Software

Automated Design of Restraint Layer of an Inflatable Vessel

A Mathcad computer program largely automates the design and analysis of the restraint layer (the primary load-bearing layer) of an inflatable vessel that consists of one or more sections having cylindrical, toroidal, and/or spherical shape(s). A restraint layer typically comprises webbing in the form of multiple straps. The design task includes choosing indexing locations along the straps, computing the load at every location in each strap, computing the resulting stretch at each location, and computing the amount of undersizing required of each strap so that, once the vessel is inflated and the straps thus stretched, the vessel can be expected to assume the desired shape.

Prior to the development of this program, the design task was performed by use of a difficult-to-use spreadsheet program that required manual addition of rows and columns depending on the numbers of strap rows and columns of a given design. In contrast, this program is completely parametric and includes logic that automatically adds or deletes rows and columns as needed. With minimal input from the user, this program automatically computes indexing locations, strap lengths, undersizing requirements, and all design data required to produce detailed drawings and assembly procedures. It also generates textual comments that help the user understand the calculations.

This program was written by Gary Spexarth of **Johnson Space Center**. *Further information is contained in a TSP (see page 1).*

This invention is owned by NASA, and a patent application has been filed. Inquiries concerning nonexclusive or exclusive license for its commercial development should be addressed to the Patent Counsel, Johnson Space Center, (281) 483-0837. Refer to MSC-23906.

TMS for Instantiating a Knowledge Base With Incomplete Data

A computer program that belongs to the class known among software experts as output truth-maintenance-systems (output TMSs) has been devised as one of a number of software tools for reducing the size of the knowledge base that must be searched during execution of artificial-intelligence software of the rule-based inference-engine type in a case in which data are missing. This program determines whether the consequences of activation of two or more rules can be combined without causing a logical inconsistency. For example, in a case involving hypothetical scenarios that could lead to turning a given device on or off, the program determines whether a scenario involving a given combination of rules could lead to turning the device both on and off at the same time, in which case that combination of rules would not be included in the scenario.

This program was written by Mark James 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 (626) 395-2322. Refer to NPO-42710.

Simulating Flights of Future Launch Vehicles and Spacecraft

Marshall Aerospace Vehicle Representation in C (MAVERIC) is a computer program for generic, low-to-high-fidelity simulation of the flight(s) of one or more launch vehicle(s) or spacecraft. MAVERIC is designed to accommodate multi-staged vehicles, powered serially or in parallel, with multiple engines, tanks, and cargo elements. Engines can be of jet or conventional rocket types, using either liquid or solid propellants.

MAVERIC includes generic subsystem software models for propulsion systems, mass properties, reaction control systems, aerodynamic properties, guidance systems, and navigation systems. Simulations can be started at points other than liftoff. Also included are guidance-system software models that accommodate the ascent, orbit, coasting, deorbiting, entry, terminal-area-energy-management, approach, and landing phases of flight.

Options to use different wind profiles and atmospheres are included. A Monte Carlo capability is provided for modeling dispersions associated with atmospheric effects (including winds), propulsion, navigation, aerodynamics, and mass properties. Failures of engines and other subsystems can be modeled. The program is written in the C programming language, which makes it possible for the program to have high degrees of modularity, reusability, and maintainability, thereby also facilitating modification for modeling new vehicles.

This program was written by James W. McCarter of **Marshall Space Flight Center**. *Further information is contained in a TSP (see page 1).*

This invention is owned by NASA, and a patent application has been filed. For further information, contact Sammy Nabors, MSFC Commercialization Assistance Lead, at sammy.a.nabors@nasa.gov. Refer to MFS-31989-1.

Control Code for Bearingless Switched-Reluctance Motor

A computer program has been devised for controlling a machine that is an integral combination of magnetic bearings and a switched-reluctance motor. The motor contains an eight-pole stator and a hybrid rotor, which has both (1) a circular lamination stack for levitation and (2) a six-pole lamination stack for rotation. The program computes drive and levitation currents for the stator windings with real-time feedback control. During normal operation, two of the four pairs of opposing stator poles (each pair at right angles to the other pair) levitate the rotor. The remaining two pairs of stator poles exert torque on the six-pole rotor lamination stack to produce rotation. This version is executable in a control-loop time of 40 μs on a Pentium (or equivalent) processor that operates at a clock speed of 400 MHz. The program can be expanded, by addition of logic blocks, to enable control of position along additional axes. The code enables adjustment of operational parameters (e.g., motor speed and stiffness, and damping parameters of magnetic bearings) through computer keyboard key presses.

This program was written by Carlos R. Morrison of **Glenn Research Center**. *Further information is contained in a TSP (see page 1).*