Safe and Efficient Robotic Space Exploration with Tele-Supervised Autonomous Robots

Alberto Elfes  
*NASA Jet Propulsion Laboratory*

John M. Dolan  
*Carnegie Mellon University*

Gregg Podnar  
*Carnegie Mellon University*

Sandra Mau  
*Carnegie Mellon University*

Marcel Bergerman  
*Carnegie Mellon University*

Follow this and additional works at: <http://repository.cmu.edu/robotics>

Part of the Robotics Commons
Safe and Efficient Robotic Space Exploration with Tele-Supervised Autonomous Robots

Alberto Elfes*, John M. Dolan, Gregg Podnar, Sandra Mau, Marcel Bergerman

*Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA, 91109-8099
elfes@jpl.nasa.gov

Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213-3890
jmd@cs.cmu.edu

Abstract

A successful plan for space exploration requires the commissioning of fleets of robots to prospect, mine, build, inspect and maintain structures, and generally assist astronauts, rendering the overall mission as safe as reasonably achievable for human beings, the most precious resource. The authors are currently developing, under the support of NASA, a Robot Supervision Architecture (RSA) which will allow a small number of human operators to safely and efficiently telesupervise a fleet of autonomous robots. This represents a significant advance over the state of the art, where currently one robot is overseen by a group of skilled professionals. In this paper we describe some aspects of this work, including the architecture itself for coordination of human and robot work, failure and contingency management, high-fidelity telepresence, and operation under limited bandwidth. We also present highlights of our first application: wide area prospecting of minerals and water in support of sustained outposts on the Moon and on Mars.

Introduction

NASA has initiated the implementation of its Vision for Space Exploration by planning to return human beings to the Moon by 2018 and then proceed to Mars by 2030. This bold, risky, and costly enterprise will require that all possible actions be taken to maximize the astronauts’ safety and efficiency. The authors believe that this can be facilitated by fleets of robots autonomously performing a wide variety of tasks such as in-space inspection, maintenance and assembly; regional surveys, mineral prospecting and mining; habitat construction and in-situ resource utilization (ISRU); etc. These robots will be telesupervised by a small number of human ground controllers and/or astronauts, who will be able to share control with and teleoperate each individual robot whenever necessary, all from a safe, “shirtsleeve” environment.

In this paper we present the Robot Supervision Architecture (RSA), a multilayered architecture for human telesupervision of a fleet of mobile robots. This research is supported by the Advanced Space Operations Technology Program of NASA’s Exploration Systems Mission Directorate. Our objective is to demonstrate that the RSA enhances the telesupervisor’s efficiency in a real-world mineral prospecting task (see Figure 1) while allowing supervision of the robot fleet from the relative safety of a lander, orbiter or ground station. The architecture is general enough to accommodate a wide range of applications and to span the entire spectrum of so-called sliding autonomy levels, from “pure” teleoperation to supervised control to “fully” autonomous [6].

Figure 1. Artist’s conception of telesupervised wide-area robotic prospecting on the Moon.

Our Robot Supervision Architecture addresses the following important challenges:

1. Coordination of human and robot work: The RSA is designed both to minimize the need for humans to perform costly and risky extra-vehicular activities (EVA), in which astronauts in space suits execute tasks in orbit or on lunar/planetary surfaces; and to multiply their efforts beyond direct constant teleoperation of one robot. It does so by allowing the robots to operate as autonomously as technically possible, receiving assistance whenever necessary; and by allowing the human to assume partial or total control of a robot or its task-specific instrumentation whenever appropriate. While we do not explicitly address issues of tightly-coupled coordination between collocated humans and robots (i.e., joint task performance), the architecture supports augmenting the robots in these scenarios when autonomy is insufficient and continued assistance is critical.

1 This work is supported by NASA under Cooperative Agreement No. NNA05CP96A.
2. Enhanced human safety and efficiency: Allowing a human to telesupervise a fleet of autonomous robots from the safety of a “shirtsleeve” environment is a prudent and efficient approach to future space exploration. As our research progresses, we will test the efficiency gains over the baseline of a single human in a space suit performing EVA by tasking a fleet of robots to execute wide-area prospecting; and we will compute system performance metrics that take into account the area covered, prospecting accuracy, terrain difficulty, human safety, number of robotic rovers, human effort defined as the degree of human task intervention in full-time equivalents, and task completion time.

3. Failure and contingency management: The architecture explicitly defines a Hazard and Assistance Detection subsystem which operates on multiple levels. At the robot level, it assesses deviations from standard or expected operating conditions, both with respect to the robot’s health and its assigned task. At the workstation level, the Hazard and Assistance Detection subsystem is responsible for queuing and prioritizing all robot alerts and requests, based on a criterion that takes into account the type of hazard, estimated time for the telesupervisor to fix it, and the predicted time to reach critical danger.

4. Remote operations with bandwidth constraints: Our system relies on both high-bandwidth radio links for geometrically-correct, high-fidelity telepresence and teleoperation of any robot in the fleet, and lower-bandwidth links for command and telemetry exchange between the robots and the telesupervisor workstation. Should the high-bandwidth video link malfunction, the system design provides for graceful fallback to lower-bandwidth communications.

It is important to note that our approach is optimal when the human telesupervisor and the robot fleets are “near” each other, meaning that they are separated by a roundtrip communication delay of no more than about 300 milliseconds (28,000 mi / 45,000 km distant). It is also applicable with short telecommunication delays, such as between the Earth and Moon.

This paper is structured as follows. In the next section we discuss the novelty of our work with respect to the state of the art. In the following section we describe the architecture itself, as an overarching paradigm under which all other subsystems reside. It addresses specifically challenges #1 and #2 above. In the sequel we address challenge #3, presenting the Hazard and Assistance Detection subsystem and its underlying protocols. In the next-to-last section we present details of our first application area, the wide-area mineral prospecting task, including a task-specific performance metric. The last section presents conclusions and future work. For completeness, we note that our approach to challenge #4 is presented in more detail in another paper [10].

Related Work

The most important aspect of our work, namely, creating a Robot Supervision Architecture that allows a human safely and efficiently to telesupervise a fleet of autonomous robots, encompasses a variety of robotic technologies. We review here only research focused on human-robot interaction for space exploration, or which is strongly applicable to the area.

The state of the art in robotic space exploration is the Mars Exploration Rover (MER) mission [8]. Spirit and Opportunity combined have logged over 10 km and operated for over 1200 sols (Martian days). This is achieved by assigning a large team of highly skilled professionals to download telemetry and imagery, interpret the data, plan the rover activities, and program and upload commands every sol, in addition to a large science team to select science targets and tasks. In contrast, we are multiplying one human’s capability to telesupervise a large number of robots, while still allowing the human to perform other tasks. Another difference between MER and this work is that, because of the long communication delays between Earth and Mars, the only possible way of operating the rovers is via batch command sequences which are executed in autonomous mode, whereas the RSA accommodates a large variety of operation modes.

The sliding autonomy aspect of space exploration is one of great importance. Heger et al. [6], in particular, have developed an architecture geared towards humans and robots “jointly performing tightly coordinated tasks.” They focus on “how small teams of robots in space and a few humans on Earth could work together to assemble large orbital structures,” while we focus on maximizing an astronaut’s efficiency by coordinating a large fleet of robots.

From the point of view of direct assistance to and collaboration with astronauts, a relevant project is Robonaut [1]. Robonaut’s focus is a space robot “designed to approach the dexterity of a space suited astronaut.” From the point of view of RSA, a Robonaut would be another robotic device whose operation could be coordinated using our architecture. When teleoperated, Robonaut’s main similarity with our work is the telepresence capability implemented with stereo cameras. However, Robonaut’s use of a head-mounted display and converged cameras differs from our geometrically-correct remote viewing system.

Other human-multiprobot architectures are those of Nourbakhsh et al. [9] and Sierhuis et al. [12]. The former focuses on urban search and rescue operations; their architecture allows for simultaneous operation of real-world and simulated entities. The latter have created a Mobile Agents Architecture (MAA) integrating diverse mobile entities in a wide-area wireless system for lunar and planetary surface operations. Our work is conceptually
similar to these, but it differs in that we focus on human safety and efficiency in a planetary exploration environment by providing high-fidelity telepresence and a hazard and assistance detection methodology that seeks to optimize the use of human attention resources given a set of robot assistance requests.

Finally, we note the work of Fong et al. [5], where the authors also develop an architecture for supervision of multiple mobile robots. Their work and ours differ in the assistance request protocols and our use of stereoscopic telepresence.

**The Robot Supervision Architecture**

The RSA is implemented as a multilayered multi-robot control and coordination architecture that can accommodate different configurations of robotic assets based on previous work by Elfes [3], [4]. Here, “multilayered” means that robot system control is performed at multiple levels of resolution and abstraction and at varying control rates. Likewise, “replicated” means that the fundamental activities of perception, decision-making and actuation occur at each layer of the architecture. A diagram of the overall RSA architecture is shown in Figure 2 and explained below.

The Autonomous Navigation System (ANS) is replicated on each robot for local rover navigation. In the same way, each robot's Hazard and Assistance Detection (HAD) system is tightly coupled with the local ANS, and is supported up through the layers of the RSA architecture for high-level decision-making and handover to the telesupervisor.

The Human Telesupervisor oversees the entire mission planning and execution, being able to assume a wide range of roles – from “pure” supervision while monitoring the progress of the assigned tasks; to monitoring the performance of the fleet of autonomous robots; to “pure” teleoperation of any robot vehicle or its subsystems. This means that the RSA covers the entire sliding autonomy spectrum as defined in [6]. The reader should note that this is not to be confused with the so-called levels of interaction engagement [11], as we are not dealing with the issue of robots interacting with humans in a “social” way.

**Task Planning and Monitoring and Robot Fleet Coordination** lie at the core of the Robot Supervision Architecture. The high-level mission plans are created and edited with the Planning Tools, and are then assigned to the Robot Fleet Coordination module, which decomposes them into tasks and assigns these to the individual robot controllers (see Figure 3).

![Figure 2. RSA system-level block diagram and main data paths.](image_url)
Robot Fleet Coordination also collects operational results from all robots and integrates them for convenient human monitoring and analysis. The Monitoring Level also includes Perception — Decision-Making — Actuation sequences to monitor multi-robot operations at the system-level, and to analyze for high-level hazard and assistance detection. Robot Fleet Coordination imagery and telemetry are combined for building regional imagery and maps, and are also presented graphically to the telesupervisor, as shown in Figure 4.

As depicted in Figure 5, each individual Robot Controller subsystem is responsible for receiving a collection of tasks from the Robot Fleet Coordination and monitoring its execution. The robot controller has direct access to all of the robot’s subsystems to drive actuators and read sensor data. When a robot is a relatively complex combination of mobility, manipulation, and other engineering or science subsystems, the corresponding robot controller may be implemented as a collection of modules responsible for each one of them. Each robot controller is currently subdivided into the Autonomous Navigation System and Prospecting Task Support modules, respectively responsible for controlling the mobility and prospecting subsystems. The ANS is responsible for decomposing the navigation path assigned to it and reporting its progress. The initial Prospecting Task Support software at the robot level is very simple: it merely supports the control and monitoring of the prospecting tools.

Robot Telemonitoring allows each robot to be constantly monitored at low bandwidth by the human telesupervisor with imagery and data updated regularly from each prospecting robot vehicle. As shown on the right in Figure 4, each robot has a “dashboard” which includes streaming images from one of the robot’s cameras, and graphical depictions of status data such as battery charge, attitude, motor temperatures, and any other monitored telemetry. The Hazard and Assistance Detection (HAD) system (see next section) automatically monitors each robot. When the telesupervisor should be made aware of a hazardous condition, it is on that robot’s dashboard that it is indicated. This is exemplified for Robot #2 in Figure 4 by an orange surround of the robot view.
Telepresence and Teleoperation subsystems: We make a distinction between monitoring the operation of each robot, and telepresently taking control of a robot. Whereas monitoring is supported by simultaneous low-bandwidth data streams from each of the robots, telepresence is supported by high-bandwidth stereoscopic imagery and other telesensory modalities one robot at a time. It provides not only the stereoscopic visual, but also aural, and attitude proprioceptive feedback that allows for more immersive telepresence.

Teleoperation involves direct human control of a single robot when a vehicle must be remotely driven rather than operating under its Autonomous Navigation System; and when the task-specific tools must be operated manually. Joystick, keyboard, and task-specific input devices support this. This subsystem is implemented over a dual-path data communication infrastructure, where the low-bandwidth path is used for communication of commands and telemetry data, and the high-bandwidth path is used for stereoscopic video. In addition to the described functionality, each subsystem is a source for data which are both archived for later analysis, and also provided in part as a stream for access by a Distant Expert who can consult as required.

Hazard and Assistance Detection

The Hazard and Assistance Detection (HAD) subsystem is responsible for the following high-level capabilities:
- Single-rover hazard and assistance assessment.
- Multi-rover assistance request prioritization and management.

Its overall goal is to assess situations where a robot requires assistance, alert the operator about them, and optimally prioritize tasks for the operator such that the overall pause-time of rovers in the team is minimized without endangering them and, consequently, the mission. “Pause-time” is the duration of time when the rover pauses while waiting for telesupervisor attention. Task-specific assistance requests and automatically-identified science opportunities both benefit from, and are handled within the hazard assistance and scheduling tools; terminology which is used throughout our current discussion.

The detection of hazards involves a fusion of various sensor inputs to generate a comprehensive picture of the situation. Both the lower-level HAD located on each rover and the higher-level HAD located in the telesupervisor workstation are implemented as a perception – decision-making – actuation sequence.

At the lower level, the perception aspect includes receiving inputs from the various sensors and subsystems, including such data as motor current, battery levels, and odometry. The decision-making aspect involves assessing the whole spectrum of available sensor data and determining whether there is a potential for hazard. If such a hazard is identified, the urgency of the situation is evaluated by examining which hazard state it is in (normal green, warning yellow, or high-alert red) and by calculating a prediction of how quickly it will degrade (maximum allowable neglect time) and the estimated time to fix the problem (fix-time). The actuation aspect involves responding to the detection of a red alert hazard by halting the actuators on the robot and alerting the supervisor with a hazard flag and the time-based estimates mentioned above. If the flag is merely a yellow warning and not a high-alert red, then the supervisor is cautioned about the situation, but the operation of the rover remains uninterrupted.

At the higher level on the supervisor side, HAD perception receives cautionary (yellow) and high-alert (red) hazard flags, and their related telemetry from the rovers. The decision-making aspect prioritizes the hazard flags in the Assistance Queue such that total pause-time for the rover team is minimized while not endangering any rover. The resulting actions (actuation) are to inform the telesupervisor of the hazard or potential hazard through the supervisor workstation control panel.

Since a telesupervisor oversees multiple rovers, it is not feasible for the operator to be fully aware of each rover’s situation at all times. When a hazard is flagged, the operator must be made aware of exactly what hazard was detected and why it occurred. To get a better picture of the situation, data from various sensors and the video feed are monitored by the HAD algorithm. The HAD subsystem was designed as in Figure 6 to address these requirements. Each subsection of the diagram is explained in a top-down fashion in the following sections.

Robot Fleet Coordination Level: Prioritized Assistance Queue

The major motivation for developing the HAD algorithm is to be able to identify, alert and prioritize requests for the telesupervisor’s attention. It also optimizes and recommends a particular order to the operator for fixing multiple rovers efficiently. The goal that our algorithm focuses on is to minimize overall robot pause-time – to get the least number and shortest overall duration of inoperative rovers – without neglecting any long enough to endanger them. These alerts are sorted and prominently listed for the telesupervisor in the Assistance Queue on the workstation.

The following simplifying assumptions were adopted in the prioritization algorithm: (i) robots operate independently of each other, (ii) context switching from one robot being serviced to another is instantaneous, and (iii) a rover is considered “rescued” as long as its associated parameters are back into the green state. The authors are aware that real-world situations are more complex and demand a more sophisticated implementation; for the time being, however, these assumptions are not overly restrictive, and they will be addressed in future work.
**Prioritization – Minimize Pause-Time & Harm**

Crandall and Goodrich [2] proposed the idea of a cycle of robot effectiveness where effectiveness is modeled to degrade during autonomous operation and restored to original values through human teleoperation. The time periods during which the rover acts autonomously on the one hand and under human intervention on the other are respectively dubbed Neglect-Time and Interaction-Time. We adopt a similar concept such that when a hazard flag occurs, Neglect-Time is the amount of time the operator does not address the issue, and Fix-Time (similar to Interaction-Time) is the amount of time it takes the operator to resolve the hazard situation. We assume that during the repair the rover is not accomplishing its prescribed goal, thus while not “down” it is “paused” for the duration of the Fix-Time.

An example of the most basic case is given to demonstrate the derivation of the prioritization scheme. Imagine two rovers with hazards flagged simultaneously and constant estimated Fix-Times. Rover 1 requires a Fix-Time of 10 units of time and Rover 2 requires 5 units. Assuming that the performance of Robot 1 is independent of that of Robot 2, prioritizing robots by “shortest Fix-Time first” minimizes pause-time (see Figure 7, where the notation “R1: 10” indicates “it takes 10 units of time to fix robot 1” and “R2 waits: 10” indicates “robot 2 waits for attention for 10 units of time”).

When we incorporate the idea that flags may arrive at different times, then the time delay between requests, $\Delta t$, has to be taken into account. Since one arrives before the other, the problem is no longer which robot to address first, but whether it is more efficient to switch tasks in the midst of servicing the first robot. As a simplification, we neglect the possible time-cost in switching tasks midway. As illustrated in Figure 8, if the Fix-Time of the new hazard from Robot 2 ($\Delta t_{R2}$) is shorter than the remaining Fix-Time of Robot 1 ($\Delta t_{R1} - \Delta t$), then switching would be more efficient (see Figure 8a, where $R_1$: 10, $R_2$: 3, $\Delta t$: 5). If, on the other hand, the Fix-Time of Robot 2 is longer than the remaining Fix-Time of Robot 1, switching is less efficient, as in Figure 8b, where $R_1$: 10, $R_2$: 5, $\Delta t$: 7. The criterion used for the switching decision is thus $\Delta t_{R2} < \Delta t_{R1} - \Delta t$.

**Figure 7. Simultaneous arrival of assistance requests.**

**Figure 8. Asynchronous arrival of assistance requests.**
The above examples assume that the severity and therefore the required Fix-Time of unaddressed hazards do not increase with time. In fact, hazards may worsen during periods of neglect. We borrow the term “Neglect-Time” (wait time in Fig. 7 and 8) from [2], but note that it refers there to the entire time between periods of human intervention, whereas our Neglect-Time denotes only the time between the flagging of a hazard and the point of human intervention. The total time from flagging a hazard to restoring it to the safe, green zone can then be expressed as the sum of Neglect and Fix-Time (\(\Delta t_{\text{neglect}} + \Delta t_{\text{fix}}\)). This is illustrated in Figure 9, where \(\Phi\) is a generic hazard parameter.

![Figure 9. Neglect and Fix-Time.](image)

With this additional complexity of Neglect-Time added, the criterion for deciding to switch tasks for maximum efficiency changes to

\[\Delta t_{R2,\text{neglect}} + \Delta t_{R2,\text{fix}} \leq \Delta t_{R1,\text{neglect}} + \Delta t_{R1,\text{fix}} - \Delta t \quad [1]\]

Figure 9 also shows the trapezoidal model we adopted to estimate Neglect and Fix Times linearly. The justification for this model is given in a subsequent section.

In the general case, the hazard severity increases during neglect, as reflected in Figure 9 by \(\Phi\) moving closer to the maximum allowable value.. Given the rate of change of \(\Phi\) we can make an estimate of the “maximum allowable neglect time remaining,” \(\Delta \Phi_{\text{max, neglect}}\), before \(\Phi\) hits this ceiling as seen in Figure 11 (\(\Delta \Phi_{\text{max, yellow}} = \Phi_{\text{yellow}} - \Phi\)) and \(\Delta \Phi_{\text{max, red}} = (\Phi_{\text{red}} - \Phi)\). Another important consideration for scheduling is therefore that this deadline for fixing be met. This means that on top of evaluating the switching criteria for maximum efficiency, there needs to be an evaluation for feasibility of switching without endangering any robots. Thus these two criteria emerge:

\[\Delta t_{R1,\text{max, neglect}} > \Delta t_{R2,\text{neglect}} + \Delta t_{R2,\text{fix}} + \Delta t \quad [2a]\]

\[\Delta t_{R2,\text{max, neglect}} > \Delta t_{R1,\text{neglect}} + \Delta t_{R1,\text{fix}} - \Delta t \quad [2b]\]

There are four possible scenarios given these two equations as seen in Figure 10.

The prioritization scheme for each of these 4 cases becomes:

Case 1: Switch from R1 to R2 if Equation [1] is true.
Case 2: Stay with R1.
Case 3: Switch to R2.
Case 4: Switch from R1 to R2 if

\[\Delta t_{R2,\text{max, neglect}} < \Delta t_{R1,\text{max, neglect}}\]

Case 1 is based purely on minimizing pause-time. Cases 2 and 3 are invoked when one of the rovers would reach its critical limit if pause-time is minimized. The reasoning for Case 4 is to notify the telesupervisor of which problem is most likely to reach critical limits first. The human can best judge which situation is more easily fixable despite potentially reaching those critical limits or whether it would be hopeless to address one or the other.

![Figure 10. Maximum allowable neglect-time cases.](image)
The HAD scheme for prioritizing flags from multiple rovers is thus to place the first flag that arrives into the first position in the Assistance Queue and switch the order with a subsequent incoming flag only if it meets the criteria based on Equations [1] and [2] and the cases outlined. In essence, Equation [1] minimizes pause-time while Equation [2] prevents further harm.

In a dynamic situation, the Fix-Time as well as a maximum allowable Neglect-Time can be estimated based on current values and known models for resolving a hazard. This is done on the rover side and communicated to the operator side for the prioritization scheme given above.

Sorting - Urgency Levels
The HAD prioritization scheme is further categorized into three levels of urgency: normal green, warning yellow and alert red. The regions are separated by threshold values for the parameters being monitored, as shown in Figure 11. These thresholds are based on mission and robot design.

In green, the rover operates autonomously and reports telemetry at regular intervals to the telesupervisor workstation. In yellow, the telesupervisor is warned of a potential upcoming hazardous situation. Autonomous operations continue but telemetry is reported more frequently and troubled sensors highlighted. In red, everything is the same as yellow except that the rover halts itself immediately and waits for telesupervisor attention. The red and yellow flags are prioritized by the above scheme separately. Then the red flags are listed first in the Assistance Queue followed by the yellow.

The reason for having a yellow warning region is to have a buffer and additional warning for the operator. It is always better to be more cautious especially when it comes to the expensive endeavor of planetary exploration.

Each event receives a time stamp. The relevant telemetry from vision and sensors are saved, as well as a record of the actions taken. The operator is asked to briefly state what and why s/he is taking such an action either through text or voice recording.

Robot Controller Level: Fix-Time for Hazards
As introduced above, the estimated amount of time necessary for a human to remedy a hazard is known as “Fix-Time.” The prioritization description above implies that it is more efficient in terms of total pause-time to tackle a hazard with shorter Fix-Time before turning to rovers with longer Fix-Times.

As shown in Figure 9 and Figure 11, we propose a trapezoidal model as an approximation for Neglect- and Fix-Time. This allows for a linear simplification of the curve. Fix-Time consists of two linear portions, \( \Delta_{\text{fix}} = \Delta_{\text{base}} + H \Delta_{\text{neglect}} \) where the flat portion of Figure 9 is the first term (\( \Delta_{\text{base}} = t_{c - t_{0}} \)) and the sloped portion is the second term (\( H \Delta_{\text{neglect}} = t_{e - t_{c}} \)), with the coefficient \( H \) being the ratio of the rates of degradation and repair (\( \Phi_{\text{neglect}} / \Phi_{\text{fix}} \)). The rate of repair \( \Phi_{\text{fix}} \) is based on specific models for the HAD situation; for example, for the case of a low battery, \( \Phi_{\text{fix}} \) would be the rate of recharge.

The rate of degradation \( \Phi_{\text{neglect}} \) on the other hand is dependent on the current situation; for example, the case where a battery is discharging normally would be different from a case where there is a short circuit discharging it.

\( \Delta_{\text{base}} \) describes the inevitable turnaround time it will take once the human teleoperator starts to address the problem. Also, in the most basic case where HAD monitoring is binary, for example the camera is on or off, the sloped term would obviously be zero. Conversely, for cases where the turnaround is nearly instantaneous, then \( \Delta_{\text{base}} \) may equal zero instead. The sloped portion \( H \Delta_{\text{neglect}} \) described above gives an estimate based on current parameters of approximately how long the repair will take to bring \( \Phi \) back into green based on the situation (\( \Phi_{\text{neglect}} \)) and the hazard type (\( \Phi_{\text{fix}} \)).

Robot Controller Level: Neglect-Time
The Neglect-Time consists of the first sloped portion in the trapezoidal approximation of the curves in Figure 9 and Figure 11. This value is derived simply by tracking the amount of time elapsed between when \( \Phi \) transitioned from yellow to green and when the telesupervisor begins to work on the rover.

One reason for keeping track of Neglect-Time is for determining approximately what the Fix-Time should be. Another critical reason is to ensure that the task is addressed before a certain deadline as described in the next section.

\[ \Phi_{\text{max}} \]

\[ \Phi_{\text{red\_threshold}} \]

\[ \Phi_{\text{yellow\_threshold}} \]

Figure 11. Monitor and flag.

Robot Fleet Coordination Level: Hazard Logs from HAD Telemetry on the Robot Controller Level
Scientists and engineers may want later to study the detected hazards, so all relevant data in the hazard situation must be recorded and saved. This log includes a sequence of events starting when the HAD flag was raised and lasting until the operator hands back control to the rover.
Robot Controller Level: Maximum Allowable Neglect Time Remaining

There has been much work on prediction for navigation, in particular, for obstacle avoidance during navigation. However, there is much less discussion of predictive algorithms for other aspects of rover health and performance. We propose HAD as a general, all-encompassing predictive and optimizing algorithm for monitoring and handling any expected, or unexpected, hazards.

For each possible HAD case evaluated, each will monitor at least one sensor value (such as pitch angle) or calculated value (such as odometry), as well as their respective rates of change. Let $\Phi$ denote any variable being monitored. When a rover is traversing, monitoring just the value $\Phi$ itself will only inform the rover of a hazard after the fact (e.g., rover already stuck). However, if we also monitor the rate of change of this value, we can catch any interesting trends of decline/incline and get a time estimate of when $\Phi$ will reach some threshold value.

Two parameters define the yellow hazard zone: $\Phi_{\text{yellow\_threshold}}$ and $\Phi_{\text{red\_threshold}}$. The red alert zone is similarly defined by $\Phi_{\text{red\_threshold}}$ and $\Phi_{\text{max}}$. Thus the prediction equations become for yellow,

$$\Delta t_{\text{yellow\_max\_neglect}} = \frac{\Phi_{\text{red\_threshold}} - \Phi_{\text{cur}}}{\Phi_{\text{cur}}}$$

[3a]

and for red,

$$\Delta t_{\text{red\_max\_neglect}} = \frac{\Phi_{\text{max}} - \Phi_{\text{cur}}}{\Phi_{\text{cur}}}$$

[3b]

What we get is a predicted time, based on current trends, before the rover’s $\Phi$ will reach a dangerous threshold. This advance warning will give the operator time to potentially address the issue before it degrades further and it will also influence how HAD flags are prioritized in the Assistance Queue on the operator side.

This estimate of “maximum allowable neglect time” remaining before hitting a threshold (Equations [3a] and [3b]) provide a deadline by which the operator must address the issue. It does this through Equations [2a] and [2b] to evaluate the feasibility for switching HAD tasks on the operator side during the prioritization process.

Wide-Area Prospecting

Our first-year test for the RSA is to autonomously search an area for in situ resources with assistance from a human telesupervisor when needed. A set of onboard instruments for each rover represents an analysis system that will function as a stand-in for a suite of instruments for the Moon. It is not the purpose of this project to design a complete, integrated, chemical analysis system, but rather to demonstrate the interactions between human telesupervisors and prospecting robots, and validate their performance in identifying resources in the field. In the future, the prospecting instruments will be expanded to include sampling tools.

A chosen area will be prospected using a predetermined search algorithm defined prior to the test (see Figure 12). This simple grid-search algorithm is akin to any initial terrestrial prospecting task where no a priori mineral information is known about the area. It is also analogous to sub-surface sample prospecting, such as core boring, which is one of our target prospecting tasks in the future.

![Figure 12. Robot paths within prospecting area.](image)

To provide a tractable prospecting task for development and validation, we have elected to do a surface study with limited manipulation and non-contact sensor-based sampling. Because in the future the prospecting task will involve discrete prospect sites with physical soil and/or core samples collected and returned to base for analysis, the initial study is designed to be an analogue, with surface sensor data being taken in a similar fashion at discrete prospect sites. This surface sensor will provide data that can be analyzed without the need to deploy expensive sampling tools (e.g., a mass spectrometer) on each of the robots. This will be accomplished by adapting a video camera to collect visual sample data for an area of approximately 0.25 m$^2$ at each sampling site.

The target material density measurements collected at each sampling site together with the coordinates of each site will be used to build a map of material density over the prospecting area. An algorithm that infers a distribution over the whole prospecting area will be employed. This map will then be compared to the resource map maintained by our geologist who seeded the prospecting area to characterize the accuracy of the prospecting aspect of the system.

To quantitatively assess how well the system performs in the prospecting task, we propose a basic performance metric based on the following notions: 1) greater area, accuracy, terrain difficulty, and safety of prospecting coverage mean increased performance; 2) greater effort and time required mean decreased performance. Given these factors, we propose the following metric:

$$P = \frac{ACTS}{(R / w + H_E) t}$$
where:

\( P \): performance in units of \((\text{area accurately prospected}/\text{effort-time})\).

\( A \): area covered.

\( C \): prospecting accuracy; \( C = 1 \) corresponds to the highest possible accuracy and \( C = 0 \) corresponds to the lowest possible accuracy.

\( T \): terrain difficulty factor \((T \geq 1)\) with \( T = 1 \) corresponding to the easiest terrain (a flat surface without obstacles).

\( S \): safety factor \((S \geq 1)\) with \( S = 1 \) corresponding to the least safe task performance, i.e., via 100% EVA.

\( R \): number of robotic rovers (integer).

\( H \): number of humans (integer, but does not occur in the performance formula); note that although our project’s focus is on a system in which a single human astronaut controls multiple rovers, the metric is general enough to allow for multiple humans.

\( H_E \) = human effort defined as the degree of human task intervention in full-time equivalents \((0 \leq H_E \leq H)\); e.g., if one human intervenes 30 min. during a 1-hr. task, \( H_E = (30/60) = 0.5 \); if three humans intervene 15, 30, and 45 min. respectively during a 1-hr. task, \( H_E = (15/60) + (30/60) + (45/60) = 1.5 \).

\( w \): factor allowing commensurability of human and rover time by giving the relative value of the former to the latter; e.g., \( w = 4 \) sets human time to be four times as valuable as rover time.

\( R/w + H_E \): combined human-rover effort.

\( t \): time required to cover a.

We will report on the results obtained with this metric after we conclude our indoor and outdoor tests in the Fall of 2005.

**Conclusion**

The work presented in this paper summarizes parts of a larger technology development effort being undertaken by the authors under NASA support and in cooperation with NASA centers. Other aspects of this effort include the Autonomous Navigation System, based currently on standard binocular vision [13]; the Telepresence and Teleoperation System [10]; and other task-specific elements. Our ultimate goal is to deliver the entire Robot Supervision Architecture to NASA at technology readiness level 6 [7], after extensive field tests where one human will telesupervise a fleet of eight to ten autonomous robots performing mineral prospecting and core sample drilling. Specifically with respect to HAD, in the near future more sophisticated assessment protocols will be implemented, possibly using a Bayesian framework for dynamic state estimation [13]. This will improve the ability to identify obstacles and will also aid in the performance of opportunistic science in that features of interest can be detected with greater accuracy and frequency.

**References**


