Technology & Science
IND news
INTERPLANETARY NETWORK DIRECTORATE JET PROPULSION LABORATORY
September 2002 Issue 16

In This Issue

Charles Stelzried and Michael Klein

Technology–Information Systems

Rebecca Castano, et al., in "Sample Selection and Virtual Return: On-Board Scene Prioritization and 3D Reconstruction," describe how increased computational resources of a spacecraft can improve utilization of deep space communications channels. Scene prioritization and 3D scene reconstruction using on-board computation are used for this application. The benefits of such advanced on-board processing become increasingly apparent as missions engage in longer autonomous operations.

C O N T E N T S

Technology

Science

Amir Fijany, et al., in "An Advanced Model-Based Diagnosis Engine: New Approach Enables Efficient Automated Diagnosis of Large and Complex Systems," note that general diagnosis methods have increasing exponential computational costs. A new and powerful diagnosis that mitigates this limitation is applicable to spacecraft systems.They are currently validating and benchmarking new software based on this diagnosis.

Forest Fisher, et al., in "CLEaR: Closed Loop Execution and Recovery — A Framework for Unified Planning and Execution," describe how intelligent behavior for robotic agents requires a careful balance of fast response and deliberate consideration. The primary objective of CLEaR is to provide a tightly coupled approach to coordinating goal-driven and event-driven behavior. On-board spacecraft decision-making for Mars surface exploration rovers, and the Deep Space Network (DSN) ground station communication antenna station automation are two separate research development efforts at JPL dependent on the CLEaR concept.

Technology–Communication Systems

Robert Clauss, in "New Ruby Maser Designs Optimized for the DSN," describes multiple-cavity maser designs, with calculated 4.5 kelvin (K) module noise temperatures at Ka-band, 2 K at X-band, and 1.3 K at S-band. These masers will provide the lowest possible noise temperatures while meeting the DSN's frequency, bandwidth, and gain-

ripple requirements. These rugged, robust amplifiers are the least sensitive to Radio Frequency Interference (RFI) of any low-noise amplifier (LNA) in the DSN. In addition, these new masers can be used in the commercial 4 K refrigerators presently used with the wider bandwidth and higher noise temperature HEMT LNA systems.

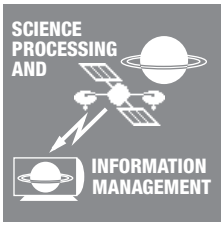
According to Houfei Fang, et al., in "Inflatable/Self-Rigidizable Reflectarray Antenna," space inflatable technology is expected to enable the development of ever larger antenna apertures with very-low launch masses and volumes for use with future space missions. This article describes an experimental three-meter diameter Ka-band (32 GHz) reflectarray antenna with an inflatable and self-rigidizable structure.

Technology–Non-Deep Space

Meemong Lee, et al., describe in "In-Situ Site Knowledge System," the In-Situ Knowledge System (ISSKS) a transition from manual to automated operations. ISSKS has three tiers composed of property modeling and synthesis, sensor system and mobility platform models and operation interface protocols. ISSKS provides for testing autonomy algorithms and on-board instrument software. Terrain Property Synthesis, Material Synthesis and Texture Synthesis are described.

Science

Last, Peter Gorham, et al., in "Update on Goldstone's Lunar Ultra-high Energy Neutrino Experiment" use the Deep Space Network antennas at Goldstone to search for neutrinos with energies greater than 10^{18} eV. When these neutrinos interact in the upper 10 meters of the Earth's moon, electromagnetic pulses are created that are expected to be detectable by large radio telescopes at Earth. Even with modest amounts of tracking time the enormous volume of material encompased by the beam of DSN antennas allows a resulting sensitivity that is enough to constrain some models for diffuse neutrino fluxes at ultra-high energy.



Sample Selection and Virtual Return:

Onboard Scene Prioritization and 3D Reconstruction

Rebecca Castaño, Robert C. Anderson, Cagatay Basdogan, Andres Castaño, Jonathan Howell, and Aaron Kiely

Introduction

The computational resources of a spacecraft can be used to reduce the load on deep space communications channels. For example, data compression reduces the transmitted data volume at the expense of increased computational overhead. As technology has advanced, the relative cost of these two resources has not decreased at the same rate: advances in semiconductors, processors, and memory design have reduced the cost of onboard computation much faster than advances in communications systems. Hence, increased use of onboard processing is becoming cost-effective for certain tasks. This is reflected in recent research stressing increasingly capable autonomous spacecraft. The challenge of performing sophisticated tasks onboard lies in the design of algorithms that can perform them autonomously and reliably.

In this article we give a brief overview of two tasks that use onboard computation to make more effective use of the constrained downlink on planetary rovers: *scene prioritization* and *three-dimensional (3D) scene reconstruction.* Scene prioritization is the process of selecting images observed by the rover and prioritizing them for downlink in an order similar to that which a scientist would assign. High priority images are selected for transmission, while lower priority images may be transmitted at lower fidelity or deleted to clear memory for new data. 3D scene reconstruction is the task of encoding the information contained in different views of an interesting scene into a unified 3D model. This process enables scene information to be downlinked as 3D structure information instead of independently transmitting the many images used to create the 3D model. This technique reduces transmitted data volume by removing the redundancy present in different views of the same scene. These tasks are potentially synergistic—the prioritization task enabling further automation of the selection of high priority scenes and the reconstruction task assisting in selecting scene features used to prioritize scenes.

Scene Prioritization

A rover on Mars acquires images of many scenes of varying scientific importance. Some will have interesting rocks or rock configurations, while others will be more barren land scenes that have little scientific value, as illustrated in Figure 1. Since onboard cameras can collect images at a rate that far exceeds the spacecraft's ability to transmit them, it is important that images selected for downlink have high scientific value. Scene prioritiza-

(a)

(b)

Figure 1. Scenes with different geological values: (a) wide variety of rock sizes and types has a higher science value than the relatively featureless scene, (b) where the few visible rocks are mostly buried in the uniform soil.

tion performs exactly this image selection task: it sorts a set of images according to their scientific value to determine a transmission order and to yield high quality data given the resources allocated.

Scene prioritization sorts a set of images according to their scientific value to determine a transmission order and to yield high quality data given the resources allocated

Scene prioritization involves two processes: the detection of scene features and the use of these features to assess the scientific value of the scene. These two processes are the algorithmic equivalents of the way geologists might evaluate an image. First, they identify features in the scene, e.g., the size, color, shape, texture and arrangement of rocks, or features of the topography, layers in a cliff face, etc. Then they use these features to assign a science value to the image. Images with interesting features—rocks with unusual colors or textures or interesting rock configurations, such as rocks along a line that might indicate a dry river bed—would be ranked higher than images without distinctive features.

Feature Extraction

The goal of the 3D scene reconstruction task is to collect and efficiently transmit data that will allow the reconstruction of a 3D scene on Earth

The first step in scene evaluation is the extraction of features of interest from the scene. Our work has focused on properties of rocks in the scene, and thus we begin by locating rocks in a stereo image pair. To locate the rocks, the stereo range information, already calculated for navigation purposes, is transformed to produce a height image, i.e., an image where the value of each pixel represents the elevation of the point above the ground plane of the scene. Level contours in the height image are then calculated and these contours are connected from peaks to the ground to identify the rocks [1]. Rock properties including albedo and visual texture are then extracted from the rocks identified. We measure albedo, which indicates the reflectance properties of a surface, by computing the average gray-scale value of the pixels that comprise the image of the rock. Visual texture can be described by intensity variations at different orientations and spatial frequencies within the image. We measure texture using a filter bank in which each filter is tuned to a different combination of orientation and spatial frequency [2].

Image Ranking

The features extracted from a group of images are then used to rank the images by partitioning the rocks into classes, each class consisting of rocks with similar properties. To determine the classes, the property values are concatenated to form a feature vector and a weight is assigned to the significance of each property. Different weight assignments can be used as a function of the particular properties that are of interest. For example, albedo and texture are typically used to distinguish types of rocks, but rock size and location may be used if sorting is of interest. For each class of rocks, we find the most representative rock in the class, i.e., the single rock in any image that is closest to the mean of the set. We give a high priority to the image containing this rock. Since some classes of rock may be more common than others, this prioritization method ensures that all classes of rocks in a set of images are represented. The optimal number of classes can be determined using cross-validation techniques [3]. Without prioritizing the rocks by class, a disproportionate number of rocks from the most common classes are more likely to be represented in the downlinked data, and classes that have fewer members may not be represented at all.

To validate our prioritization methods, we determined the correlation between automated ranking and a ranking of the same data set defined by scientists [4]. Humans are very adept at selecting between two options, but may have difficulty ranking a larger set of items. Consequently, we developed a web-based pair-wise image *prioritizer,* whose graphical interface is shown in Figure 2. A sequence of image pairs is presented to a user who then identifies the higher priority image. The results from this pair-wise ranking are compiled into a complete ranking of the image set [5]. The web-based prioritizer permits us to collect rankings from scientists who may be located across the country.

Clearly scientists with different expertise and interests will prefer different features, e.g., a hydrologist might be interested in the presence of round rocks, while a volcanologist might be interested in porous highly-textured rocks, and a microbiologist might be interested in white rocks that could indicate the pres-

ence of carbonates. Hence, it is not possible to define an objective "most interesting feature." To partially address this issue, we not only compare expert rankings to the ranking results obtained by our algorithms, but we also compare the rankings of experts to each other to determine the consistency across the set of scientists.

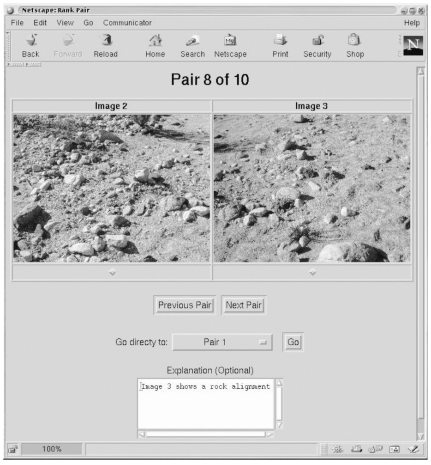


Figure 2. Web-based image-set prioritizer. A sequence of image pairs is displayed to the expert who is prioritizing them. For each pair, the expert clicks a radio button to select the higher-ranking image. A box is available for comments on why a particular decision was made. When a selection has been made for all pairs, the complete ranking is computed and displayed. This ranking can be compared with previous expert rankings and can be included in a composite ranking based on multiple experts.

Scene Reconstruction

The goal of the 3D scene reconstruction task is to collect and efficiently transmit data that will allow the reconstruction of a 3D scene on Earth. This might allow one to transmit a "virtual rock" from Mars or to use scene information for rover navigation or other purposes. Such information could be transmitted simply using a sequence of stereo image pairs from different viewpoints within a scene or different views of an object of interest. However, the number of images collected might be quite large, and it is difficult to optimize the collection and selection of such images to maximize the value of a set of images of a scene. Thus, we investigate the process of constructing an onboard 3D representation of the scene, efficiently transmitting this information and reconstructing 3D objects or scenes on the ground. This approach can more effectively use downlink resources when the equivalent number of images transmitted is large. The 3D structure information can also be arranged hierarchically, which permits progressive transmission of low- to high-fidelity representations of the scene. The 3D scene reconstruction process consists of the following steps: image acquisition and range estimation, range registration, data compression/decompression, integration, and visualization.

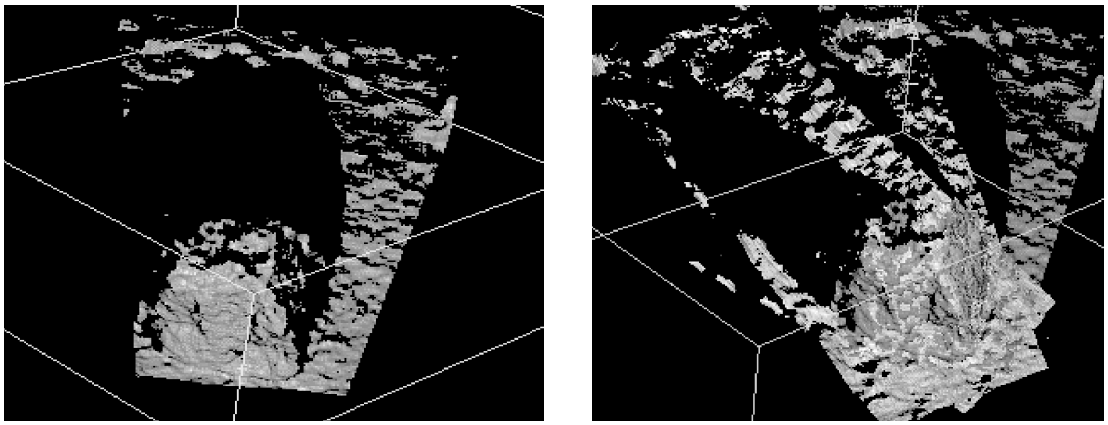
Image Acquisition and Range Estimation

Image acquisition consists of obtaining stereo image pairs of the scene from different points of view. To accomplish this task we have built a small mobile platform with a stereo head to capture stereo image sequences, while adjusting parameters such as the pitch and yaw of the cameras, their focal length, gain and aperture, and the frame rate of the sequence. This platform provides a way to obtain image sequences at the JPL Mars yard to support the prioritization and the reconstructions tasks of the project without requiring the availability of a full planetary rover prototype.

Once image pairs are acquired, we obtain range estimates of the scene using the JPL stereo ranging algorithm [6], which will be used on the Mars Exploration Rover (MER), operating on a computer on the cart. The range estimates produce a "cloud" of points in 3D space, or point cloud, taken from the estimated locations of the objects seen in the stereo pair. As an example, Figure 3a shows a point cloud of a rock and part of its background. The accuracy with which the point cloud data matches the actual object locations depends on several factors, most notably the quality of the calibration of the cameras, the focal length of the lens, and the resolution of the image.

Range Registration

The range registration process reconciles the various point clouds of the scene obtained from different points of view to form a cohesive representation of the scene. The process of registering two point clouds requires finding the transformation that best positions one point cloud over the other to obtain a single point cloud that combines the two originals. This pair-wise registration procedure is repeated until all the point clouds of the scene are merged into a single point cloud. Figure 3b shows an example of the registration of three scene views using a package based on the Iterative Closest Point (ICP) algorithm [7].



(a) (b)

Figure 3. Scene with a rock: a) point cloud from one view, b) registration of point clouds from three different views.

Data Compression/Decompression

A point cloud of sufficient density provides enough information to reconstruct a solid model of a 3D scene or object. Thus the next step is to efficiently transmit the point cloud; it is not necessary to construct a solid model (e.g., a wireframe mesh) of the scene or object onboard. As a first step before compression, the point cloud is reduced to a manageable size by removing from dense areas some of the closely spaced points that add little information. The color values of all the closely spaced points of a given area are used to assign a color value to a remaining point that will represent the set, i.e., we have a colored point cloud where each point of the cloud has an associated color value.

(a) (b) (c)

Figure 4. The Stanford bunny: a) original high-resolution model (35,947 points using 1,294,092 bits without compression), b) reconstructed from 783 points (1,837 bits), c) reconstructed from 4,663 points (23,057 bits).

The point cloud data are compressed and transmitted using a progressive compression scheme. Progressive transmission arranges compressed data hierarchically so that as more compressed data are received, successively higher fidelity data reconstructions are possible. Figure 4 illustrates progressive transmission using the Stanford bunny. At each step in compression, we have a set of points that provide an approximate representation of the information contained in the complete point cloud. We use the next set of compressed bits to increase the number of points in the cloud and/or refine the accuracy of the positions of existing points. The compressor is adaptive, learning about the likelihood that a point will be split into two or more points at each stage based on previously transmitted information in the data set. This probability information is used by an entropy coder to compress the update information. The compressed point cloud data is decompressed on the ground to recover an approximate representation of the original point cloud.

Data Integration and Visualization

The "colored" point cloud recovered from the decompressor provides the input to the integration step, where a solid model (i.e., a wireframe mesh) is computed from the point cloud information. This computation requires estimating the normal vectors to the surface of the scene for each point in the cloud. This estimation is done using Hoppe's algorithm [8], which computes a surface normal vector for each point based on the location of nearby points. We use a minimal spanning tree to resolve conflicts between normal vectors pointing "inward" or "outward" to the surface. Once the surface normal

vectors have been estimated, the marching cubes algorithm [9] is used to construct a wireframe mesh. Finally, the individual patches are colored using the color values of nearby points. The result, similar to that shown in Figure 5, can be displayed using a 3D display and/or manipulated using a haptic interface, which is a device that allows the user to feel the shape and texture of the object using a mechanical force feedback system.

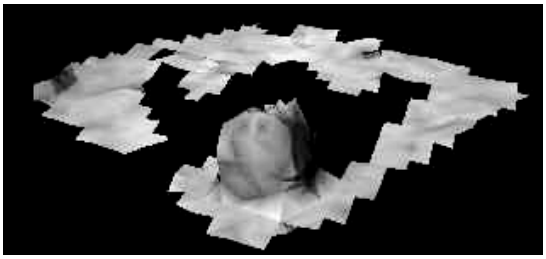


Figure 5. 3D scene reconstruction of scene with rock registered in Figure 3b.

Conclusion

Scene prioritization and 3D scene reconstruction are two techniques that can exploit increased spacecraft computational capabilities to significantly reduce the burden on the communications system. Prioritization can be a valuable tool to maximize the science return on a mission when large areas are explored, but only a limited amount of data can be returned to Earth. 3D scene reconstruction can lead to high compression of raw data while providing scientists with detailed information about particular targets or scenes of high interest. The benefits of such advanced onboard processing become increasingly apparent as missions engage in increasingly longer and more ambitious autonomous operations.

Acknowledgement

The 3D scene reconstruction work builds on earlier work by Roberto Manduchi, now at University of California, Santa Cruz, and Isaac Cohen and Alexandre Benotmane of the University of Southern California. The authors also gratefully acknowledge Mark Levoy of Stanford University for providing the source code of the Scanalyze package used for registration and the JPL Machine Vision group for providing the camera calibration and stereo ranging applications. The Stanford bunny data, created by Greg Turk and Mark Levoy, was obtained from the repository at the Stanford Computer Graphics Laboratory.

References

[1] V. Gor, R. Castano, R. Manduchi, R. C. Anderson and E. Mjolsness, "Autonomous rock detection for Mars terrain," *American Institute of Aeronautics and Astronautics*, Aug. 2001.

[2] R. L. Castano, T. Mann, and E. Mjolsness, "Texture analysis for Mars rover images," *Proc. Applications of Digital Image Processing XXII*, SPIE Vol. 3808, July, 1999.

[3] P. Smyth, "Clustering using Monte-Carlo cross-validation," *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, 1996.

[4] E. L. Lehmann and H.J.M. D'Abrera, *Nonparametrics: Statistical Methods Based on Ranks*, Prentice Hall, 1998.

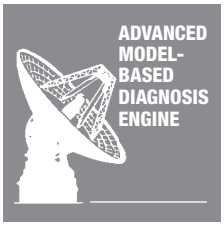
[5] D. A. Seaver and W. F. Stilwell, "Procedure for using expert judgments to estimate human error probabilities in nuclear power plant operations," Nuclear Regulator Commission, GPO Sales Programs, Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, D.C., 1983.

[6] L. H. Matthies, "Stereo vision for planetary rovers: stochastic modeling of near real-time implementation," *Intl. Journal Computer Vision*, 8(1):71-91, 1992.

[7] P. J. Besl and N.D. McKay, "A method for registration of 3D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-256, 1992.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH*, pp. 71-78, 1992.

[9] W. E. Lorenson and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH*, Vol. 21, No. 4, pp.163-169, 1987.



An Advanced Model-Based Diagnosis Engine:

New approach enables efficient automated diagnosis of large and complex systems

Amir Fijany, Anthony Barrett, Farrokh Vatan, Mark James, and Ryan Mackey

These new approaches are a significant improvement over the existing state-of-the-art

Introduction

The number of Earth orbiters and deep space probes has grown dramatically over the last decade and this trend is expected to continue. This rate of growth has brought a new focus on autonomous and self-preserving systems, all of which depend on fault diagnosis. Although diagnosis is needed for any autonomous system, current approaches tend to be "ad hoc," inefficient, and incomplete. Systematic methods of general diagnosis exist in literature, but all suffer from two major drawbacks that limit their practical applications. First, they tend to be large and complex and hence difficult to apply. Second, and more importantly, they rely on algorithms with exponential computational cost and thus become impractical as the number of system components increases.

We have developed a new and powerful diagnosis engine [1] that overcomes these limitations through a two-fold approach. First, we propose a novel and compact reconstruction of the General Diagnosis Engine (GDE), one of the most fundamental approaches to model-based diagnosis. We then present a novel algorithmic approach for minimal diagnosis set calculation. This approach uses a powerful yet simple representation to map minimal diagnosis set calculation onto two well-known problems: *Boolean satisfiability* [2] and *integer programming* [3].

Boolean satisfiability problems take the form of a logical expression. They comprise many parts and include a search to find whether a combination that makes the expression *true* exists. These problems are common in digital electronics and have been studied thoroughly. The mapping onto Boolean satisfiability provides access to an existing library of highly-efficient algorithms, permitting solution with superpolynomial rather than an exponential cost.

Integer programming problems, on the other hand, are problems of optimization where the variables may take only integer values. For example, how many cars and trucks should a factory produce to earn the most profit given that cars and trucks cost different amounts to build and that one cannot build a fraction of a car. There are integer programming algorithms that can be efficiently applied to systems with as many as several thousand components. Together, these new approaches are a significant improvement over the existing state-of-the-art that can handle, at most, approximately 50 components. In addition, the integer programming approach allows determination of the solution size before actually solving the problem, a previously unavailable insight that can be exploited to simplify diagnosis problems themselves.

This article describes the strengths and weaknesses of both new approaches. We will begin by describing the diagnosis problem in general, and then consider how the problem may be restated in different forms.

Overview of Model-based Diagnosis

System diagnosis is the task of identifying faulty components within a system – components that cause the system not to function as intended. The diagnosis problem arises when *symptoms* are observed, i.e., when the system's actual behavior contradicts expected behavior. To diagnose the problem, we must determine which set of components must be faulty in order to fully explain all observed symptoms. Of course, what we really want is the *minimal set* of faulty components. We can always explain all symptoms by asserting the trivial so-

lution — that all components may be faulty — but this does not provide any useful information.

Model-based diagnosis, first suggested by Reiter [4] and later expanded by de Kleer, Mackworth and Reiter [5], is the most disciplined general technique for system diagnosis. This technique reasons from first principles, employing a model of how devices work and how they are connected. The model provides a representation of correct system behavior and describes the system's structure. Given such a model, it is then possible to ask for reasons why expected behavior was not observed.

Model-based diagnosis techniques have been widely recognized due to the following advantages [6]:

- The same model can be used to search for single or multiple faults in the system
- Different diagnosis algorithms can be used for a given model
- A clear formal basis for judging and determining diagnoses exists

As an example, Figure 1 illustrates a model of a hydrazine propulsion subsystem, in this case for Cassini's attitude thrusters [7]. This model consists of a set of connected components, where the connectors represent pipes in the actual system. Pressure sensors P_1 through P_6 are monitored for discrepancies while valves open and close during normal operations.

Model-based diagnosis focuses on logical relations between components of a complex system. The function of each component and the interconnection between components are all represented as a logical system called the *system description*. Any expected behavior of the system is logically consistent with the system description. If components of the system are faulty, observations of the system will be inconsistent with the system description. Therefore, determining which components are faulty (diagnosis) is the same as finding which parts of the model must be considered abnormal in order to explain all of the inconsistencies.

In summary, the diagnosis process starts with identifying symptoms that represent inconsistencies between the system's model and actual behavior. Taken individually, each symptom identifies a *candidate set* of *conflicting components* that might be at fault. These candidate sets contain all possible scenarios that might cause the symptom – often a large set. The *minimal diagnosis* is the simplest scenario (the smallest combination of faulty components) that intersects all of these candidate sets.

At this juncture, it should be apparent that a brute-force solution of this problem is quite difficult. The number of possible scenarios that can explain a single symptom tends to be quite large, and grows rapidly as the component count increases. Brute-force approaches

Model-based diagnosis techniques have been widely recognized

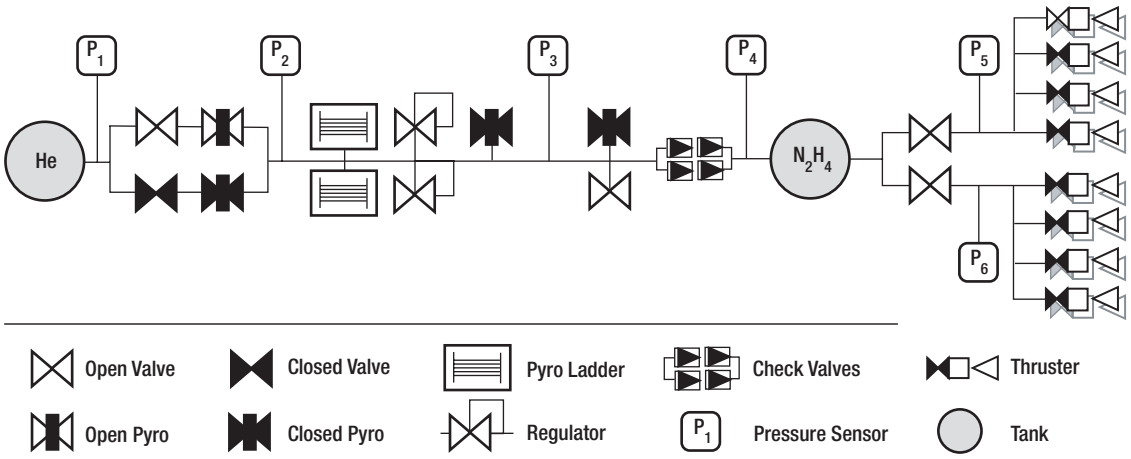


Figure 1. Diagram of Cassini attitude control thrusters

are, in fact, exponentially complex. This puts a hard computational limit on its range of applicability, a limit that falls well short of many typical spacecraft systems.

Because of this limitation, spacecraft engineers often employ f*ault protection modes*, which define a set of symptoms-to-cause rules. In this approach, human experts attempt to predict and determine the causes of all possible faults ahead of time. This approach overcomes the exponential cost by relying upon the expertise of the system designers. As a result, this approach is costly, prone to human errors, and incomplete (since it is impossible to predict all faults in advance). We have learned from Cassini designers that development of its fault protection modes took more than 20 work years.

A novel reconstruction of general diagnosis engine

To describe usage of the General Diagnosis Engine (GDE) [8], we return to Cassini's attitude thrusters example, as illustrated in Figure 1. Computationally, each connector corresponds to a set of variables, and each component corresponds to a set of rules that update these variables. In this example, each pipe is associated with pressure and flow. The valves are represented by rules that select these variables depending on whether or not a valve was commanded open or closed, assuming that the valve is functioning correctly. GDE performs a causal simulation by observing the variables and computes the values of other variables using the rules. These computations have underlying assumptions about whether or not components are working properly. GDE tags each value with the assumptions that affect its computation.

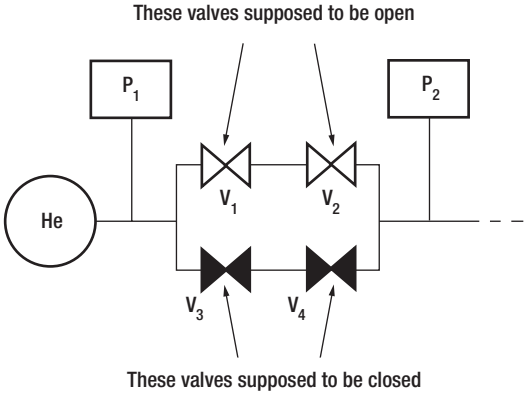


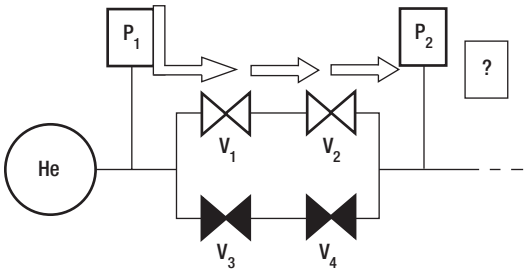
Figure 2. Portion of the Cassini thruster system

A discrepancy arises when two different values are assigned to the same variable. For instance, consider the two open valves between pressure sensors P_1 and P_2, as illustrated in Figure 2. Assuming that both valves work, we should have the same pressures at each sensor. However, if the values we measure are different from what we expect from the model, we have a discrepancy.

If the sensors report substantially different pressures, then at least one sensor or one valve must be faulty. In general, whenever GDE computes two incompatible values for the same variable, this means there is a contradiction in the supporting assumptions. To illustrate this process, let us suppose we measure different values at the two pressure sensors: P1 measures 500 PSI and P2 measures 0 PSI. Obviously one of the sensors or one of the two valves is not working properly, but which one? GDE attempts to answer this problem by setting up different assumptions and testing them against the model. This gives us a way to distinguish between assumptions that are correct, and those that cannot be true.

In our example (Figure 3), let us consider two different sets of assumptions and how they affect the model. In this case, our assumptions that P_1 and the two valves work give us one pressure (Figure 3a) and assuming P_2 and the two valves work gives us a different pressure (Figure 3b). Clearly something is wrong with these assumptions and we need to combine them to find out which part is incorrect.

When failures occur, multiple incompatible assumption sets appear. For instance, if we continue our simulation to the right, after we compute P_2 we will compute two values at P_3 with different assumption tags, and so on through the entire model. If we measure the wrong value at P_3, we combine P_3 with one (or both) of these tags to form additional incompatible assumption sets. This process continues until the simulation completes.



Assume: Sensor P_1 and valves V_1 and V_2 are working

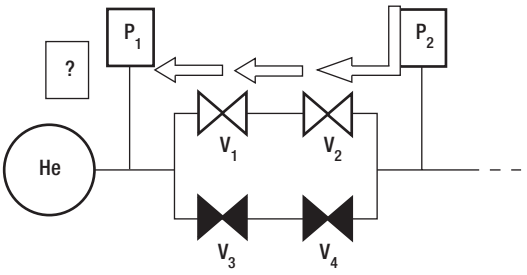
Then: Pressure at V_1 = Pressure at P_1 = 500 PSI

Pressure at V_2 = Pressure at V_1 = 500 PSI

Pressure at P_2 = Pressure at V_1 = 500 PSI

BUT: Contradicts measurement at P_2

(a)



Assume: Sensor P_2 and valves V_1 and V_2 are working

Then: Pressure at V_2 = Pressure at P_2 = 0 PSI

Pressure at V_1 = Pressure at V_2 = 0 PSI

Pressure at P_1 = Pressure at V_1 = 0 PSI

BUT: Contradicts measurement at P_1

(b)

Figure 3. Navigation through the model given (a) one set of assumptions and (b) a different set of assumptions

After causal simulation, we must find the minimal set of assumptions that intersects with all of these incompatible sets. This set contains the actual diagnoses of the root causes for all of the contradictory measurements. However, GDE suffers from the two main limitations of other model-based diagnosis approach; that is, the complexity of software makes its application difficult, and there is an exponential computational cost for finding the minimal set.

To overcome difficulties in application, we have developed a novel and compact reconstruction of GDE. Traditionally, GDE has been implemented using an inference engine to reason about the model combined with an Assumption-Based Truth Maintenance system (ATMS) to keep track of the assumptions. Our rational reconstruction of GDE merges the ATMS with the inference engine. The two pieces have many similarities and combining them dramatically simplifies the algorithm: our implementation contains fewer than 150 lines of LISt Processing (LISP) code. Directly linking these functions also facilitates the use of computation reduction heuristics, such as tagging each assumption with a probability to focus the system on relatively likely faults.

The computational limitation is far more challenging. It is easy to describe the problem of minimal set determination, but the computation itself is intractable. We have recently proven [9] that it belongs to a special class of problems (*NP-Complete* problems) that cannot be solved in less than exponential time. However, we have found a way to map the diagnosis problem onto Boolean satisfiability and integer programming problems, allowing us to take advantage of existing algorithms with vastly improved performance. While these algorithms are not complete, having special cases for which they will not work, these special cases are extremely unlikely to arise in real spacecraft systems. We may also take advantage of different algorithms, selecting the best performing method for each particular problem.

Calculation of Minimal Diagnosis Set as Solution of Hitting Set Problem

The *hitting set problem*, also known as the *transversal problem*, is one of the key problems in combinatorics of finite sets and the theory of diagnosis. The problem is described as follows: Suppose we have a set of objects, and a collection of nonempty subsets. A hitting set (or transversal) is a subset that meets every set in the collection C: it must have at least

one object in common with every subset in the collection. There is always at least one hitting set. For example, the entire set of objects intersects all of the subsets and is therefore a hitting set.

For the diagnosis problem, we are only interested in *minimal hitting sets*, hitting sets that contain no "extra" objects. If we exclude any element in a minimal hitting set, it no longer intersects all of the subsets in the collection.

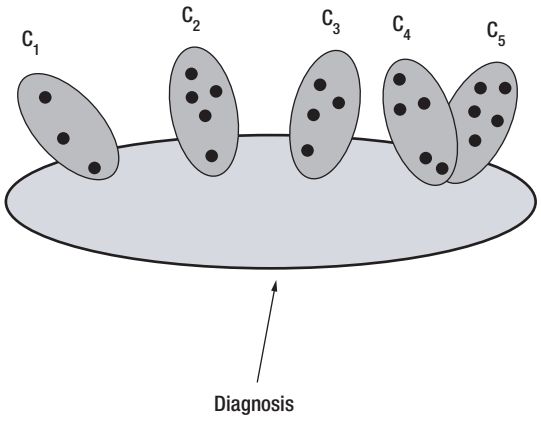


Figure 4. Diagnosis as a hitting set of conflicts

Solving the hitting set problem is similar to calculation of minimal diagnosis set. Each symptom identifies a set of conflicting components as initial candidates and minimal diagnoses are the smallest set of components that intersect all candidate sets. In Figure 4 above, each blue oval represents all of the possible scenarios that could explain a single symptom. Each black dot represents a scenario, i.e., a specific combination of components that could be faulty. Any minimal hitting set is a possible way to explain all of the symptoms. The smallest minimal hitting set, or the diagnosis with the smallest number of failed components, is the best diagnosis.

Reiter's hitting set algorithm [4] is a well-known algorithm to find minimal hitting sets. But if we want to find the smallest minimal hitting set, this algorithm is not efficient, requiring exponential time and an exponential amount of computer memory. We will describe an improved algorithm that requires only a linear amount of memory, though it still requires exponential time.

Hitting Set Problem Mapping onto Boolean Satisfiability

Another way to represent the sets of conflicts, like the ovals in Figure 4 above, is in the form of a table. We can create a matrix, where all of the possible conflicts are the rows, and the columns are possible combinations of failed components.

Each row corresponds to a measurement being something unexpected. That unexpected measurement could be caused by a number of different problems. If a particular problem scenario could cause that measurement to change, we put a "1" in that entry. If that problem scenario would not affect the measurement, we write a "0". The result looks something like Table 1 below:

Table 1. Representation of conflicts-components relationship

	M_1	M_2	• • •	M_n
C_1	1	0	• • •	0
C_2	0	1	• • •	1
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
C_m	1	1	• • •	0

To go backwards from this table, each conflict (row) can be thought of as the union of all the elements for which there is a "1" in that row. Likewise, a hitting set is any set that intersects all of the conflicts. If we represent the hitting set by a vector, that vector must have at least one "1" in common with every conflict. We can use this representation to set up a Boolean satisfiability problem, in which we create a logical formula and search for solutions that satisfy the formula. Here, we are after the hitting sets, so we construct a formula that is True if a given set intersects all of the conflicts. Our formula is quite simple.

With this representation, we can bring to bear our previous breakthrough [9] regarding determination of hitting sets in general. Reference [9] describes an algorithm that produces partial solutions to the problem — so-called "prime implicants" or components/

combinations that are definitely part of the failure—far faster than brute force solutions. As the number of components gets larger, the amount of computer time required for this algorithm increases like $n\ t + n^{\log n}$, where n is the number of components in the system and t is the time needed to evaluate the Boolean function. This is slow, but it is much faster than the 2^n calculations needed for brute-force methods. Our algorithm also has the advantage that it does not produce repetitions.

This algorithm is best suited to problems that have relatively few different hitting sets, giving faster results in this case. This is usually true for practical space systems, since most are designed hierarchically and with sensors placed throughout the subsystems.

However, we may run into cases that simply do not work efficiently with this algorithm—such is the danger of any approximate method, but we can apply a similar line of reasoning to tackle the problem in yet another fashion. This other method, based on integer programming, allows us to use a different group of algorithms, some of which may work where the Boolean satisfiability approach comes up short. These different approaches can be evaluated during design of the diagnostic system, and the best algorithm chosen for that particular problem.

Hitting Set Problem Mapping onto Integer Programming

We will return to the representation in Table 1 to describe the mapping onto the 0/1 integer programming problem. Note that each row of the matrix corresponds to a *subset* and each column to a *member*.

Another way to phrase the minimal hitting set problem is the following: We wish to find the smallest collection of columns of the matrix, such that when the column vectors are added together, the result has a "1" (or greater number) in every single position. We can describe this as an 0/1 integer programming problem:

minimize $wt(\mathbf{x})$ subject to $\mathbf{Ax} \geq \mathbf{b}$

where A is the matrix, **b** is a vector containing all 1's, and wt(**x**) is the *Hamming weight*, which is simply the number of 1's in the vector **x**. When this is solved, the vector **x** represents the minimal hitting set, the solution to the diagnosis problem.

The equation above represents a special case of the 0/1 integer programming problem since the matrix only contains 1's or 0's. Interestingly, our above derivation can also be applied to the monotone Boolean satisfiability problem.

While these subtle rewordings of the diagnosis problem appear simple, they provide immediate access to a wealth of algorithms devised for Boolean satisfiability and integer programming problems. In practical terms, this permits a tremendous improvement in diagnosis speed and the size of problems that can be treated.

Conclusion

We have developed a two-fold approach that overcomes major limitations of current model-based diagnosis techniques. Our method begins with a novel and greatly improved reconstruction of GDE and then incorporates a novel algorithmic approach for minimal diagnosis set calculation. Mapping onto Boolean Satisfiability enables the use of efficient algorithms with superpolynomial rather than exponential complexity. Mapping onto 0/1 integer programming enables the use of algorithms that can efficiently solve the problem for as many as several thousand components. These two separate approaches allow us to choose the algorithm that best suits a specific problem.

In addition, the integer programming formulation allows, for the first time, *a priori* determination of the minimum number of faulty components [1]. We are currently devising new techniques to exploit the special matrix structure of this problem to develop yet more efficient algorithms.

We are also presently validating and benchmarking a new software tool based on this technology. Even though the preliminary version of this new diagnosis engine is

We are currently devising new techniques to exploit the special matrix structure of this problem to develop yet more efficient algorithms

intended to demonstrate validity of the mapping and not the algorithms, and therefore uses a brute-force approach, it has been able to efficiently diagnose example systems with up to 23 components and up to over 300 conflict sets. Our current effort focuses on integration of efficient algorithms for solution of the Integer Programming Problem. Using commercially available algorithms, much better results are expected soon. Based on prior benchmarks of these algorithms, we anticipate a minimum performance improvement of two orders of magnitude, with considerable potential for further improvement using customized algorithms and further problem refinement. Once this tool is completed, we will test its application to a variety of problems in the Deep Space Network, paving the way for future infusion along with other autonomy technologies.

References

[1] A. Fijany, F. Vatan, A. Barrett, and R. Mackey, "New approaches for solving the diagnosis problem," *The JPL Interplanetary Network Progress (IPN) Report*, 42-149, 15 May, 2002

[2] V. Gurvich and L. Khachiyan, "On generating the irreducible conjunctive and disjunctive normal forms of monotone Boolean Functions," *Discrete Applied Mathematics*, 96-97, pp. 363-373, 1999.

[3] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[4] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intelligence* 32, pp. 57-95, 1987.

[5] J. de Kleer, A. K. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," *Artificial Intelligence*, 56, pp. 197-222, 1992.

[6] C. Mateis, M. Stumptner, D. Wieland, and F. Wotawa, "Model-based debugging of Java programs," *Proc. of the Fourth Int. Workshop on Automated Debugging*, pp. 260-265, 2000.

[7] "Cassini Program Environmetal Impact Statement Supporting Study, Volume 3: Cassini Earth Swingby Plan," JPL D-10178-3, November 18, 1993.

[8] J. de Kleer and B. Williams, "Diagnosing Multiple Faults," *Readings in Model- Based Diagnosis*, Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[9] F. Vatan, "The complexity of diagnosis and monotone satisfiability," submitted to *Discrete Applied Mathematics*, 2001. Also to appear in *NASA J. Tech Brief*, 2002.

CLEaR: Closed Loop Execution and Recovery —

A Framework for Unified Planning and Execution

Forest Fisher, Tara Estlin, Daniel Gaines, Steve Schaffer, Caroline Chouinard, Russell Knight

Introduction

Intelligent behavior for robotic agents requires a careful balance of fast reactions and deliberate consideration of long-term ramifications. The need for this balance is particularly acute in space applications, where hostile environments demand fast reactions and remote locations dictate careful management of consumables that cannot be replenished. However, fast reactions typically require procedural representations with limited scope, and handling long-term considerations in a general fashion is often computationally expensive.

Robotic agents performing under hard resource and time constraints in uncertain environments require careful balancing of both deliberative and reactive reasoning [1]. As in most domains with uncertainty, a task may fail or produce unexpected results leading to plan failures. If the robot is also under hard time deadlines and resource constraints, a task requiring a different time or resource allocation than planned could fail at future points in the plan. In some cases, the robot may be able to retry a failed task, use more time, or take up more of a resource without causing a problem.

In this article, we focus on the high-level autonomy issues of balancing reaction and deliberation, and declarative and procedural representations as they relate to planning and execution.

Issues in Planning and Execution

We define autonomy as a system's ability to handle complex tasks involving environmental feedback without external intervention or supervision. Examples include operating free-flying robotic spacecraft for mapping and science observation missions, operating surface rovers on extra-terrestrial missions, and tracking spacecraft using ground communications stations. While there are many aspects of autonomy, we focus on issues arising from automated planning and scheduling in the context of real-time task execution. These consist of plan and task representation, system control, plan generation, plan execution, monitoring, and error handling.

Knowledge Representation

A complete autonomous system consists of hardware, software (engines) and models. This article focuses on software and models. Software and models may be defined either declaratively or procedurally. Software (e.g., a planning engine or task execution engine) provides the control information necessary to use the information provided by models. A model is the knowledge about a domain or situation.

Procedural models represent domain knowledge by embedding it in a control stream. This is sometimes referred to as "arbitrary code." It is often quite difficult to separate the control information from the domain information in procedural models. Declarative models are static representations about events, objects, and their relationships. The software required to use a declarative model may be quite complex; this is the topic of considerable artificial intelligence research.

The tradeoff between procedural models and declarative models is that while procedural models can be quickly encoded for specific domains, procedural models are considered brittle and difficult to change. Declarative models do not commit to any particular control path, and thus in theory can be more flexible with respect to uncertain and unknown events. However, the software

that uses these models can be slow and often requires more computational resources than software that uses models.

Plan Generation and Execution

Plan (and/or schedule) generation is the act of devising a set of actions (the plan) to realize a task or set of goals. In order to realize a task or achieve a goal, a system can use deliberation and/or reaction. Deliberation is the process of producing sub-tasks that when executed, result in the realization of tasks/goals. A collection of executable sub-tasks is also referred to as a plan. Deliberation is the search required to find an appropriate plan; it is not guaranteed to find a solution.

The tradeoff between deliberation and reaction is that deliberation may require more resources (time and computation) than reaction, resulting in missed deadlines. Reaction may waste resources. We see the use of both techniques to achieve robust autonomy

Reaction is the act of producing the next executable sub-task required to achieve a task. Therefore, reactive systems generate a plan one step at a time, only generating the next step in the plan and waiting for the results before continuing. Reaction performs little or no search, and thus is able to provide a bounded time response. It is a very powerful technique that deals well with real-time issues. However, reaction has the weakness that it may take harmful actions that result in suboptimal performance or even unnecessary failure because it does not perform look-ahead. Even though it makes its decisions locally without look-ahead, reaction still has many powerful capabilities such as the ability to respond to execution errors in a timely fashion through generic error handlers and task trees as well as the ability to synchronize multiple execution threads.

The tradeoff between deliberation and reaction is that deliberation may require more resources (time and computation) than reaction, resulting in missed deadlines. Reaction may waste resources trying to achieve a task and might not solve the problems associated with achieving a task at all. Further deliberation allows for plan optimization and the ability to solve the problem within the global context of the robotic system and/or mission desires. We see the use of both techniques to achieve robust autonomy.

Plan execution is the act of realizing a task given a plan. Technologies used for plan execution include (among others) execution software and mode identification. Execution software takes the representation of a plan and controls the hardware to achieve the tasks in the plan. Mode identification is the act of using sensor information and plan context to determine the state of the environment and the state of the autonomous system. The combination of execution software and mode identification allows plan execution to achieve tasks of a plan and to know that the tasks have been achieved. Likewise, failure to achieve a state can be known and reported back to the plan generation system. Reactive plan generation immediately provides a new task (probably very similar to the failed task) to be executed. If plan generation is deliberative, a new plan or revised plan is produced.

Declarative and procedural representations of models as well as deliberative and reactive plan generation are required for intelligent robotic systems.

A Unified Planning and Execution System for High-level Robotic Automation

We have developed a high-level decision-making capability for future autonomous robotic systems called Closed-Loop Execution and Recovery (CLEaR). CLEaR currently consists of two major components with support for interacting with a third general class of components:

- **A Continuous Planner** that provides capabilities for initial command sequence (plan) generation and continuous updating of that plan (i.e., replanning) based on changing operating context and goal information.
- **A Reactive Task-level Executive** that provides task-level control capabilities for a robotic system, including execution and monitoring of the plan, as well as mediation between a planner and low-level robotic functionality.
- **Domain Specific Solvers:** For example in the rover domain, a *global path planner* that provides global path planning information about predicted routes to both the planner and executive is plugged into the framework.

We will begin by first introducing the CLEaR framework for unified planning and execution, then give a more detailed description of the individual components that comprise the framework, and then discuss how the framework addresses many of the issues presented. In the Applied Domains section we will briefly describe how this framework has been used to date.

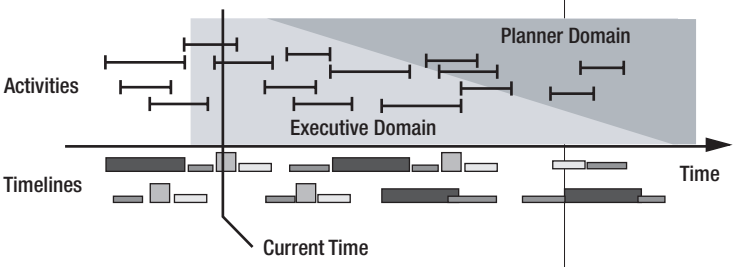
CLEaR Framework

The CLEaR unified planning and execution framework [2] was developed to tightly integrate planning and execution capabilities. Currently, CLEaR is a hybrid controller system that is built on top of the Continuous Activity Scheduling, Planning, Execution and Replanning (CASPER) continuous planner and the Task Description Language (TDL) executive system. CASPER provides a soft-real-time capability for performing plan generation, execution, monitoring and replanning. Versions of the CLEaR framework have been demonstrated for Deep Space Network (DSN) antenna control [3] and planning and execution support for planetary rovers [4].

CLEaR's primary objective is to provide a tightly coupled approach to coordinating goal-driven and event-driven behavior. Many past approaches have followed a three-level architecture style where the planning and executive processes are treated as *black box* systems. This is in contrast to how CLEaR enables the planner and executive to interact with each other and more effectively share the responsibility for decision making. In part, this is managed through shared plan information and continual updates of state being made available to both the planner and executive. CLEaR also provides heuristic support for deciding when certain plan conflicts should be handled by the planner vs. the executive. For instance, if a rover gets off track during a traverse, the reaction of the planner and executive needs to be coordinated. If the executive believes it can resolve the navigation delay within the original allotted time, it will manage the plan changes. However, if the executive identifies that the repair will require more time or resources than allotted by the planner, it will allow the planner to use its global perspective to fix the problem.

Figure 1 depicts the concept of shared responsibility in the plan modification process. In the bottom half of the figure, the timelines depict states and resources affected by the plan activities. In the top half of the figure, the I-bars represent planning activities with the left edge of the bar representing start-time and bar's length depicting the duration of the activity. The shaded wedges, creating the background of the activities, depict the primary reasonability regarding plan manipulation over time. In this figure, time advances from left to right and is marked by the *Current Time* marker. As you move into the future (to the right) the changing thickness of the wedges are depicting the decreasing responsibility of the executive and the increasing responsibility of the planner.

Figure 1. CLEaR concept diagram



CASPER Planner

Planning in CLEaR is provided by the CASPER system [5]. Based on an input set of science goals and a rover's current state, CASPER generates a sequence of activities that satisfies the goals while obeying relevant resource constraints and operations rules. Plans are produced by using an *iterative repair* algorithm that classifies conflicts and resolves them individually by performing one or more plan modifications. CASPER also monitors current rover state and the execution status of plan activities. As this information is acquired, CASPER updates future-plan projections. Based on this new information, new conflicts and/or opportunities may arise, requiring the planner to replan in order to accommodate the unexpected events.

TDL Executive

Most executive functionality in CLEaR is performed by the TDL executive system [6]. TDL was designed to perform task-level control for robotic control and to mediate between a planning system and low-level robot control software. It expands abstract tasks into low-level commands, executes the commands, and monitors their execution. It also provides direct support for exception handling and the fine-grained synchronization of subtasks. TDL is implemented as an extension of C++ that simplifies the development of robot control programs by including explicit syntactic support for task-level control capabilities. It uses a construct called a *task tree* to describe the tree structure that is produced when tasks are broken down into low-level commands.

An Instantiation of the CLEaR Framework

Figure 2 depicts an instantiation of the CLEaR framework for a rover domain. In this instantiation, the CASPER planner selects planning activities to be executed by the TDL executive, which in turn, commands the low-level robotic control software. For the rover domain, the low-level robotic control software is provided by the Coupled-Layer Architecture for Robotic Autonomy (CLARAty) Functional Layer [7]. As CLARAty Functional Layer carries out the commands, command status (such as "in progress" or "completed"), and vehicle state and resource updates are provided. This enables TDL to perform execution time monitoring of the tasks being performed. Similarly, any of these updates that pertain to information being tracked in the planner, including planning activity status, is provided to the planner. CASPER then uses these updates to project the state of the agent forward in time. If these updates create inconstancies with the projected state, operations constraints, or mission planning goals, the plan is modified through replanning. In the rover domain, the planner and executive both query a path-planner for information such as distance between two locations. This information is used to sequence targets and provide estimates on how long traverses will take, which impacts the number of activities that can be placed into the plan.

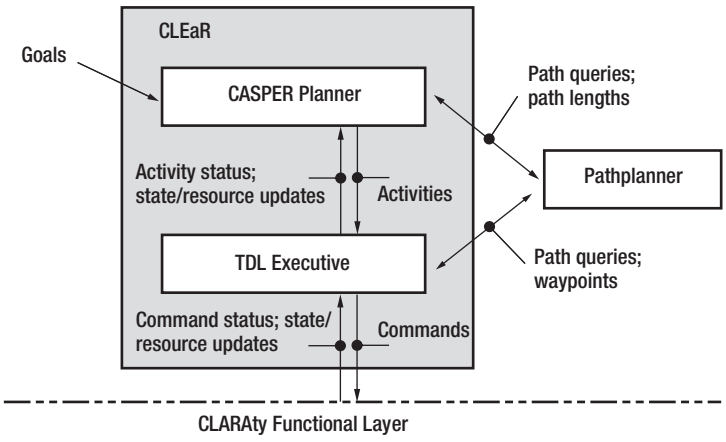


Figure 2. Planning and execution system

Unified Planning and Execution

For automation tasks involving tight time constraints and hard resource and state constraints, both deliberative and reactive types of reasoning are necessary. To date, many approaches have combined the deliberative planning process and the reactive executive in a *black box* fashion. This makes tracking planning constraints difficult for the executive to do during reactive execution of the plan. By unifying the planning and execution process, the passing or access to constraint information is simplified. Although less work has been done to date on unifying the representation, this unified approach also reduces the need to duplicate system model information (hopefully reducing development time, modeling errors, and difficulty of validation) in the planning and executive processes because both access the same runtime information.

We previously identified that a reactive approach can be short sighted, while a deliberative approach can be computationally expensive (time consuming). The continuous planning approach enables CLEaR to replan more frequently thus solving smaller problems and enabling the deliberative process to occur more quickly. CLEaR also attempts to identify the need to re-plan sooner by coupling the execution monitoring with the plan updates and by doing simple reasoning in the executive to predict a likelihood of success or a need to re-plan.

Applied Domains

The CLEaR framework has been applied primarily to two domain areas: onboard decision-making for Mars surface exploration rovers and DSN ground station communication/antenna station automation. To date both of these applications of the CLEaR framework have taken place under research/technology development efforts here at JPL.

Rovers

In conjunction with the CLARAty task [8], the CLEaR framework has been applied to provide the first instantiation of the CLARAty Decision-Layer (DL) [9]. In this application, CLEaR attempts to maximize science return based on a set of high-level goals (objectives) selected by a mission/science team and the model describing the rover's capabilities along with the operations constraints, which could include flight rules. CLEaR does so by performing command sequence generation, execution, monitoring and onboard re-planning of a rover, its science instruments, and resources. This contrasts with the current state-of-the-practice where command sequences are generated on the ground and uploaded to the rover for execution where there is little ability to adapt the sequence in response to unexpected events. Figure 3 shows a picture of Rocky-7 and Rocky-8, the research rovers that CLEaR has been applied to. Figure 4 shows a set of science targets, rocks, a planned path, and the actual path traversed to the science targets used during the 2001 year-end demonstration of the CLEaR software running as part of the CLARAty level one milestone.

DSN

As a component of the Deep Space Station Controller (DSSC) or Common Automation Engine (CAE) technology demonstration effort, CLEaR is used to provide antenna station subsystem command sequence generation, execution, monitoring, and replanning to provide robust execution of downlink communication passes by configuring and commanding the appropriate subsystems. CLEaR generates and modifies the command sequence by dynamically piecing together smaller sequences in order to achieve the desired equipment state. This contrasts with the current state-of-the-practice of executing static rigid sequences, which do not adapt well to unexpected situations like subsystem resets. In this application, CLEaR monitors the progress of the command sequence with the aid of a very capable Fault Detection and Isolation (FDI) component that provides system-wide monitoring and contributes to state estimation [10]. Figure 5 shows a cluster of 34–meter beam waveguide antennas at Goldstone, CA. Each of these antennas was used during the validation phases of the DSSC task.

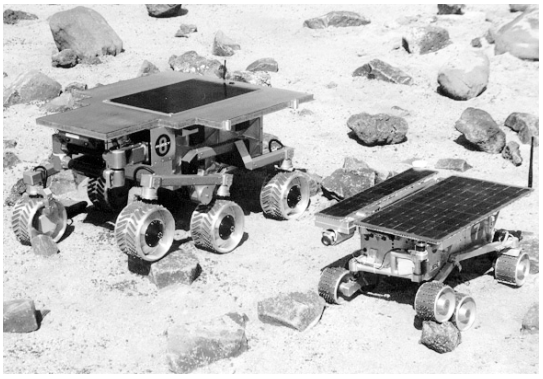


Figure 3. JPL research rovers, Rocky-8 and Rocky-7

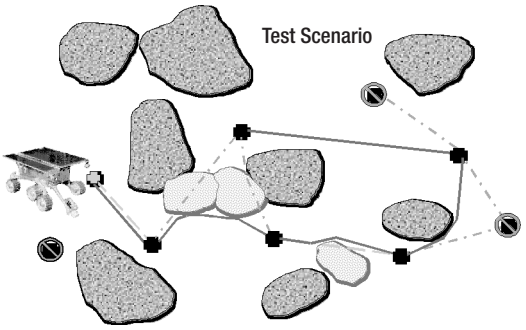


Figure 4. Typical demo scenario

Figure 5. 34-meter beam waveguide cluster at Goldstone, CA

Acknowledgements

The research described in this article was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors would also like to acknowledge the contributions of the ASPEN, CASPER, CLARAty, Rocky-8 and NMC teams. With out their support this work would not have been possible.

References

[1] R. Knight, F. Fisher, T. Estlin, B. Engelhardt, and S. Chien, "Balancing Deliberation and Reaction, Planning and Execution for Space Robotic Applications," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Maui, Hawaii, November 2001.

[2] F. Fisher, D. Gaines, T. Estlin, S. Schaffer, C. Chauinard, "CLEaR: A Framework for Balancing Deliberation and Reactive Control," *Proceedings of the AIPS On-line Planning and Scheduling Workshop*, Toulouse, France, April 2002.

[3] F. Fisher, M. James, L. Paal, B. Engelhardt, "An Architecture for an Autonomous Ground Station Controller," *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[4] T. Estlin, F. Fisher, D. Gaines, C. Chouinard, S. Schaffer, and I. Nesnas, "Continuous Planning and Execution for an Autonomous Mars Rover," 3rd International NASA Workshop on Planning and Scheduling for Space, Houston, Texas, October 2002 (to appear).

[5] S. Chien, R. Knight, S. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

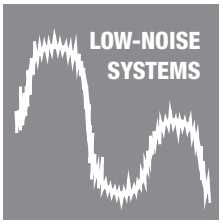
[6] R. Simmons and D. Apfelbaum, "A Task Description Language for Robot Control," *Proceedings of the Intelligent Robots and Systems Conference*, Vancouver, CA, October 1998.

[7] I. Nesnas, R. Volpe, T. Estlin, H. Das, R. Petras, and D. Mutz, "Toward Developing Reusable Software Components for Robotic Applications," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Maui, Hawaii, November 2001.

[8] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "The CLARAty Architecture for Robotic Autonomy," *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[9] T. Estlin, R. Volpe, I. Nesnas, D. Mutz, F. Fisher, B. Engelhardt, and S. Chien, "Decision-Making in a Robotic Architecture for Autonomy," *Proceedings of the International Symposium, on AI, Robotics and Automation for Space*, Montreal, Canada, June 2001.

[10] M. James, R. Mackey, L. Dubon, "An Autonomous Diagnostic and Prognostic Monitoring System for NASA's Deep Space Network," *TMOD Science Program News*, June 2000.



New Ruby Maser Designs Optimized for the DSN

Bob Clauss

Introduction

Multiple Cavity Maser (MCM) designs that use ruby as the amplifying material are being analyzed with a new computer technique. Two recent publications entitled "A Spin System Model for Coupled Cavity Masers," [1] and "The Paramagnetic Ground State of Ruby – Revisited," [2] describe the theory and the analysis tool development in detail. The ruby spin system model was added to a modified rectangular waveguide mode-matching program [3]. The result of this work gives the maser designer a computer analysis tool that enables optimization of multiple-cavity ruby maser designs. No such tool was previously available and cavity maser builders relied on cut-and-try techniques that did not produce an optimized design.

Cavity maser designs with maximally flat amplification responses covering 31.8 gigahertz (GHz) to 32.3 GHz (500 megahertz [MHz] bandwidth), 8400 MHz to 8450 MHz (50-MHz bandwidth), and 2270 MHz to 2300 MHz (30-MHz bandwidth) are now being produced. These bandwidths match the deep-space-to-Earth allocations at Ka-band and X-band, and exceed the 2290 MHz to 2300 MHz allocation at S-band to support Very-Long-Baseline Interferometry (VLBI) for spacecraft navigation. Four-stage amplifier designs show net gains of 33 decibels (dB) to 44 dB with gain flatness better than +/- 0.5 dB. These new masers are designed to operate in the same 4-kelvin (K) Gifford-McMahon cycle cryogenic refrigerators that are currently used in the Deep Space Network (DSN) for cooling high-electron-mobility transistor (HEMT) amplifiers.

Use of these new MCM designs, with 4.5-K module noise temperatures at Ka-band, 2 K at X-band, and 1.3 K at S-band, will reduce the DSN's receiving system operating noise temperature and improve the DSN's sensitivity. Increasing the effective collecting area of the antennas can also be used to increase the DSN's sensitivity. For example, a 20% noise temperature reduction gives a 25% sensitivity improvement, the same as arraying each 70-meter (m) diameter antenna with a 35-m diameter antenna having the same efficiency and noise temperature. The cost of implementing the added antenna area is much higher than the cost of implementing these masers.

The cost of implementing the added antenna area is much higher than the cost of implementing these masers

Why Develop Cavity Masers

The new cavity maser designs will provide the lowest possible noise temperatures at S-band, X-band, and Ka-band. These masers are least affected by Radio Frequency Interference (RFI) of any low-noise amplifier because they do not generate intermodulation products in the presence of RFI. They are rugged amplifiers that are not susceptible to microscopic failures, voltage transients, or burnout, even when irradiated with several watts of signal frequency power.

Cavity masers were used in the DSN between 1960 and 1965 to support missions at 960 MHz and planetary radar at 2,388 MHz [4]. These masers had very narrow bandwidth (0.75 MHz at 960 MHz and 2.5 MHz at 2,388 MHz). The potential of cavity masers was not recognized at that time. For example, A. E. Siegman, in "Microwave Solid State Masers" [5] (published in 1964) indicated: 'It is the author's opinion that the cavity maser is now essentially obsolete and that, as stated at the beginning of Chapter 7, "any maser development at the present time should begin immediately with the traveling-wave approach, avoiding the temptation to build a cavity maser first 'just for practice.'"

Reasons for developing traveling-wave masers included the promise of wide bandwidth and wide tuning ranges. The DSN needs of that period (1965 through 1980) included the support of:

- Planetary radar at 2388 MHz
- Deep space missions at 2295 MHz
- Manned-spaceflight missions between 2270 MHz and 2290 MHz
- X-band deep space allocation between 8400 MHz and 8500 MHz
- X-band planetary radar at frequencies between 8500 MHz and 8550 MHz
- Very-long-baseline interferometry (VLBI) at 7840 MHz.

Today, the plans for future deep space missions can best be met with the new multiple cavity-maser designs

Traveling-wave masers were developed to support these needs, and reflected-wave masers were developed to support radio astronomy applications [4]. Today, the plans for future deep space missions can best be met with the new multiple cavity-maser designs. Wider bandwidth applications in the DSN, such as continuum radio astronomy observations, can be supported with HEMT-based systems.

Multiple-Cavity Designs Description

The amplifying element in each MCM stage is a ruby rectangular prism. Each 32-GHz MCM stage contains a small piece of ruby that is about 1/11th of a carat (one carat is equal to 200 milligrams) with dimensions of 2.4384 millimeter (mm) by 1.8288 mm by 1.016 mm. Rubies of this size and shape could make an attractive supplement to a piece of jewelry with a larger central diamond, if the ruby was red. Pink ruby is used in masers because the optimum chromium oxide percentage (0.035 % to 0.05 %) gives the best maser performance. Red ruby used in jewelry has about 0.5% chromium oxide, and would not work as a maser material.

The chromium oxide concentration determines the spin density in ruby. The ruby in each 32 GHz maser stage contains 100 quadrillion (100,000,000,000,000,000) spins, each being able to absorb or emit one quantum (hf) of energy, where h is Planck's constant (6.626176 E-34 joule seconds) and f is the frequency in hertz. This limits the Continuous Wave (CW) output power of the maser to the order of about one microwatt.

The amplifying element in each stage of an X-band maser (8400 MHz to 8450 MHz) is much larger than the 32 GHz MCM. The size scales approximately inversely with frequency when ruby-filled waveguide cavities are used. The X-band ruby dimensions are 9.276 mm by 6.957 mm by 3.865 mm (5 carats) if scaled from the 32 GHz design. Other considerations, such as pump frequency resonance, cause the designer to use slightly different dimensions than those obtained by scaling. Final design work is yet to be completed.

The supply of high quality maser ruby currently on hand is sufficient to build hundreds of cavity masers. Ruby previously purchased for the construction of S-band and X-band traveling-wave masers and K-band (18-25 GHz) reflected wave masers can be reused to build the cavity masers. "Old" masers should not be discarded surplussed when removed from the DSN. Ruby from the traveling-wave and reflected-wave masers implemented in the DSN during the past 35 years should be saved for future use.

A Ka-band MCM design was described in a TMOD Progress Report [6]. The designs optimized for Ka-band and X-band would appear similar to those described in this report, but the performance of the optimized design is far superior to that of the earlier designs, as discussed below.

Each MCM stage consists of a ruby-filled waveguide cavity, with empty waveguide

signal- and pump-coupling cavities adjacent to the ruby-filled cavity. A pump reject filter that confines the pump energy to the maser structure precedes the signal-coupling cavity in the signal waveguide. Pump energy is fed to the maser through a pump-frequency waveguide at the other end of the maser structure. The signal to be amplified is directed to and from the cavities by a circulator. The amplified signal returns to the circulator through the same waveguide used to transfer the signal into the maser structure (a reflection-type amplifier). Figure 1 shows this arrangement in block diagram form. Isolators are used between stages when multiple stages are used. Figure 2 shows a cutaway drawing of a single-stage 32-GHz maser structure.

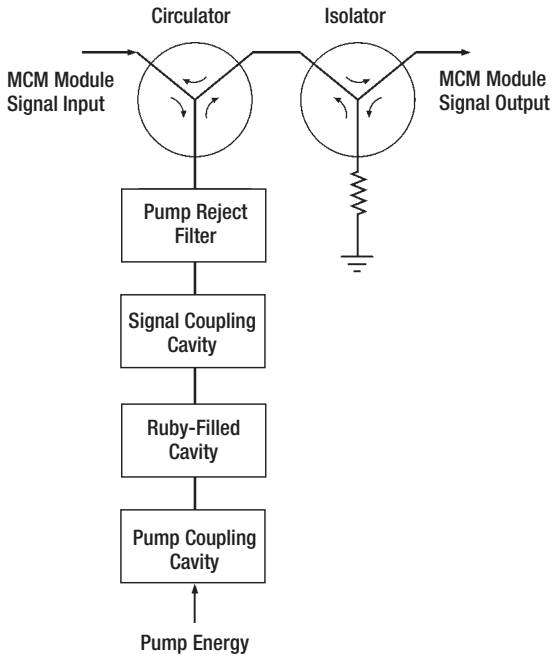


Figure 1. MCM single stage

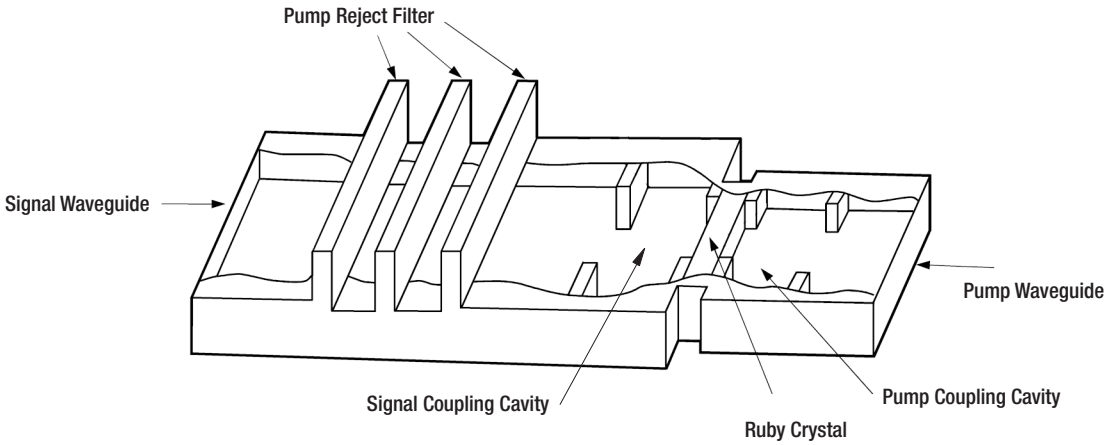


Figure 2. MCM single stage 32 GHz maser structure, perspective view

Gain, Bandwidth, and Noise Temperatures

Figure 3 shows the amplification response of the 32-GHz MCM stage. Figure 4 shows the response of a six-stage amplifier. The flat response is the result of optimizing the ruby spin system negative reactance with the reactance of the microwave cavities. The positive resistance and reactance of the unpumped ruby spin system becomes negative (negative resistance and negative reactance) when pumped. The negative resistance causes amplification and the negative reactance changes the amplification bandwidth.

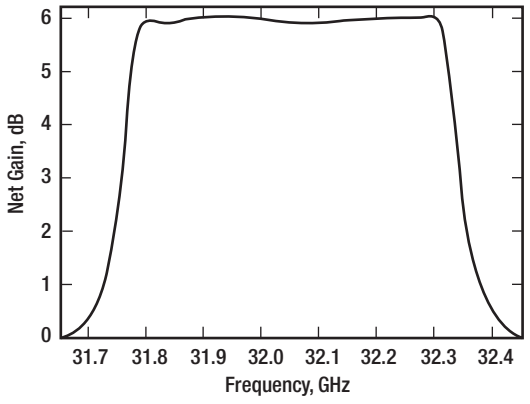


Figure 3. Amplification frequency response of single stage Ka-band MCM stage

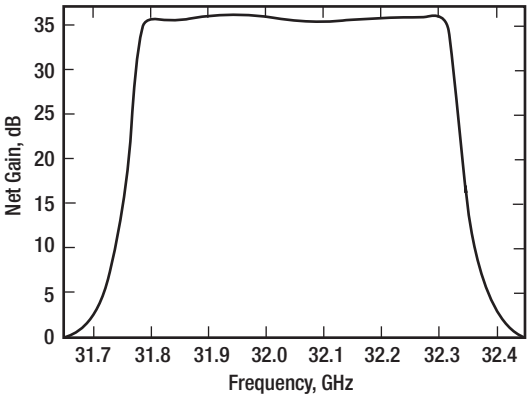


Figure 4. Amplification frequency response of six-stage Ka-band MCM

Figure 5 shows the unpumped response of the MCM stage. The difference in the shape of the unpumped ruby spin system loss response and the pumped amplification response highlights the positive and negative reactance effects. Broadband structures, such as traveling-wave masers, do not exhibit this effect: the pumped amplification response shape is the mirror image of the unpumped ruby spin system loss response shape in a broadband maser structure.

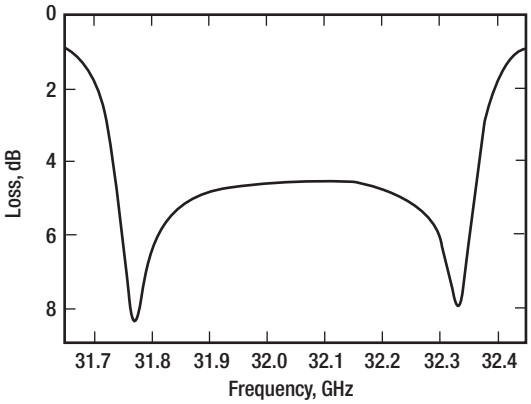


Figure 5. Frequency response of unpumped ruby-spin-system loss

X-band and S-band MCM designs using waveguide cavities were analyzed and optimized. The approach used resulted in a flat amplification response of 50 MHz for the X-band design and 30 MHz for the S-band design (Figures 6 and 7). The S-band cavity design for a practical amplifier would use smaller resonant elements than waveguide cavities. The current analysis tools depend upon the use of rectangular waveguide. The optimum values for cavity loaded-Q and coupling factors can be determined[1] . A structure using coaxial, stripline, or comb-type resonators can be designed having the same loaded-Q values and coupling factors.

The flat amplification response of a single stage remains flat in a multi-stage amplifier and the skirts of the response become steeper. If a somewhat flat response of a single stage has a gain ripple of 0.1 dB, a four-stage amplifier will have 0.4 dB of gain ripple.

The noise temperatures of the MCMs will not vary measurably within the flat response of the amplifier because the cavity and component losses change little within this frequency range. Previous DSN traveling-wave masers experienced noise temperature variations within the maser's bandwidth and tuning range.

The noise temperatures of the MCMs calculated at the cryogenic input terminal of the amplifier assembly are identified as the MCM "module" noise temperature. This practice is consistent with the use of module noise temperatures for HEMTs. Components added between the module and the antenna feed system increase the system noise temperature. Because the MCM assembly contains circulators and isolators, there is no need to add isolators between the MCM and the feed system components. The MCM acts as a preselector filter so there is no need to add filters between the MCM and the feed system components. Reduced noise temperatures result.

The multi-stage Ka-band, X-band, and S-band MCM design net gain values were analyzed to be 36 dB, 44 dB, and 40 dB respectively, when cooled to 4 K. The calculated module noise temperatures of these designs, when cooled to 4 K, are 4.5 K at Ka-band, 2 K at X-band, and 1.3 K at S-band. Use of refrigerators that cool the maser ruby to a physical temperature below 4.5 K will reduce the MCM module noise temperature as shown in Table 1.

[1] Loaded Q is a term that considers quality with external circuitry connected to the resonant circuit; unloaded Q considers quality of the resonant circuit without external circuitry being connected to the circuit. Our amplifier's performance depends upon the unloaded Q of the circuit *and* the effect of the external circuit connected to the resonant circuit (multiple waveguide cavities), so the "loaded Q" term is used.

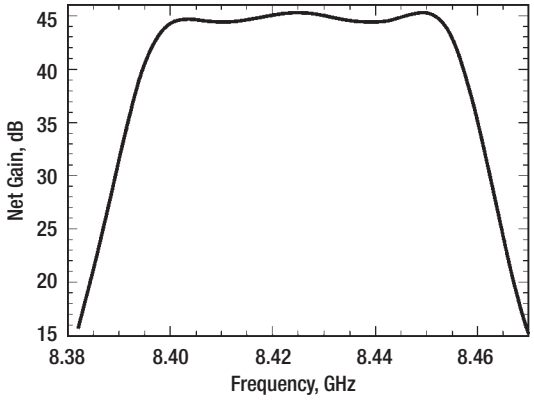


Figure 6. Amplification frequency response of 4-stage X-band MCM

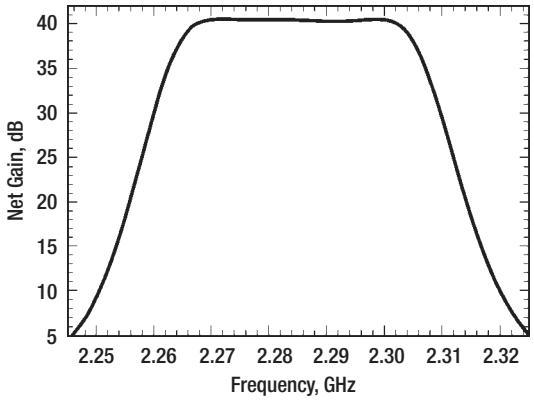


Figure 7. Amplification frequency response of 4-stage S-band MCM

Conclusion

The new analysis tools and resulting MCM designs offer significant improvements in sensitivity for all DSN antennas, covering the deep-space-to-earth allocations at Ka-band, X-band and S-band. These MCM systems can be implemented with a very high return-on-investment.

Acknowledgement

Many have contributed to the development of ruby masers for the DSN since 1955. Our leader and teacher in the early years was Dr. Walter H. Higa. Dr. James S. Shell is the current leader who developed the new analysis programs, enabling cavity-maser designs far superior to those developed in the past. The author of this article, with 43 years of maser and DSN technology development experience, was pleasantly surprised recently by the performance improvements enabled by these new analysis tools. These new analysis tools are impressive and unique contributions to development of ultra-low-noise masers for the DSN.

References

[1] J. Shell, "A Spin System Model for Coupled Cavity Masers," *The Interplanetary Network Progress Report* 42-151, Jet Propulsion Laboratory, Pasadena, California, to be published November 15, 2002.

[2] J. Shell, "The Paramagnetic Ground State of Ruby — Revisited," *The Interplanetary Network Progress Report* 42-150, Jet Propulsion Laboratory, Pasadena, California, August 15, 2002.

[3] D. J. Hoppe, "Modal Analysis Applied to Circular, Rectangular, and Coaxial Waveguides," *The Telecommunications and Data Acquisition Progress Report 42-95*, July – September 1988, Jet Propulsion Laboratory, Pasadena, California, pp.89 – 96, November 15, 1988.

[4] James S. Shell, et al., "Ruby Masers for Maximum G/Top," Proceedings of the IEEE, Vol. 82, No. 5, pp. 796 – 810, May 1994.

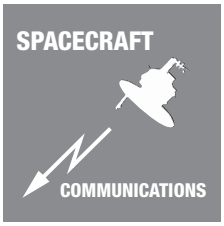
[5] A. E. Siegman, "Microwave Solid-State Masers", *McGraw-Hill Electrical and Electronic Engineering Series*, New York: McGraw-Hill Book Company, 1964.

[6] J. S. Shell and R. C. Clauss, "A 32-Gigahertz Couple-Cavity Maser Design," *The Telecommunications and Missions Operations Progress Report 42-142*, April – June 2000, Jet Propulsion Laboratory, Pasadena, California, pp.1 – 22, August 15, 2000.

Physical Temperature, K	Module Noise Temperature, K		
	Ka-band	X-band	S-band
4.5	4.9	2.2	1.5
4.0	4.4	2.0	1.3
3.5	4.0	1.8	1.2
3.0	3.6	1.6	1.0
2.5	3.2	1.4	0.9
2.0	2.8	1.2	0.7
1.5	2.4	1.0	0.5
0.0	1.6	0.4	0.1

Note: Module noise temperature values for a hypothetical 0.0 K refrigerator are based on the Quantum Noise Limit, preceded by a circulator with 0.15 dB loss.

Table I. MCM module noise temperatures vs. physical temperature



Inflatable/Self-Rigidizable Reflectarray Antenna

Houfei Fang,
Michael Lou,
John Huang

Introduction

With the advancing of space sciences, future space missions demand larger and larger apertures with very low launching masses and volumes. Space inflatable technology will revolutionize future space structures to accommodate these demands. This article discusses an inflatable/self-rigidizable structure for a 3-meter Ka-band (32 GHz) reflectarray antenna [1]. The Radio Frequency (RF) component of this antenna is a flat membrane reflectarray with many copper patches that are illuminated by an offset feed horn. The membrane is supported by an inflatable/self-rigidizable boom/frame structure. Booms of the inflatable/self-rigidizable structure can be flattened. The flattened booms are rolled up on two mandrels, and the membrane is rolled up on a composite cylinder when the antenna is in stowed configuration. After the antenna is launched into space, it is inflation-deployed and the dynamics of the deployment is controlled by a deployment control system. Compared to other types of deployable antennas, this antenna offers a much larger aperture with extremely lightweight and high-package efficiency. This antenna uses a flat surface instead of a parabolic surface as the RF component. A flat "natural" thin-membrane surface is much easier to accomplish and maintain than a curved "non-natural" parabolic surface. It is also believed that for a long-term space mission, a flat surface is more reliable than a thin-membrane parabolic surface.

Movie Screen Antenna

Due to the deployment process, this antenna is also called the "movie screen" antenna. Deployment involves the pressurization and unrolling of two inflatable booms. It involves virtually no moving parts compared to other mechanically-deployed antennas. This flat design clearly has the advantages of less weight, lower development cost, and better deployment reliability.

The most important component of the antenna is the RF surface of the membrane. The large circular portion of the membrane carries approximately 200,000 RF patches. Figure 1 gives the close-up view of the RF patches. For circular polarization, these patches are identical, but they are rotated at different angles [2] to compensate for the phase difference incurred from the different path lengths traveled by the waves from the feed horn to other elements. The membrane's outer portion around the RF section

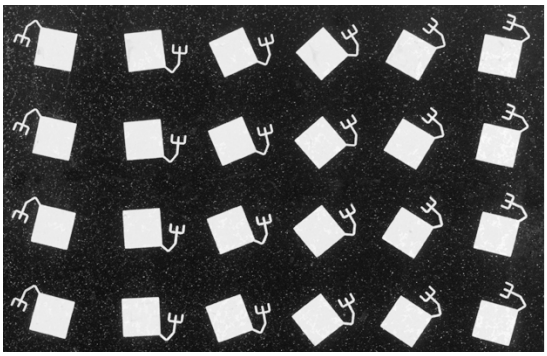


Figure 1. Close-up of the RF patches

connects the RF area to the catenary system, which is then attached to the supporting structure by constant force springs. The whole supporting structure is designed to hold and tension the membrane, to avoid wrinkling it and to keep the membrane flat.

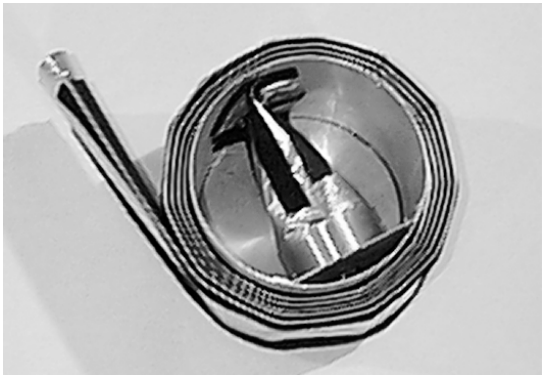
Inflatable/Self-Rigidizable Boom

"Inflatable" means the structure is deployed by pressurization. After a structure is deployed, pressure still has to be kept inside the structure to maintain its rigidity. Due to material imperfections and small damages caused by micrometeoroids, small leaks are unavoidable. Large amounts of make-up gas would have to be carried to space for a long-

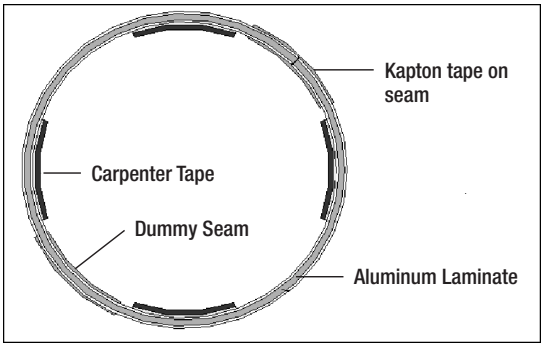
term mission, which would be very costly or even unrealistic.

A new inflatable/self-rigidizable method, namely Spring Tape Reinforced (STR) aluminum laminate boom, has been developed by this research for the "movie screen" antenna [3]. This technology does not require pressure inside an inflatable structure after deployment. Figure 2a shows a deployed 5-meter long STR aluminum laminate boom. Figure 2b shows the STR boom rolled up on a 165-mm-diameter mandrel. Figure 2c is the cross-sectional view of the STR boom. A typical STR boom consists of a tube formed with aluminum laminate. Four spring tapes are attached to the inside wall of the tube in the axial direction. At this time, the commercially available stainless steel measuring tapes, commonly known as carpenter tapes, are used. With a wall thickness less than 0.1 millimeters, a STR boom can be easily flattened, rolled up (or folded up), and deployed by a relatively low inflation pressure. The buckling capability of a STR aluminum laminate boom is very high, due mainly to the high modulus of elasticity and curved cross-sectional profile of the spring tapes. Spring tapes are very effective in resisting inward buckling, and the aluminum laminate wall is very stable in resisting outward buckling. Therefore, these two components effectively complement each other in resisting local crippling of the boom. Several 5-meter long, 7.6-centimeter-diameter booms have been assembled and tested. The weight of each boom is only 0.9 kilograms. The axial buckling load carrying capability of this kind of boom can reach 74 kilograms (with pin-pin boundary conditions).

(a)



(b)



(c)

A new inflatable/self-rigidizable method has been developed by this research for the "movie screen" antenna

Figure 2a. A deployed 5-meter long STR aluminum laminate boom

Figure 2b. 5-m long boom rolled up on a 165-mm-diameter mandrel

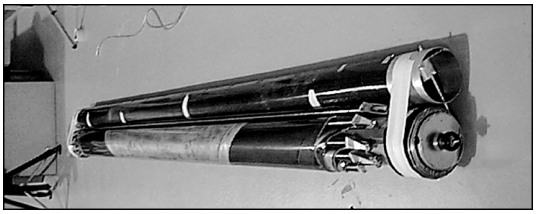
Figure 2c. Cross-sectional view of the STR boom

Engineering Model and Deployment

After a few years of effort, an engineering model of the 3-meter inflatable/self-rigidizable reflectarray antenna has been developed. Figure 3a shows the antenna in the stowed configuration, and Figure 3b shows the antenna in the deployed configuration.

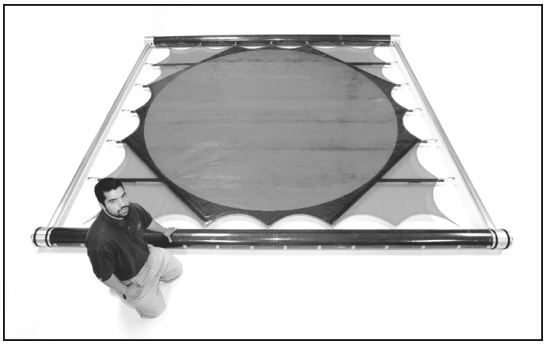
One of the most important tasks of this study is to demonstrate the deployment process of this engineering model. In order to have a smooth deployment, a structure was designed and built to support the antenna and eliminate some of the gravitational effects during deployment. This supporting structure is composed of two tracks and five pairs of moving arms as shown in Figure 4. Every moving mandrel is attached by a roller that rotates on a track to eliminate resistance during deployment. Five pairs of arms were all initially in a lowered position. During deployment, each pair of arms raises up to support one of the cross bars right after the cross bar separates from the bundle. Arms are actuated by pneumatic cylinders.

Several deployment tests were successfully conducted and Figures 4 show the process of the deployment, from packaged, to partially deployed, to fully deployed.



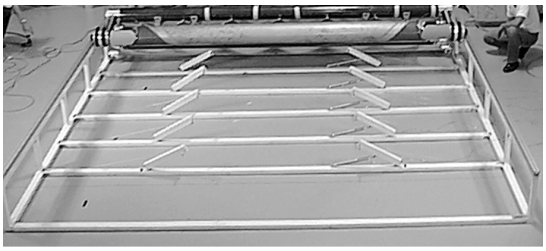
(a)

Figure 3a. Antenna in the stowed configuration

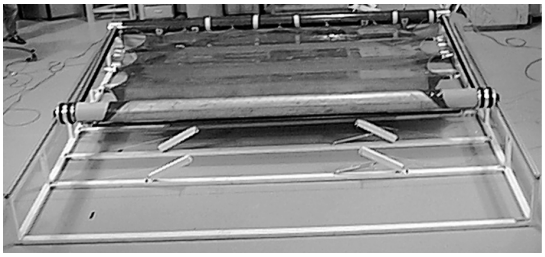


(b)

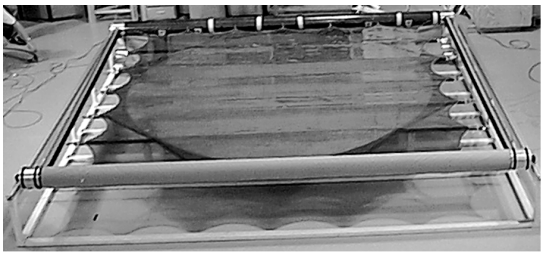
Figure 3b. Antenna in the deployed configuration



(a)



(b)



(c)

Figure 4. Demonstration of the deployment

Conclusions and Future Tasks

An engineering model of the 5-meter inflatable/self-rigidizable reflect array antenna has been developed, and the inflation deployment of this antenna has been demonstrated. In order to prepare the inflatable/self-rigidizable reflectarray antenna for space missions, several tasks remain to be accomplished. The first is constructing the antenna launch restraining system. During launch, the antenna has to withstand high acceleration, vibration, and acoustic impact. In order for the antenna to survive the launch, a restraining system is essential to hold the packaged antenna in place. The second task is investigating the structural thermal distortion. The space thermal environment is very harsh and could cause distortion of the inflatable structure as well as of the RF membrane. The third task is studying the effects of damping on the antenna's dynamic responses to spacecraft maneuvering. The sensitivities of damping locations will be investigated and extra damping will be applied to those most effective places. The fourth task is performing in-space deployment dynamics analysis. Due to the gravity and air damping, deployment dynamics test of a large inflatable space structure on the Earth is very difficult and costly. Deployment dynamics analysis is therefore a necessary task for a space mission. Developing a larger aperture antenna is also under consideration.

Acknowledgments

The authors wish to thank Lih-Min Hsia, Grigor Kerdanyan, Pierre Leung, Ubaldo Quijano, Elvis Garay, Giovany Pelaez, Jiao Tian, Agop Derderian, and Joel Rodriguez, all of California State University at Los Angeles for their contributions to this research effort.

References

[1] H. Fang, M. Lou, J. Huang, L. Hsia, and G. Kerdanyan, "Design and Development of an Inflatable Reflectarray Antenna," *The IPN Progress Report*, http://ipnpr.jpl.nasa.gov/, Issue 42-149, May 2002.

[2] J. Huang and R. J. Pogorzelski, "A Ka-Band Microstrip Reflectarray with Elements Having Variable Rotation Angles," *IEEE Trans. On Antennas & Propag*. Vol. 46, pp. 650-656, May 1998.

[3] M. Lou, H. Fang, and L. Hsia, "Development of Space Inflatable/Rigidizable STR Aluminum Laminate Booms," AIAA 2000-5296, presented at Space 2000 Conference & Exposition, Long Beach, California, 19-21 September 2000.

In-Situ Site Knowledge System

Meemong Lee and Richard J. Weidner

Introduction

The In-Situ Site Knowledge System (ISSKS) described in this paper is designed to provide a progressive automation framework so that the transition from human-based operations to automated operations can be incrementally experimented and validated. The term "automation" in this context refers to a process of formulating a higher-level abstraction that can be machine-interpreted to a set of multiple low-level commands. The level of abstraction increases as the capability of the low-level commands improves.

The ISSKS is developed in three tiers. The first tier is composed of site property modeling and synthesis to construct a virtual in-situ mission environment. The second tier is composed of sensor system and mobility platform models for simulating science and engineering data products during an in-situ exploration. The third tier is composed of a set of operation interface protocols providing site information and mission data products to a wide range of autonomous in-situ operation activities including rover navigation, hazard detection, and instrument operation planning.

In collaboration with the Mission Simulation Facility at Ames Research Center and the CLARAty (Coupled Layer Architecture for Rover Autonomy) team at JPL, the ISSKS team is developing an autonomous science validation environment. The ISSKS provides a virtual in-situ site to test operation scenarios, synthetic measurements to test autonomy algorithms, and an operation interface to test on-board instrument software.

Virtual Site

A virtual site refers to a computationally constructed area that can be used for the site determination analysis by enabling simulation of the relevant in-situ mission operations. The physical properties of a virtual site can be set according to the desired analysis. The physical properties for safety considerations may be limited to the surface structure while those for science return may require material composition. Thus, it is important to modularize the physical properties to allow an application-driven virtual environment synthesis.

The ISSKS categorizes the site properties into three types: terrain, material, and texture. The synthesis process of the three site-property types is discussed in this section. Emphasis of the virtual site synthesis process has been on implementation of an evolutionary synthesis framework that can explore a wide range of knowledge uncertainties and progressively introduce more complex site phenomena. The progressiveness applies to the site knowledge as well as the representation resolution of the knowledge.

The ISSKS provides a virtual in-situ site to test operation scenarios, synthetic measurements to test autonomy algorithms, and an operation interface to test on-board instrument software

Terrain Property Synthesis

Safety hazards are generally expressed with maximum elevation, high-relief surface features, rock abundance, and dust density [1]. The terrain property represents geometric structure of a site and it is described as a combination of a global surface elevation model and a set of surface features. The global surface elevation model can be either artificially synthesized based on the slope distribution and roughness specification or derived from a morphology database for a desired terrain type.

The virtual site maintains a morphology database that can be used to derive the digital elevation models of various global features. The morphology database integrates the gazetteer created by the NASA Planetary Data Systems (PDS) based on the Viking Orbiter

data. The Gazetteer consists of a wide range of geomorphology types, such as Catena, Chaos, Chasma, Mensa, Mons, Planum, Valles, etc. Figure 1 shows an image of AURORAE Planum and a digital elevation model (DEM) constructed from the image by

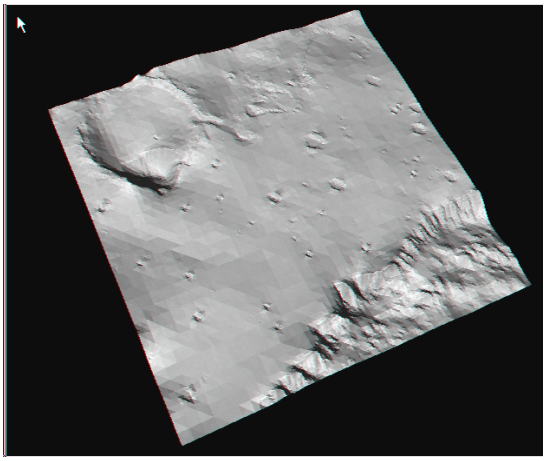


Figure 1. Derived surface elevation using photo-clinometry algorithm.

applying a photo-clinometry algorithm developed at Mission Simulation and Instrument Modeling group at Jet Propulsion Laboratory.

In addition to the global morphology, various local terrain characteristics can be described including rocks and craters. In order to handle the large dynamic range, a set of feature object classes is defined. A feature object class is an abstraction of the feature that can be dynamically instantiated into multiple specific feature objects. The abstraction constructs a static feature template and its instantiation creates a specific feature by dynamically morphing the feature template.

The virtual site implements Golembek's crater model [2] that divides a crater into four piece-wise linear-sloped regions. Interleaving of the surface evolution and crater addition can be used to represent different aging of the craters. The surface evolution step integrates the craters with the background terrain as the roughness is added to the surface. The surface object keeps a crater list and updates it when a crater is added, with location and size information. The crater list is used to maintain desired crater density and intercrater distance.

Rocks can be added to the site after the site surface has been evolved to the highest spatial resolution. The ISSKS implements a rock object as a 2.5D shape whose size can be manipulated in three directions (x, y, z). The x and y directional morphing is performed in a manner similar to the dynamic morphing of craters and the z directional manipulation is performed by assigning a random burial depth for each rock. As each rock is placed on a terrain, its orientation can be specified as well. Figure 2 illustrates a view of a synthetic site that was created by merging a digital elevation model constructed from a natural site and synthetic rocks.

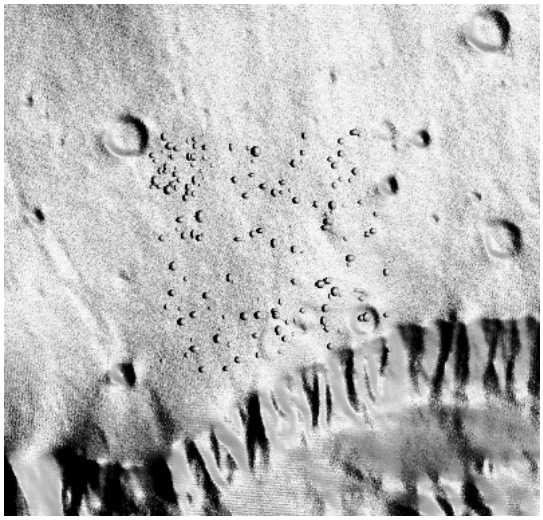


Figure 2. A view of the merged terrain

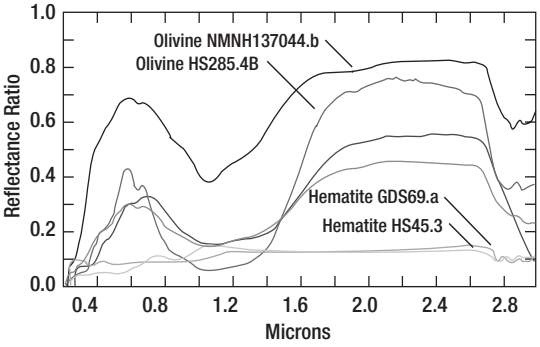
Material Synthesis

The material property of a site is very important to understand the geologic evolution process of Martian surface [4]. The mineral content of a rock may indicate its origin as well as its weathering process. The ISSKS allows material assignment to the site by providing a comprehensive mineral database and a mixture object class. The mineral database contains ~500 minerals covering the wavelength range between 0.3 μ and 3.0 μ in 0.1 μ interval.

The material property of a site is described as surface-type-specific mineral mixtures where a mineral mixture indicates physical mixing of multiple minerals. The physical mixing synthesizes an area represented by a data sample being composed of multiple minerals. Thus, the material property synthesis is sensitive to the spatial resolution. The ISSKS currently supports a Mixture object with

three mixture types "linear" (equal composition), "random" (a random composition) or "weighted" (a specific proportion). The Mixture object allows minerals to be added as members and generates a mixed mineral based on the specified mixing type.

The radiometric information of a site can be constructed over a selected wavelength range for a desired spectral resolution by synthesizing rock mixtures and soil mixtures. The mixture type *random* applies uniformly distributed random weights over the mixture member minerals to generate a specific rock mixture spectrum. Thus, one mixture can be employed to synthesize a wide range of spectral variations. Figure 3 illustrates a set of soil mineral spectra used in creating soil mixtures.



The three dimensional multi-spectral dataset is synthesized in three steps. First, a feature mask is composed from the rock list and the crater list, indicating for each pixel location whether it is a part of a rock, a crater, or regular surface. For the nonrock pixel, the soil mixture spectrum is generated by randomly mixing the soil mineral members. For the rock pixel, the rock mixture spectrum is generated by randomly mixing the rock mineral members.

Texture Synthesis

The texture property is somewhat ambiguous in terms of its corresponding physical phenomena due to the spatial resolution dependency of the property. For example, a rocky terrain may be modeled with individual rocks for close-up observation or as a rock texture for remote observation. In general, the texture property in ISSKS is designed to capture the site properties that are not explicitly represented by the terrain or the material due to the spatial or spectral resolution.

The ISSKS currently implements the texture property in an application specific manner. In a rover traverse application, the texture property includes the soil type information relevant to the surface interaction with a rover (e.g., slippage, sinkage) while in an instrument observation application, it includes the surface roughness property relevant to the light (or signal) scattering patterns (i.e., photometric or backscattering effect).

Measurement Simulation

Site knowledge is constructed from analyzing various instrument measurements gathered from a wide range of observation platforms. Terrain properties are estimated by measuring thermal inertia (rock abundance), radar absorption (roughness), stereo imaging (range), and laser ranging (elevation). Material properties are estimated with color filters and near IR spectrometers. Soil mechanical properties require various sensors yet to be developed.

The term "measurement" refers to site phenomena captured by a sensor in a specific operation state at a specific time. The ISSKS implements the measurement simulation for multiple observation platforms and instrument types. The observation platforms include a lander, a rover, and an in-situ mobility device (e.g., panorama gymbal). The instrument types include passive instruments (e.g., visible and IR cameras) and active instruments (e.g., laser or radar altimeter, and synthetic aperture radar). The observation platforms introduce platform-unique geometric projection while the instrument types introduce instrument-specific signal detection and propagation response.

The passive instruments detect reflected sunlight while the active instruments detect reflected signals after transmitting them. In this section, measurement simulation is described with respect to the passive instruments in three aspects: photometric phenomena created by the Sun, geometric projection imposed by the sensor's viewing geometry, and instrument response governed by operation control and instrument system characteristics [3].

Figure 3. Mineral members for soil mixture generation

The ISSKS also implements a dataframe object for physical-to-instrument space coordinate transformation reflecting viewing direction, field-of-view, image frame size, and detector resolution

Photometric Phenomena

The photometric phenomena in this context indicate the intensity variation introduced by the Sun including global brightness, shading and shadowing. Global brightness is important for instrument operation parameter setting while shadowing and shading are important for image analysis algorithms. The ISSKS currently implements a Sun object and a sky object for photometric phenomena synthesis. The Sun object is used to compute solar irradiance property and apparent direction from the site and the sky object is used to compute diffusion from the atmospheric layer, dust density, and backscattering. The shadowing and shading phenomena are synthesized as a dynamic property of the terrain since the phenomena is caused by the interaction between the terrain object and the Sun object.

Shadowing refers to obscuration of sunlight while shading refers to the intensity variation caused by the incidence angle of the sunlight (the angle between the surface normal and the Sun's direction). As the Sun's direction changes, a shadow mask is generated first identifying the regions where the sunlight is obscured. For the regions that are not obscured, shading coefficients are computed as a function of the incidence angle at the local surface. The shadow mask generation is performed pixel-by-pixel by tracing the light path until the end of the path is reached or an obstacle is identified. During the shadow mask generation, the elevation map of the terrain is temporarily rotated so that the Sun's direction is parallel with the line direction of the terrain. The alignment with the Sun's direction prevents the jaggedness introduced from accessing the elevation map in a slanted line direction. Current implementation assumes the reflection to be all directional (diffused surface reflection). The texture property can be used to represent the surface with exceptional photometric properties.

Geometric and Radiometric Projection

The geometric projection refers to an observation platform dependent rendering process and the radiometric projection refers to an instrument type dependent spectral sampling process. The ISSKS provides Cartesian and panorama coordinate objects for representing site information so that the rendering process can be performed effectively for a given observation platform.

For the panoramic stereo camera, a panorama site is constructed for each side of the stereo camera. The panorama site represents the terrain and texture maps as panorama map objects. The panorama map object employs camera-centric azimuth and elevation angles as the coordinate reference. The panorama map object provides the coordinate transforms from/to Cartesian map objects. For the Entry, Descent, and Landing (EDL) descent camera, the terrain and texture objects remain in the Cartesian map coordinate.

The ISSKS also implements a dataframe object for physical-to-instrument space coordinate transformation reflecting viewing direction, field-of-view, image frame size, and detector resolution. The coordinate transformation involves pixel-by-pixel line-of-sight computation and intersection with the terrain object or the sky object. The dataframe object can be used as a base class for deriving instrument-specific dataframe objects. For example, a stereoframe object is derived from the dataframe object for the stereo panorama camera and it is applied to represent the two imaging space coordinate systems of the left and right cameras.

As discussed in the material property, surface albedo information can be computed by applying radiometric projection to the material spectra of the intersected location. Radiometric projection refers to resampling of the spectral information over the selected wavelength region following the instrument's spectral response characteristics. Depending on the instrument type, more than one wavelength region may be selected.

The general brightness of the surface and shadowing is correlated with the elevation angle of the Sun. The backscattering brightness of the horizon is computed from the surface brightness and inversely propagated. Color is obtained by applying the radiometric projection of the soil and rock mixture materials.

Instrument Response Characteristics

The geometric and radiometric projection described above represents a noise-free sensor data product. The distortions due to various instrument response characteristics are divided into three types: spatial, spectral, and amplitude distortion. The spatial distortion type includes nonlinear pixel size across the field-of-view and blurring from the point spread function of the optics as well as from the platform motion during exposure. The spectral distortion combines the quantum efficiency of a detector and the transmission efficiency of optics. Finally, the amplitude distortion includes light transfer curve, thermal noise, and read noise that alters the linearity of the physical intensity.

Measurement simulation of each instrument requires distinct distortion properties either from the instrument itself or from the way the instrument is operated. For example, a wide-angle camera has a large spatial distortion from the nonlinear pixel coverage, while a descent imager introduces blurring from spacecraft motion during the imaging. Thus, it is important to modularize the noise property simulation so that relevant noise properties can be selectively added to the measurement simulation.

The above three aspects of measurement simulation are directly linked with the planetary science data processing steps: the instrument calibration process removes the instrument system noise and restores radiometric properties; the image-to-physical space conversion reverses the geometric projection, and image intensity to surface radiance conversion removes the photometric effects. Therefore, it is important to perform measurement simulation with sufficient accuracy for science data processing validation

An instrument system that is composed of multiple devices, such as a panoramic stereo camera system, is simulated via a super object. Figure 4 illustrates an example data product of a panoramic stereo camera. A stereoframe object, derived from a dataframe object is employed for representing the image space of each camera reflecting viewing geometry and spatial resolution. The rendering process involves terrain rendering and sky rendering. The terrain process starts from the bottom of the scene finding the intersection with the terrain object and proceeds to the top until no intersection is found. The sky area is filled by combining the brightness information from the sky object (as described in the photometric property) and the backscattering introduced by the rendered terrain.

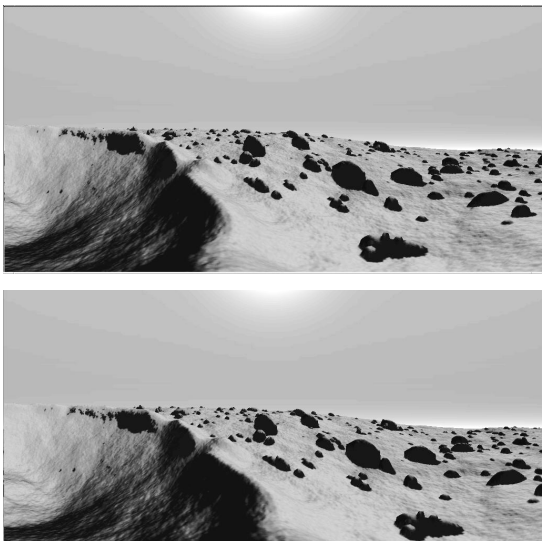


Figure 4. An Example Stereo Camera Frame

Operation Interface

The operation interface tier is composed of a set of operation interface protocols providing site information and mission data products to a wide range of autonomous in-situ operation activities. The operation interface protocol is designed so that a spacecraft system or a rover can virtually perform in-situ exploration including hazard detection during EDL, rover traversing, automated rover navigation, science target selection, and sample acquisition. The protocol may be provided as a library of function calls or as a remote procedure call over a socket. For window platforms, intelligent operation agents are implemented employing Microsoft's Distributed Component Object Model (DCOM) mechanism.

Three types of operation interfaces have been implemented to support three research activities: 1) Site information server with the rover dynamics simulation team, 2) Descent Imager server with the EDL research team, 3) PanCam server with the vision-based research team. The site information server provides the real world interaction to a rover while the descent imager server and PanCam

server provide simulated data products from the respective instrument systems. Each server is implemented with a domain-intelligent-interface agent that understands the relationship among the virtual site, observation platform, and instrument system.

Future Direction

Knowledge engineering of the in-situ science domain will involve development of a hierarchical site property description taxonomy that can derive low-level properties from scientific expressions. Knowledge engineering of the instrument design process will require a flexible mapping method between the instrument design specification and the instrument models. Knowledge engineering of autonomous in-situ exploration will involve operation scenario modeling and site knowledge reconstruction algorithms.

The future work will also address more complex in-situ mission environments with multiple science stations, where each science station is equipped with a set of heterogeneous science instruments on a rover platform. The coordination among the multiple science stations needs to be pursued in two levels: an external source that defines a science goal, and the internal observation-based site reconstruction that develops a geographic knowledge set of the site utilizing the measurements from the virtual instruments. The observation planning controls the virtual instruments to obtain necessary measurements to acquire site knowledge. The ultimate purpose of these activities is to develop an on-site information infrastructure that enables collaborative operation planning among multiple rovers and multiple instruments.

References

[1] M. P. Golombek, and N. T. Bridges, "Erosion rates on Mars and implications for climate change: Constraints from the Pathfinder landing site," *J. Geophys. Res*., Vol. 105, No. E1, 1841-1853, 2000.

[2] M. P. Golombek, "Introduction to the Special Section: Results from Mars Pathfinder Part 2," *J. Geophys. Res*., Vol. 105, No. E1, 1719, 2000.

[3] M. Lee, R. Swartz, and R. Weidner, "SceneGen," *NASA Tech Brief*, 1997.

[4] L. A. Soderblom, "The Composition and Mineralogy of the Martian Surface from Spectroscopic Observations," *Mars*, University of Arizona Press, H. H. Kieffer, M. B. Jakosky, C. W. Snyder, M. S. Mathews, ed., pp. 557-593, 1992.

Update on Goldstone's Lunar Ultra-high Energy Neutrino Experiment

Peter Gorham[a], *Kurt Liewer*[b], *R. Milincic*[a], *Charles Naudet*[b], *David Saltzberg*[c], *D. Williams*[c]

Introduction

Since 1999 we have been conducting a series of experiments using the Deep Space Network (DSN) antennas at Goldstone to search for ultra-high energy neutrinos. At neutrino energies of about 10^{18} electron volts (eV), cascades in the upper 10 meters (m) of the radio-transparent lunar regolith should result in electromagnetic pulses that are detectable by large radio telescopes at earth [1,2]. Recent accelerator results [3,4] have confirmed the 1962 prediction of Askaryan [5,6] that electromagnetic cascades in dense media produce strong coherent pulses of microwave Cherenkov radiation. These confirmations strengthen the motivation to use this effect to search for cascades induced by the predicted diffuse backgrounds of high energy neutrinos, which are associated with the presence of 10^{20} eV cosmic rays in many models. One prior experiment has been reported, using the Parkes 64-m telescope [7] with about 10 hours of livetime. Since mid-2000, our project Goldstone Lunar Ultra-high energy neutrino Experiment (GLUE) has moved into a new status as an ongoing experiment, and receives more regularly scheduled observations. To date we have acquired over 80 hours of accumulated livetime. Although this total livetime is a small fraction of what would be possible with a dedicated system not subject to spacecraft telecommunications priorities, the volume of material on the Moon's surface that we are sensitive to is enormous, exceeding 100,000 kilometers cubed (km^3) at the high energies. The resulting sensitivity is enough to begin to constrain some models for diffuse neutrino backgrounds at energies near and beyond 10^{20} eV.

Description of experiment

Emission geometry and signal characteristics

The lunar regolith is an aggregate layer of fine particles and small rocks. It consists mostly of silicates and it has a typical depth range of 10 to 20 m in the maria and valleys, but may be hundreds of meters deep in portions of the highlands [8]. It has a mean dielectric constant of ~3, a density of 1.7 gm cm^{-3} and a field attenuation length at 2 gigahertz (GHz) of ~9 m [9]. In Figure 1 we illustrate the signal emission geometry. At 10^{20} eV, the interaction length L_{int} of an electron or muon neutrino for the dominant deep inelastic hadronic scattering interactions is about 60 km [10]. Upon interaction, a 10-m long cascade then forms as the secondary particles multiply. Compton scattering, positron annihilation, and other scattering processes then lead to a ~20% negative charge excess which radiates a cone of coherent Cherenkov emission at an angle of 56 degrees, with a Full Width at Half Max (FWHM) of 1 degree. The radiation is in the form of a sub-ns pulse through the regolith to the surface where it is refracted upon transmission.

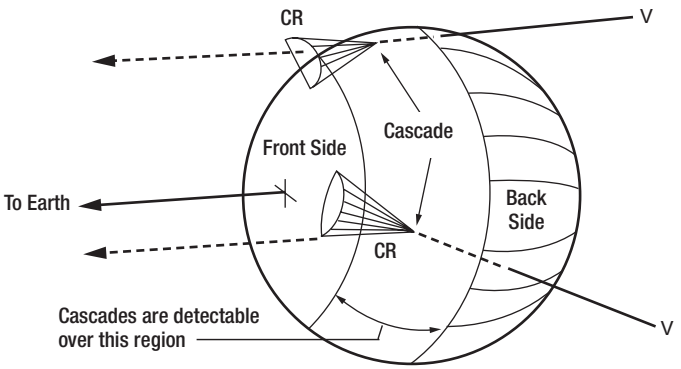


Figure 1. Schematic of the geometry for lunar neutrino event detection, where CR indicates Cherenkov radiation and n represents a neutrino.

[a] Department of Physics and Astronomy, University of Hawaii
[b] Jet Propulsion Laboratory
[c] Department of Physics and Astronomy, UCLA

This results in a larger effective area of the lunar surface over which events can be detected, as well as a greater acceptance solid angle

At the regolith surface, the resulting microwave Cherenkov radiation is refracted strongly into the forward direction. Scattering from surface irregularities and demagnification from the interface refraction gradient fills in the Cherenkov cone. This results in a larger effective area of the lunar surface over which events can be detected, as well as a greater acceptance solid angle.

The predicted radio emission [1,2] peaks in the microwave region, and is well matched to typical capabilities of the large antennas of the DSN and other large ground-based facilities. An additional advantage is that the beam sizes of the DSN antennas are of the same order of magnitude as the Moon's angular diameter in the sky. At frequencies above 2 GHz, ionospheric delay smearing is unimportant, and the signal should appear as highly linearly polarized, band-limited electromagnetic impulses [11,12,13]. However, since there are many anthropogenic sources of impulsive radio emission, the primary problem in detecting such pulses is eliminating sensitivity to such interference.

Trigger system

The experimental approach in our initial 12 hours of observations was to use a single antenna trigger with dual-antenna data recording [14]. This was accomplished by using the local S-band signals at Deep Space Station 14 (DSS 14) to form a two-fold coincidence with an active veto from the L-band interference monitor. However, the local Radio Frequency Interference (RFI) can sometimes be so intense as to cause 100% deadtime. In order to operate in the harshest RFI environment, we have now developed an approach that utilizes both the dual S-band signals on the 70-m antenna (DSS 14) and the S-band signal from the 34-m antenna at DSS 13. The signals are configured, as shown in Figure 2, in a coincidence-type system, which has been proven to be very effective in solving terrestrial interference.

The four triggering signals from the two antennas are converted to unipolar pulses using tunnel-diode square-law detectors. Stanford Research Systems SR400 discriminators are used for the initial threshold level, and these are set to maintain a roughly constant single channel rate, typically 0.5-1 kHz/channel for DSS 14 and 30 kHz/channel for DSS 13 (DSS 13's rate is higher due to a lower threshold, compensating for the reduced aperture size). A local coincidence is then formed for each antenna's signals. The DSS 14 coincidence between both left and right circular polarizations ensures that the signals are not highly circularly polarized. At DSS 13 the S-band signal is split into a high- and low-bandpass channel, each bandpass being over 60 MHz wide and separated by over

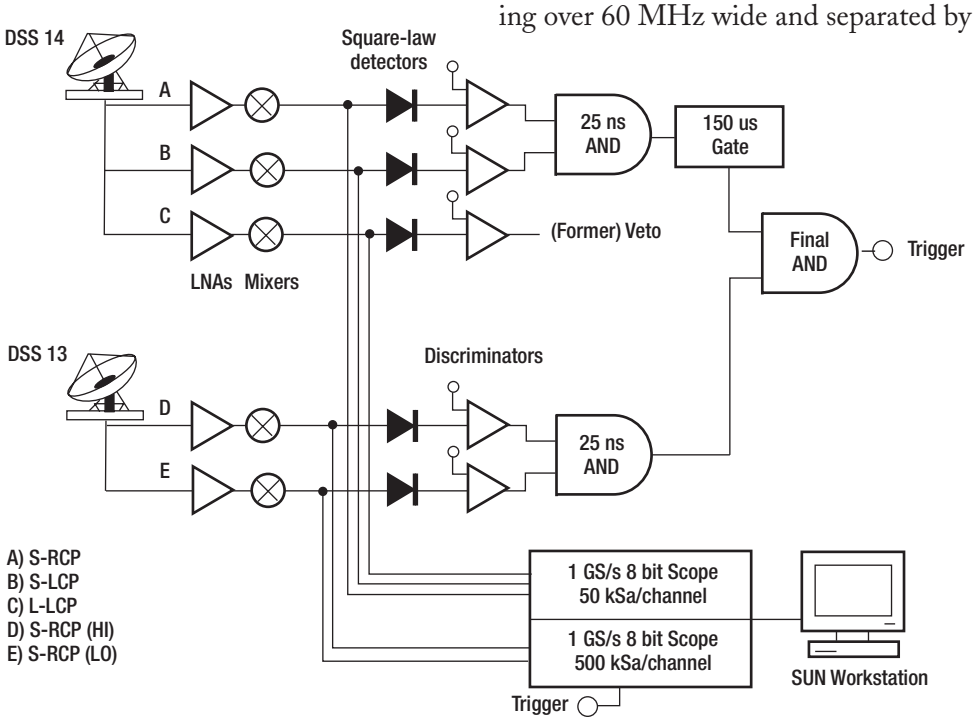


Figure 2. The GLUE trigger system used for the lunar neutrino search. The event data is stored on oscilloscopes at 1 GS/s then downloaded onto a SUN workstation.

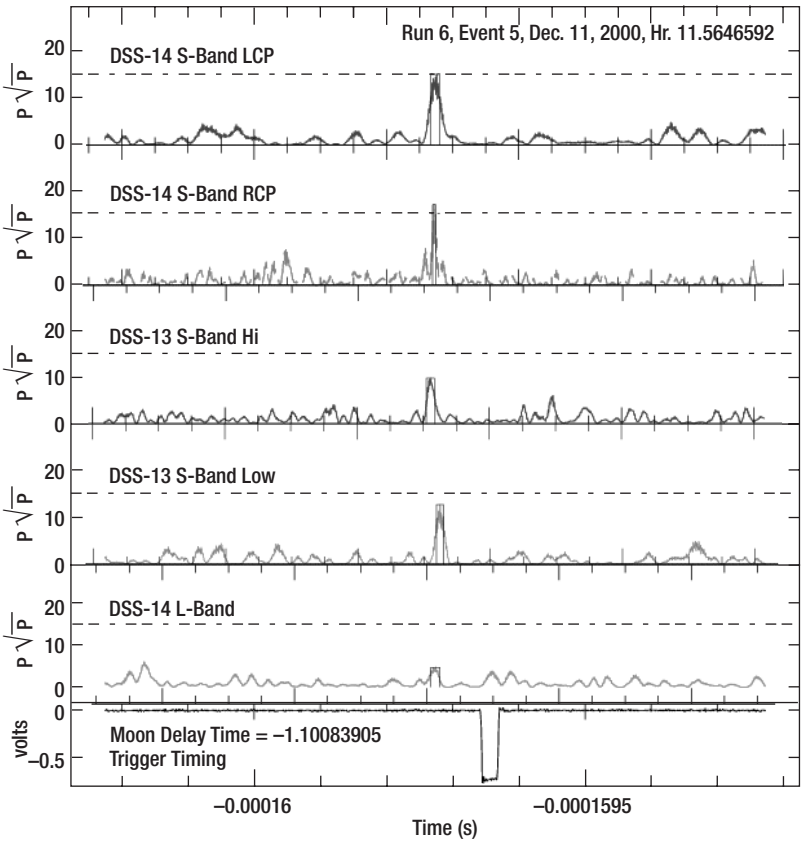
75 MHz. The split coincidence helps to ensure that the signal is broadband.

Timing, calibration, and geometric considerations

Timing and amplitude calibration are accomplished by a series of interlocking methods. Since we are critically dependent on timing to accurately estimate our delays, we internally calibrate the back-end trigger system using a synthesized Intermediate Frequency (IF) pulse signal, giving precision of order 1 ns. We then use a 2.2 GHz monocycle pulse generator projected at the antennas externally using a microwave horn to calibrate the cross-channel delays of each antenna to a precision of typically 1–2 ns. The cross-polarization timing at DSS 14 is checked with a second method which utilizes the fact that the thermal radiation from the limb of the moon is significantly linearly polarized (from differential Fresnel effects [15, 16]), and this is a positive Left-Circular Polarization (LCP) to Right-Circular Polarization (RCP) correlation that is easily detected.

Cross-antenna timing calibration is accomplished by cross-correlating a 250 ms thermal noise sample of a bright quasar, typically 3C273, recorded from both antennas at the same time and in the same polarization, using the identical data acquisition system used for the pulse detection. This procedure establishes the delay offset between the two antennas, which is of order 138 ms, but can change from run to run by tens to even hundreds of ns because of changes in fiber routing for different experiments. This measurement is accurate to typically a few ns or better.

Amplitude calibration is accomplished by the standard methods of radio astronomy, by referencing to a known thermal noise source at a known temperature. The measured system temperature during a run then fixes the value of the noise level and therefore the overall sensitivity. We also check linearity for pulsed transient response using the monocycle generators and this ensures that we have the dynamic range required to see large pulses. The Moon, with a temperature of several hundred kelvins at S-band, provides a first-order calibration of the thermal noise amplitude.



Event example

Figure 3 shows a typical event which triggered the system. The top two panes contain the DSS 14 LCP and RCP signals, and a narrow pulse is present in both polarizations, indicating a broadband spectral content and a high degree of linear polarization. The overlain rectangle on each pulse is a fit to the pulse width and is used in the visual analysis. The pulse power is normalized to the local mean power over a 250 ms window (the statistics are checked separately to ensure that they follow the expected approximately exponential distribution in intensity). The third and fourth panes show the two channels from DSS 13. The fifth pane from the top shows L-band offset feed signal from DSS 14, and no noise is present which would indicate RFI. Finally, the bottom pane shows a sum of the (negative) logic signals that formed the trigger, showing that there was good alignment. Systematic timing offsets from channel to channel are typically less than 10 ns in these data.

Figure 3. An example of a triggered event. The displayed signals are (from top to bottom): the LCP intensity, the RCP intensity, and the 34-m's high- and low-band intensity followed by the L-band off-axis interference monitor (no interference present). Finally the trigger logic summed discriminator pulses (including a 100 ns delay relative to the pulse. The box drawn around each pulse shows a fitted width indicator used for visual inspection of the events. The event file and event number are shown in the lower left hand corner.

In practice, our first pass analysis requires a pulse of equivalent 6 sigma significance. In terms of normalized power, we require a value of at least 20 times the mean power in each polarization to be considered as a candidate. The normalized power for the event in Figure 3 is seen to be to small in three of the four channels. To date we have seen no such candidates in 80 hours of livetime.

Monte Carlo results

To estimate the effective volume and acceptance solid angle as a function of incoming neutrino energy, we used the Monte Carlo technique. Events were generated at discrete neutrino energies using current best estimates of both charged and neutral current cross sections [10] and the Bjorken-y distribution. Both electron and muon neutrino interactions were included, and Landau-Pomeranchuk-Migdal effects in the shower formation were estimated [13]. At each neutrino energy, a distribution of cascade angles and depths with respect to the local surface was obtained, and a refraction propagation of the predicted Cherenkov angular distribution was made through the regolith surface, including absorption and reflection losses and a first order roughness model. Antenna thermal noise fluctuations were included in the detection process.

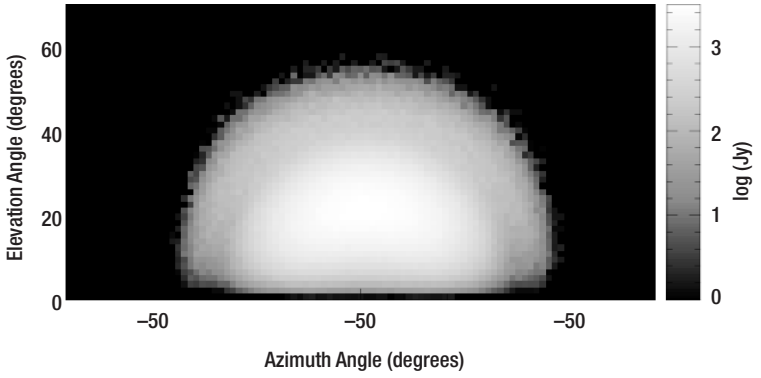


Figure 4. The microwave Cherenkov radiation pattern from an event in the lunar regolith. This plot shows a map of the average intensity projected onto the sky that would result from a typical 10^{20} eV neutrino cascade. The weaker intensity is indicated by the darker pixels.

A portion of the simulation is shown in Figure 4. Here the flux density is shown as it would appear projected onto the sky, with coordinates (0,0) corresponding to a horizontal ray with respect to the lunar surface in the direction of the original cascade (see Figure 1). The units are Janskys (Jy) as measured at earth, and the plot is an average over several hundred events at different depths and a range of angles consistent with a 10^{20} eV neutrino interaction.

Results

Our initial 90% CL limit, for 80 hours of livetime, is shown plotted (diamonds) in Figure 5. The limit is based on the observation of no events above an equivalent 6 sigma level amplitude (referenced to the 70-m antenna) consistent with the direction of the Moon. These limits assume a monoenergetic signal at each energy; thus they are differential limits and independent of source spectral model and represent the most conservative limits we can apply. Our limits just begin to constrain the highest topological defect model [20] for which we expected a total of order 1–2 events.

Additional curves in Figure 5 shows the predicted fluxes of Extremely High Energy (EHE) neutrinos from a number of models including Active Galactic Nuclei (AGN) production [17] gamma-ray bursts [18], EHE cosmic-ray interactions [19], topological defects [20,21], and the Z_0 burst scenario [22]. Also plotted are limits from about 70 days of Fly's Eye livetime [23] (accumulated in several years of runtime), which apply only to electron neutrino events.

Conclusions

We have developed a robust system for observing microwave pulses produced in the lunar regolith by electromagnetic particle cascades above 10^{19} eV. We have operated this system to achieve a livetime of 80 hours, with no large apparent signals detected to date. Due to the enormous volume of material on the Moon surface and the large gain of the DSN antennas, even with modest amounts of livetime, we have been able to set conservative upper limits on the diffuse cosmic neutrino fluxes over the energy range from $10^{19\text{-}23}$ eV. We have also begun to analyze smaller amplitude events and have some preliminary indications that a signal may be present, but this requires further study. Observation of

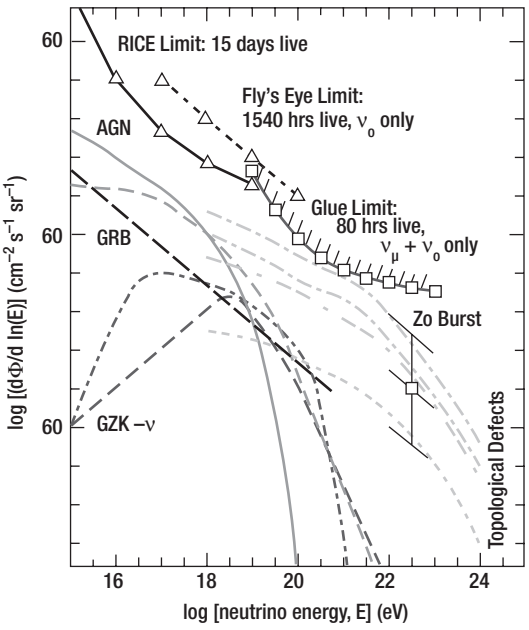


Figure 5. Plot of model neutrino fluxes and experimental limits from the Fly's Eye, Radio Ice Cherenkov Experiment (RICE), and the present work. Here "live" indicates livetime.

cosmogenic neutrinos would assist us in unlocking one of the most intriguing and intractable questions in high-energy astrophysics: What is the source of the 10^{20} ev cosmic rays?

Acknowledgments

We wish to dedicate this work to the memory of George Resch, who passed away in November 2001 and who was instrumental in making this work possible. Without his encouragement and unflagging support, this work would not have been done. In addition, we would like to thank M. Klein, T. Kuiper and the staff at Goldstone for their enthusiastic support of our efforts. This work was performed in part at the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA, and supported in part by the Caltech President's Fund, by DOE contract DE-FG03-91ER40662 at UCLA, the Sloan Foundation, and the National Science Foundation.

References

[1] I. M. Zheleznykh, 1988, Proc. Neutrino '88, 528.

[2] R. D. Dagkesamanskii, I. M. Zheleznyk, 1989, JETP 50, 233.

[3] P. W. Gorham, D. P. Saltzberg, P. Schoessow, et al., 2000, Phys. Rev. E **62,** 8590.

[4] D. Saltzberg, P. Gorham, D. Walz, et al., 2001, Phys. Rev. Lett., **86,** 2802.

[5] G. A. Askaryan, 1962, JETP 14, 441.

[6] G. A. Askaryan, 1965, JETP 21, 6587.

[7] T. H. Hankins, R. D. Ekers and J. D. O'Sullivan, 1996, MNRAS 283, 1027.

[8] D. Morrison and T. Own, 1987 The Planetary System, (Addison-Wesley: Reading, MA).

[9] G. R. Olhoeft and D. W. Strangway, 1975, Earth Plan. Sci. Lett. 24, 394.

[10] R. Gandhi, 2000, hep-ph/0011176.

[11] E. Zas, F. Halzen, and T. Stanev, 1992, Phys. Rev. D 45, 362.

[12] J. Alvarez-Muniz, and E. Zas, 1996, Proc. 25th ICRC, ed. M.S. Potgeiter et al.,7,309.

[13] J. Alvarez-Muniz, and E. Zas, 1997, Phys. Lett. B, 411, 218.

[14] P. Gorham, et al., Proc. 26th ICRC, HE 6.3.15, astro-ph/9906504.

[15] N. S. Soboleva, "Measurement of the polarization of lunar radio emission on a wavelength of 3.2 cm," Soviet astronomy - AJ, 6 (6), 873, (1963).

[16] C. E. Heiles and F. D. Drake, "The polarization and intensity of thermal radiation from a planetary surface," Icarus 2, 281 (1963).

[17] K. Mannheim, 1996, Astropart. Phys. 3, 295.

[18] E. Waxman and J. N. Bahcall, 2000, Ap.J. 541, 707, astro-ph/9902383.

[19] C.T. Hill and D.N. Schramm, 1985, Phys. Rev. D 31, 564.

[20] S. Yoshida, H. Dai, C. C. H. Jui, and P. Sommers, 1997, ApJ 479, 547.

[21] P. Bhattacharjee, C.T. Hill, and D.N. Schramm, 1992 PRL 69, 567.

[22] T. Weiler, 1999, hep-ph/9910316.

[23] R.M. Baltrusaitas, G.L. Cassiday, J.W. Elbert et al., 1985, Phys. Rev. D 31, 2192.

This newsletter is a publication of JPL's Interplanetary Network Directorate (IND). The IND Technology Program is managed by Jim Lesh and the Science Program by Michael J. Klein.

The IND Program Overview and related features are located at: http://tmot.jpl.nasa.gov. For all issues of this newsletter, click Program Overview Information and, on the resulting page, click IND News.

This publication was prepared and all work herein was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, unless otherwise indicated.

To add or delete your name from the IND News, contact Cruzita Abellana at 818-354-9071 or email Cruzita.Abellana@jpl.nasa.gov

Information Systems Manager	*Jay Wyatt*
Communication Systems Manager	*Jim Layland*
Non-Deep Space Manager	*Pat Liggett*
Science Manager	*Mike Klein*
Managing Editor	*Charles T. Stelzried*
Associate Editor	*Christopher A. Weaver, Electronic Publishing Services*
Design and Layout	*Audrey Steffan, Design Services*
Illustration and Production	*David Hinkle, Design Services*

National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL 410-065, Issue No. 16 9/02

IND Technology Program Office
Jet Propulsion Laboratory
MS 303-407
4800 Oak Grove Drive
Pasadena, California 91109-8099