

Framework is also a system-level "health"-management software system that (1) receives data from performance-monitoring and built-in-test sensors and from other prognostic software and (2) processes the received data to derive a diagnosis and a prognosis for a weapon system. This software relates the diagnostic and prognostic information to the overall health of the system, to the ability of the system to perform specific missions, and to needed maintenance actions and maintenance resources. In the development of the Prognostics Framework, effort was focused primarily on extending previously developed model-based diagnostic-reasoning software to add prognostic reasoning capabilities, including capabilities to perform statistical analyses and to utilize information pertaining to deterioration of parts, failure modes, time sensitivity of measured values, mission criticality, historical data, and trends in measurement data. As thus extended, the software offers an overall health-monitoring capability.

*This program was written by Mary Nolan, Rebecca Catania, and Gregory deMare of Giordano Automation Corp. for Marshall Space Flight Center. Further information is contained in a TSP (see page 1).*  
MFS-31644

---

## 🔍 Training Software in Artificial-Intelligence Computing Techniques

The Artificial Intelligence (AI) Toolkit is a computer program for training scientists, engineers, and university students in three soft-computing techniques (fuzzy logic, neural networks, and genetic algorithms) used in artificial-intelligence applications. The program promotes an easily understandable tutorial interface, including an interactive graphical component through which the user can gain hands-on experience in soft-computing techniques applied to realistic example problems. The tutorial provides step-by-step instructions on the workings of soft-computing technology, whereas the hands-on examples allow interaction and reinforcement of the techniques explained throughout the tutorial. In the fuzzy-logic example, a user can interact with a robot and an obstacle course to verify how fuzzy logic is used to command a rover traverse from an arbitrary start to the goal location. For the genetic algorithm example, the problem is to determine the minimum-length path for visit-

ing a user-chosen set of planets in the solar system. For the neural-network example, the problem is to decide, on the basis of input data on physical characteristics, whether a person is a man, woman, or child. The AI Toolkit is compatible with the Windows 95,98, ME, NT 4.0, 2000, and XP operating systems. A computer having a processor speed of at least 300 MHz, and random-access memory of at least 56MB is recommended for optimal performance. The program can be run on a slower computer having less memory, but some functions may not be executed properly.

*This program was written by Ayanna Howard, Eric Rogstad, and Eugene Chalfant 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40496.*

---

## 🔍 APGEN Version 5.0

Activity Plan Generator (APGEN), now at version 5.0, is a computer program that assists in generating an integrated plan of activities for a spacecraft mission that does not oversubscribe spacecraft and ground resources. APGEN generates an interactive display, through which the user can easily create or modify the plan. The display summarizes the plan by means of a time line, whereon each activity is represented by a bar stretched between its beginning and ending times. Activities can be added, deleted, and modified via simple mouse and keyboard actions. The use of resources can be viewed on resource graphs. Resource and activity constraints can be checked. Types of activities, resources, and constraints are defined by simple text files, which the user can modify. In one of two modes of operation, APGEN acts as a planning expert assistant, displaying the plan and identifying problems in the plan. The user is in charge of creating and modifying the plan. In the other mode, APGEN automatically creates a plan that does not oversubscribe resources. The user can then manually modify the plan. APGEN is designed to interact with other software that generates sequences of timed commands for implementing details of planned activities.

*This program was written by Pierre Maldague, Dennis Page, and Adam Chase 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40520.*

---

## 🔍 Single-Command Approach and Instrument Placement by a Robot on a Target

AUTOAPPROACH is a computer program that enables a mobile robot to approach a target autonomously, starting from a distance of as much as 10 m, in response to a single command. AUTOAPPROACH is used in conjunction with (1) software that analyzes images acquired by stereoscopic cameras aboard the robot and (2) navigation and path-planning software that utilizes odometer readings along with the output of the image-analysis software. Intended originally for application to an instrumented, wheeled robot (rover) in scientific exploration of Mars, AUTOAPPROACH could be adapted to terrestrial applications, notably including the robotic removal of land mines and other unexploded ordnance. A human operator generates the approach command by selecting the target in images acquired by the robot cameras. The approach path consists of multiple legs. Feature points are derived from images that contain the target and are thereafter tracked to correct odometric errors and iteratively refine estimates of the position and orientation of the robot relative to the target on successive legs. The approach is terminated when the robot attains the position and orientation required for placing a scientific instrument at the target. The workspace of the robot arm is then autonomously checked for self/terrain collisions prior to the deployment of the scientific instrument onto the target.

*This program was written by Terrance Huntsberger and Yang Cheng of NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).*

*This software is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30529.*

---

## 🔍 Three-Dimensional Audio Client Library

The Three-Dimensional Audio Client Library (3DAudio library) is a group of software routines written to facilitate development of both stand-alone (audio only) and immersive virtual-reality application pro-

grams that utilize three-dimensional audio displays. The library is intended to enable the development of three-dimensional audio client application programs by use of a code base common to multiple audio server computers. The 3DAudio library calls vendor-specific audio client libraries and currently supports the AuSIM Gold-Server and Lake Huron audio servers.

3DAudio library routines contain common functions for (1) initiation and termination of a client/audio server session, (2) configuration-file input, (3) positioning functions, (4) coordinate transformations, (5) audio transport functions, (6) rendering functions, (7) debugging functions, and (8) event-list-sequencing functions. The 3DAudio software is written in the C++

programming language and currently operates under the Linux, IRIX, and Windows operating systems.

*This program was written by Stephen A. Rizzi of **Langley Research Center**. For further information, contact the Intellectual Property Team at (757) 864-3521.  
LAR-16690-1*