present as an impurity in the propellants and/or these can form in the boundary layer as a result of interaction of the hot plume with the atmosphere during the ground testing of engines. Ten additional electronic band systems of these five molecules have been included into the code. A comprehensive literature search was conducted to obtain the most accurate values for the molecular and the spectral parameters, including Franck-Cordon factors and electronic transition moments for all ten band systems.

For each elemental transition in the RPSSC, six spectral parameters — Doppler broadened line width at half-height, pressure-broadened line width at half-height, electronic multiplicity of the upper state, electronic term energy of the upper state, Einstein transition probability coefficient, and the atomic line center — are required. Input files have been created for ten elements of Ni, Fe, Cr, Co, Cu, Ca, Mn, Al, Ag, and Pd, which retain only relatively moderate to strong transitions in 300 to 430 nm spectral range for each element. The number of transitions in the input files is 68 for Ni; 148 for Fe; 6 for Cr; 87 for Co; 1 for Ca; 3 for Mn; 2 each for Cu, Al, and Ag; and 11 for Pd.

This work was done by Gopal D. Tejwani of Jacobs Technology, Inc. for Stennis Space Center.

Inquiries concerning the technology should be addressed to the Intellectual Property Manager, Stennis Space Center; (228) 688-1929. Refer to SSC-00281, volume and number of this NASA Tech Briefs *issue, and the page number.*

Research on Spoken Dialogue Systems

Human verbal interaction with complex information sources.

Ames Research Center, Moffett Field, California

Research in the field of spoken dialogue systems has been performed with the goal of making such systems more robust and easier to use in demanding situations. The term “spoken dialogue systems” signifies unified software systems containing speech-recognition, speech-synthesis, dialogue management, and ancillary components that enable human users to communicate, using natural spoken language or nearly natural prescribed spoken language, with other software systems that provide information and/or services. The research is proceeding on several fronts: recognition of speech signals, syntactic and semantic parsing, language modeling, discourse analysis, and contact modeling.

Many of the advances made thus far in this research have been incorporated into a voice-enabled procedure-browser and reader, called Clarissa, that has been tested aboard the International Space Station. [A procedure-browser and reader is essentially a software version of an instruction manual that may describe one or more possibly complex procedure(s).] Major problems that have been addressed in developing Clarissa include creating voice-navigable versions of formal procedure documents, grammar-based speech recognition, methods for accurate detection of user’s speech directed toward a listener other than Clarissa based on grammar filtering or support vector machines, and robust, side-effect-free dialogue management for enabling undoing, correction, and/or confirmation of steps of a procedure.

Clarissa enables the user to navigate a complex procedure using only spoken input and output, making it unnecessary for the user to shift visual attention from the task at hand to a paper instruction manual or to an equivalent document displayed on a computer screen. Clarissa also provides a graphical user interface (GUI) for optional visual display of information. Clarissa has a vocabulary of about 260 words and supports about 75 different commands, including commands for reading steps of the procedure, scrolling forward or backward in the procedure, moving to an arbitrary new step, reviewing non-current steps, adding and removing voice notes, displaying pictures, setting and canceling alarms and timers, requiring challenges to verify critical commands, and querying the system as to status of the procedure.

Clarissa includes the following main software modules:

- Speech Processor — Performs low-level speech-recognition (input) and speech-synthesis (output) functions.
- Semantic Analyzer — Converts output from the speech processor into an abstract dialogue move.
- Response Filter — Decides whether to accept or reject the spoken input from the user.
- Dialogue Manager — Converts abstract dialogue moves into abstract dialogue actions, and maintains knowledge of both the context of the discourse and the progress through the procedure.
- Output Manager — Accepts abstract dialogue actions from the Dialogue Manager and converts them into lists of procedure calls that result in concrete system responses, which can include spoken output, requests for display of visual output on the GUI, or sending dialogue moves back to the Dialogue Manager.
- GUI Module — Mediates conventional keyboard and screen-based interaction with the user and accepts display requests from the Output Manager. This module can also convert keyboard input from the user into dialogue moves, which are sent to the Dialogue Manager.

Another accomplishment of this research has been the development of a targeted-help module that is highly portable in that it can be added to a spoken dialogue system, with minimal application-specific modifications, to make the spoken dialogue system more robust. The targeted-help module is intended, more specifically, for incorporation into a spoken dialogue system in which, as in Clarissa, there is a prescribed spoken language containing a limited number of words. The purpose served by the targeted-help module is to assist an untrained user to learn the prescribed language by providing help messages in response to out-of-coverage users’ utterances (that is, users’ utterances outside the prescribed language). These messages can be much more informative than “Sorry, I didn’t understand” and variants thereof generated by older, less-capable spoken dialogue systems.

The targeted-help module includes two submodules that run simultaneously: a grammar-based recognizer and a statistical language model (SLM). When the grammar-based recognizer succeeds, the ordinarily-less-accurate hypothesis generated by the SLM recognizer is not used. When the grammar-based recognizer fails and the SLM recognizer produces a recognition hypothesis, the SLM output is processed to generate a message that tells the user what was recognized as having been uttered, a diagnosis of what was problematic about the recognized utterance, and a related in-coverage example. The in-coverage example is intended to encourage alignment between the user's utterances and the prescribed language.

This work was done by Gregory Aist and James Hieronymus of the Research Institute for Advanced Computer Science; John Dowding and Beth Ann Hockey of the University of California, Santa Cruz; Manny Rayner of the International Computer Science Institute; Nikos Chatzichrisafis of the University of Geneva; Kim Farrell of QSS; and Jean-Michel Renders of Xerox Research Center Europe for Ames Research Center. Further information is contained in a TSP (see page 1). ARC-14610-1

Injecting Errors for Testing Built-in Test Software

Lyndon B. Johnson Space Center, Houston, Texas

Two algorithms have been conceived to enable automated, thorough testing of Built-in test (BIT) software. The first algorithm applies to BIT routines that define pass/fail criteria based on values of data read from such hardware devices as memories, input ports, or registers. This algorithm simulates effects of errors in a device under test by (1) intercepting data from the device and (2) performing AND operations between the data and the data mask specific to the device. This operation yields values not expected by the BIT routine. This algorithm entails very small, permanent instrumentation of the software under test (SUT) for performing the AND operations.

The second algorithm applies to BIT programs that provide services to users' application programs via commands or callable interfaces and requires a capability for test-driver software to read and write the memory used in execution of the SUT. This algorithm identifies all SUT code execution addresses where errors are to be injected, then temporarily replaces the code at those addresses with small test code sequences to inject latent severe errors, then determines whether, as desired, the SUT detects the errors and recovers.

This work was done by Thomas K. Gender and James Chow of Honeywell, Inc., for Johnson Space Center.

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

Satellite Systems Operation
Honeywell, Inc.
P.O. Box 52199
Phoenix, AZ 85072-2199

Refer to MSC-23463-1/4-1, volume and number of this NASA Tech Briefs *issue, and the page number.*

Guidance and Control System for a Satellite Constellation

Goddard Space Flight Center, Greenbelt, Maryland

A distributed guidance and control algorithm was developed for a constellation of satellites. The system repositions satellites as required, regulates satellites to desired orbits, and prevents collisions.

1. Optimal methods are used to compute nominal transfers from orbit to orbit.
2. Satellites are regulated to maintain the desired orbits once the transfers are complete.
3. A simulator is used to predict potential collisions or near-misses.
4. Each satellite computes perturbations to its controls so as to increase any unacceptable distances of nearest approach to other objects.
   a. The avoidance problem is recast in a distributed and locally-linear form to arrive at a tractable solution.
   b. Plant matrix values are approximated via simulation at each time step.
   c. The Linear Quadratic Gaussian (LQG) method is used to compute perturbations to the controls that will result in increased miss distances.
5. Once all danger is passed, the satellites return to their original orbits, all the while avoiding each other as above.
6. The delta-Vs are reasonable. The controller begins maneuvers as soon as practical to minimize delta-V.
7. Despite the inclusion of trajectory simulations within the control loop, the algorithm is sufficiently fast for available satellite computer hardware.
8. The required measurement accuracies are within the capabilities of modern inertial measurement devices and modern positioning devices.

This work was done by Jonathan Lamar Bryson, Chadwick James Cox, Paul Richard Mays, James Christian Neidhoefer, and Richard Ephrain Sacks of Accurate Automation Corp. for Goddard Space Flight Center.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Accurate Automation Corp.
7001 Shallowford Road
Chattanooga, TN 37421

Refer to GSC-14990-1, volume and number of this NASA Tech Briefs *issue, and the page number.*