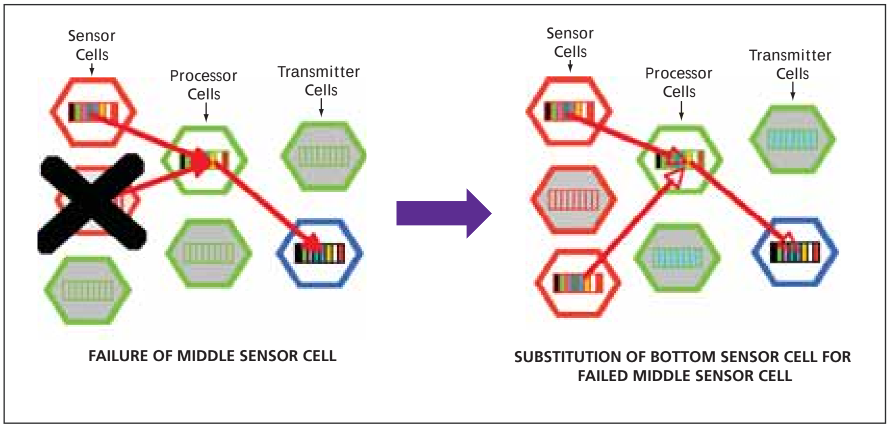

This **Simulated Prototype System** contained three sensor cells (the lowest one of which one was initially a spare), two general-purpose processor cells, and two radio-transmitter cells. Failure of the middle sensor cell initiated a biomorphic process that placed the spare sensor cell into service.

A multi-agent, organismlike computing system would be a single entity built from agents or cells. Each agent or cell would be a discrete hardware processing unit that would include a data processor with local memory, an internal clock, and a suite of communication equipment capable of both local line-of-sight communications and global broadcast communications. Some cells, denoted specialist cells, could contain such additional hardware as sensors and emitters. Each cell would be independent in the sense that there would be no global clock, no global (shared) memory, no pre-assigned cell identifiers, no pre-defined network topology, and no centralized brain or control structure. Like each cell in a living organism, each agent or cell of the computing system would contain a full description of the system encoded as genes, but in this case, the genes would be components of a software genome.

Although the cells would be independent in the sense described above, they would be tightly coupled and logically interdependent in that they would exchange information and respond accordingly. The software genome would program the system at two distinct levels: The first-level programs would describe the intercellular flow of data and control information. The second level programs would consist of program fragments conceptually similar to traditional software library modules. Each agent or cell would choose which gene to express, depending on the internal state of the cell, the genome, and the states of neighboring cells. Gene expression in each cell would involve executing a program fragment, which, when combined with all other genes in the genome, would define the full system. Because the mapping of program fragments to particular cells would not be explicitly defined, the program could run on an arbitrary configuration of cells. Indeed, cells could be added to the system (hardware upgrade) or removed (hardware failure) during operation, and the system would reconfigure itself to utilize the currently operational hardware without losing functionality. This capability for self configuration, among other capabilities, was demonstrated in a software simulation of a prototype seven-cell system (see figure).

This work was done by Kenneth N. Lodding and Paul Brewster of Langley Research Center. Further information is contained in a TSP (see page 1).
LAR-16857-1

Using Covariance Analysis To Assess Pointing Performance

NASA's Jet Propulsion Laboratory, Pasadena, California

A Pointing Covariance Analysis Tool (PCAT) has been developed for evaluating the expected performance of the pointing control system for NASA's Space Interferometry Mission (SIM). The SIM pointing control system is very complex, consisting of multiple feedback and feedforward loops, and operating with multiple latencies and data rates. The SIM pointing problem is particularly challenging due to the effects of thermomechanical drifts in concert with the long camera exposures needed to image dim stars.

Other pointing error sources include sensor noises, mechanical vibrations, and errors in the feedforward signals. PCAT models the effects of finite camera exposures and all other error sources using linear system elements. This allows the pointing analysis to be performed using linear covariance analysis. PCAT propagates the error covariance using a Lyapunov equation associated with time-varying discrete and continuous-time system matrices. Unlike Monte Carlo analysis, which could involve thousands of computational runs for a single assessment, the PCAT analysis performs the same assessment in a single run. This capability facilitates the analysis of parametric studies, design trades, and "what-if" scenarios for quickly evaluating and optimizing the control system architecture and design.

This work was done by David Bayard and Bryan Kang of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).
NPO-45308