jet Propulsion

Laboratory. *Further information is contained in a TSP (see page 1).*

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-30745.

2 Software for Optimizing Plans Involving Interdependent Goals

A computer program enables construction and optimization of plans for activities that are directed toward achievement of goals that are interdependent. Goal interdependence is defined as the achievement of one or more goals affecting the desirability or priority of achieving one or more other goals. This program is overlaid on the Automated Scheduling and Planning Environment (ASPEN) software system, aspects of which have been described in a number of prior *NASA Tech Briefs* articles. Unlike other known or related planning programs, this program considers interdependences among goals that can change between problems and provides a language for easily specifying such dependences. Specifications of the interdependences can be formulated dynamically and provided to the associated planning software as part of the goal input. Then an optimization algorithm provided by this program enables the planning software to reason about the interdependences and incorporate them into an overall objective function that it uses to rate the quality of a plan under construction and to direct its optimization search. In tests on a series of problems of planning geological experiments by a team of instrumented robotic vehicles (rovers) on new terrain, this program was found to enhance plan quality.

This program was written by Tara Estlin, Daniel Gaines, and Gregg Rabideau of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-30735.

2 Computing Gravitational Fields of Finite-Sized Bodies

A computer program utilizes the classical theory of gravitation, implemented by means of the finite-element method, to calculate the near gravitational fields of bodies of arbitrary size, shape, and

mass distribution. The program was developed for application to a spacecraft and to floating proof masses and associated equipment carried by the spacecraft for detecting gravitational waves. The program can calculate steady or time-dependent gravitational forces, moments, and gradients thereof. Bodies external to a proof mass can be moving around the proof mass and/or deformed under thermoelastic loads. An arbitrarily shaped proof mass is represented by a collection of parallelepiped elements. The gravitational force and moment acting on each parallelepiped element of a proof mass, including those attributable to the self-gravitational field of the proof mass, are computed exactly from the closed-form equation for the gravitational potential of a parallelepiped. The gravitational field of an arbitrary distribution of mass external to a proof mass can be calculated either by summing the fields of suitably many point masses or by higher-order Gauss-Legendre integration over all elements surrounding the proof mass that are part of a finite-element mesh. This computer program is compatible with more general finite-element codes, such as NASTRAN, because it is configured to read a generic input data file, containing the detailed description of the finite-element mesh.

This program was written by Marco Quadrelli of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40651.

2 Custom Sky-Image Mosaics From NASA's Information Power Grid

yourSkyG is the second generation of the software described in "yourSky: Custom Sky-Image Mosaics via the Internet" (NPO-30556), *NASA Tech Briefs*, Vol. 27, No. 6 (June 2003), page 45. Like its predecessor, yourSkyG supplies custom astronomical image mosaics of sky regions specified by requesters using client computers connected to the Internet. Whereas yourSky constructs mosaics on a local multiprocessor system, yourSkyG performs the computations on NASA's Information Power Grid (IPG), which is capable of performing much larger mosaicking tasks. (The IPG is high-performance computation and data grid that integrates geographically distributed