# Measuring Performance in Real Time during Remote Human-Robot Operations with Adjustable Autonomy

**Debra Schreckenghost and Tod Milam,** *TRACLabs*

**Terrence Fong,** *NASA Ames Research Center*

*Simulated operations during a recent NASA robotic field test demonstrated real-time computation of performance metrics for human-robot interaction that includes adjustable autonomy.*

Future NASA missions will use interactive robots for space exploration, including scientific discovery, site survey, and mission reconnaissance. Effective use of robots for these applications requires new types of remote operations.[1] Earth-based operators will remotely supervise multiple robots performing planned tasks, independently and jointly. When needed, these operators will perform manual tasks that aid and complement the robot's tasks. Ground-control teams (including scientists and engineers) will monitor the results of these tasks to adjust robot plans. To ensure that robots are used effectively for such missions, it is important to continuously assess the performance of the human-robot team during operations. An important aspect of such operations is allocating tasks between the robots and their human operators.

The ability to flexibly allocate tasks among the human-robot team is called *adjustable autonomy*, which has its roots in Thomas Sheridan's work on functional allocation between humans and machines.[2] Adjustable autonomy has been used for various human-robot operations, including physical-structure assembly,[3] object transportation,[4] and rescue operations support.[5] It also has been applied to process control[6] and spacecraft system control.[7] In addition, protocols and procedures can be flexibly adapted to mission circumstances using adjustable autonomy. The levels of autonomy can be used to gradually automate systems typically under manual control, such as spacecraft systems.[8] For the robotic field test we describe here, rover operations were intended to be mostly autonomous. We used adjustable autonomy to bring humans into these operations in a



1541-1672/10/$26.00 © 2010 IEEE
*Published by the IEEE Computer Society*

**Figure 1. The K10 robot. The images on the left and center show the K10 robot's instruments for geology reconnaissance, and the right shows the K10 robot operating at Black Point Lava Flow, Arizona.**

planned but flexible way by adapting autonomous protocols to include manual operations that improve performance.

Performance metrics for adjustable-autonomy operations have typically measured the degree of robot independence from human intervention, such as neglect tolerance[9] and interaction efficiency.[10] For such operations models, less human interaction is preferred. For NASA robotic operations, however, minimizing human interaction might not translate to better human-robot performance. For example, it might be more time and resource efficient to teleoperate the robot in difficult terrain on a planetary surface than to let the robot operate autonomously. Thus, metrics are needed to evaluate the team productivity and success of operations that allocate tasks among both humans and robots. This article defines performance metrics for such adjustable-autonomy operations and presents the results of applying these metrics during a recent NASA lunar mission simulation.

## Robotic Reconnaissance Operations

NASA is conducting analog field tests to investigate operations concepts with advanced robots and simulated flight operations. One such investigation is the use of advanced robots for *robotic reconnaissance*, which involves operating a planetary rover under remote control to scout planned sorties prior to astronaut extra-vehicular activity (EVA). Scouting is an essential phase of field work, particularly for geology. Robot instruments provide surface and subsurface measurements at resolutions and from viewpoints not achievable from orbit. This surface-level data can then be used to select locations for field work and prioritize targets to improve astronaut productivity. Robotic reconnaissance can be done months in advance or be part of a continuing planning process during human missions.

Since 2008, we have been developing and evaluating systems, operational concepts, and protocols for robotic reconnaissance.[11] To study how robotic reconnaissance can benefit human exploration, we recently conducted a field test using the NASA Ames K10 robot (see Figure 1) at Black Point Lava Flow (BPLF), Arizona. During this field test, ground control was conducted at the NASA Ames Research Center concurrent with the robot surface operations at BPLF. The science team built robot plans to visit features of interest and take instrument readings to determine which features would benefit most from astronaut EVA. During robotic reconnaissance operations, the ground-control team reviewed robot plans, then uplinked them to the robot. These plans included tasks for the robot to perform autonomously as well as tasks to be performed manually by flight controllers remote from the robot.

Similar to planetary rovers such as JPL's Mars Exploration Rover (MER), the K10 robot immediately began to execute the uplinked plan. But unlike MER, ground control supervised the robot as it executed the plan and performed planned manual tasks. They continuously monitored robot data in real time and could intervene using teleoperations when needed.

Nominal operation of the robot during reconnaissance activities was accomplished using the plan built by the science team. The K10 robot at BPLF was equipped with three types of reconnaissance instruments:

- a downward-pointing microscopic imager for surface grain size analysis,
- a panoramic imager using GigaPan technology to provide high-resolution panoramic views of the site, and
- an Optec 3D Lidar scanner to provide high-resolution depth information with centimeter-scale accuracy up to 500 meters as well as terrain reflectance information.

The K10 robot performed the following tasks autonomously at BPLF:

- *Navigate to a location*. The robot moved to a designated waypoint location. The task was considered successful when the robot was within some threshold of the waypoint location.
- *Take a microimage*. The robot took an image using the microimaging camera. The task was considered successful when the image scan completed normally.
- *Take Lidar*. The robot collected Lidar data. Four types of Lidar

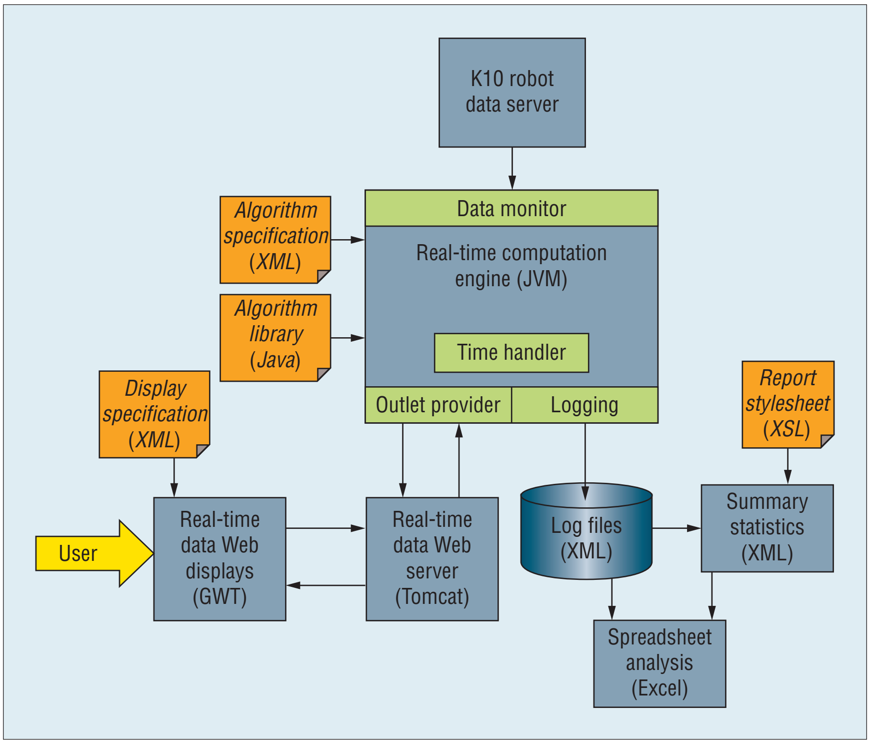


**Figure 2. Software for computing human-robot performance.**

subtasks were defined for robotic reconnaissance, including three single scans at low, medium, and high resolution and a 360-degree panoramic scan, with a low-resolution image scan taken every 30 degrees.

- *Take pancam images.* The robot took a sequence of images with the panoramic camera that was subsequently stitched into a panoramic image. Five types of pancam subtasks were defined for robotic reconnaissance, with image width and resolution varying from 6 to 136 frames per sequence.

Because the plan is intended for use by a robot, scheduled manual tasks are represented as wait tasks for the robot. During a wait task, the robot pauses all activities while a remote flight controller takes some action. The nature of the manual action is not specified in the robot's plan, but it can include manual teleoperation of the robot, or manual reconfiguration, such as changing the rover's maximum velocity.

Operators can interrupt the robot's execution of the plan by pausing an ongoing task. Once the robot is paused, they can manually teleoperate it or configure an instrument on the robot. Such unscheduled manual tasks were taken in response to problems that arose during operations. Problems requiring unscheduled manual action at BPLF included manually navigating the robot to recover from an autonomous navigation problem, securing the robot when environmental changes such as bad weather occurred (called safing), and restarting the robot controller in response to anomalous behavior. Once the manual task was complete, the execution of the plan could be resumed, or the plan could be aborted.

These capabilities were used during robotic reconnaissance to provide the following levels of autonomy when executing the plan:

- *Autonomous* refers to the scheduled tasks performed by the robot with no human action required. All tasks in the robot's plan were executed autonomously except the wait tasks.
- *Scheduled manual* refers to the scheduled tasks performed by the remote flight team. Wait tasks are the only manual tasks in this plan.
- *Unscheduled manual* refers to manual actions in response to a problem arising during the plan's execution. The ongoing task in the plan is paused prior to taking the manual action. When the action is complete, the plan can be resumed or aborted.

For the purposes of this study, manual actions taken when no plan was active are not considered nominal human-robot operations. As a result, we do not include such actions when measuring performance at these levels of autonomy.

## Measuring Performance with Adjustable Autonomy

Robot performance is measured during reconnaissance operations by monitoring robot telemetry and computing performance metrics in real time from that data. The same robot data stream used for remote ground operations is used to compute these metrics. Algorithms compute task performance measures from low-level data, including how much time the robot spends on each type of task and whether tasks complete normally. The resulting metrics are displayed in real time for use by flight controllers[12] and are archived in XML log files. These log files are transformed using XML Stylesheet Language (XSL) and XPath to derive summary statistics of daily and mission performance. Finally, both the data in the log files and the summary statistics are imported into an Excel spreadsheet for presentation as graphical charts. Figure 2 summarizes the software approach for measuring human-robot performance.

Metrics algorithms are encoded as Java object classes. XML configuration files associate these Java object classes with input data and define identifiers for referencing the computation results. Input to an algorithm can be rover telemetry or a reference to a computed value. At runtime, these XML configuration files are used to instantiate the specified Java objects and link them to rover telemetry messages or another algorithm's output. An instance of an algorithm's class is created for each specification of input data, permitting algorithms to be reused across multiple computations. Complex algorithms are composed by linking sequences of simpler algorithms—that is, one algorithm's output provides input to another algorithm. A computation is performed whenever an updated input value is received and activation conditions are met. For example, to compute the total distance a robot has traveled, robot-pose messages are passed to an algorithm for computing the distance between two poses. If the computed distance exceeds a threshold intended to exclude noisy data, the distance value is passed to a second algorithm that computes a running sum of these distances.

The state of all computed values can be saved during execution to a data log file in XML format. For the field test at BPLF, we saved log files at the end of each plan and at the end of each shift. The results we describe here are based on those log files. We used the logs created in real time because they are computed remotely from the robot and thus include statistics about data communication quality between the robot and ground control, such as the amount of time remote controllers were out of communication with the robot. These statistics provide important context for interpreting robot actions—for example, we might need to abort a plan due to an extended period with loss of signal (LOS).

For the robotic reconnaissance operation model, the role of manual intervention is to improve the rover's ability to execute plans successfully and collect the desired data in a timely and effective manner. To measure plan success, we compare the number of assigned tasks that are completed successfully with the number of tasks attempted but failed and the number of tasks abandoned. This comparison indicates the progress on mission objectives and the ability to complete assigned tasks. To measure team productivity, we compute the percentage of the work period spent doing tasks that achieve mission objectives (called *productive time*) and compare it to the time spent doing unplanned activities or doing nothing (called *overhead time*). For both these metrics, larger numbers indicate better performance. To measure team workload, we compute the percentage of the productive time spent at the levels of autonomy on each kind of task.

These metrics are not unique to human-robot operations with adjustable autonomy. We chose them because they represent a measurable performance baseline for the human-robot team that can be computed in real time and used during operations as well as compared across operational periods and missions. More specifically, we measured performance using the following algorithms.

### Plan State

Plan state is a qualitative assessment of the robot's progress on plan execution. Although four states are possible, only one state is true at any point in time: plan uploaded to robot, in plan executing, in plan idle, and out of plan. These states are computed from the execution status of the tasks in the plan whenever an updated status is received.

$$\text{PlanState}(t_i) = \begin{cases} \text{plan uploaded} & \text{status}(1, t_i) = -3 \\ \text{in\_plan\_exec} & \exists j: \text{status}(j, t_i) = -1 \\ \text{in\_plan\_idle} & (\exists j: \text{status}(j, t_i) = -2) \vee \\ & (\exists j > 1: \text{status}(j, t_i) = -3 \wedge \\ & \quad \exists j: \text{status}(j, t_i) = -1 \\ \text{out\_of\_plan} & \forall j: \text{status}(j, t_i) \geq 0 \end{cases}$$

where plan($t_i$) is the robot plan at time $t_i$, task($j$, $t_i$) is the ordinal task $j$ in plan($t_i$), and status($j$, $t_i$) is the execution status of task($j$, $t_i$). Enumerated task status values are failure > 0, successful = 0, executing = –1, paused = –2, and pending = –3.

### Task Times

The time spent performing a task begins to accumulate when the corresponding subsystem goes active and continues to accumulate until the subsystem goes inactive. A running sum of these task times for each type of robot task is computed over the shift.

$$\text{DriveTime}(t) = \sum_{i=1} t_{\text{end}(i)} - t_{\text{start}(i)}$$

where $t_{\text{start}(i)}$ is the time when the locomotor or navigator goes active, $t_{\text{end}(i)}$ is the time when the locomotor or navigator goes inactive, and $t_{\text{end}(i)} > t_{\text{start}(i)}$.

$$\text{InstrumentRunTime}(t) = \sum_{i=1} t_{\text{end}(i)} - t_{\text{start}(i)}$$

where $t_{\text{start}(i)}$ is the time when the instrument subsystem goes active, $t_{\text{end}(i)}$ is the time when the instrument subsystem goes inactive, and $t_{\text{end}(i)} > t_{\text{start}(i)}$.

The value of the task time is updated when the corresponding subsystem goes inactive. We compute task time for the Lidar, pancan, and micro-imager instruments.

### Time in Autonomous Operations

The time the robot spends operating autonomously is computed by summing the time intervals when the PlanState is in_plan_exec and the active task is not a manual task (that is, the task kind is not –1). This time is updated when a contiguous phase of autonomous operations ends.

$$\mathrm{TimeInAutoOps}(t) = \sum_{i=1} t_{\mathrm{end}\,(i)} - t_{\mathrm{start}\,(i)}$$

where kind(task($j$, $t_k$)) is the type of task $j$ at $t_k$ and –1 corresponds to a wait task,

$t_{\mathrm{start}(i)}$ is the time when PlanState($t_k$) = in_plan_exec $\wedge$ kind(task($j$, $t_k$)) $\neq -1$,

$t_{\mathrm{end}(i)}$ is the time when PlanState($t_k$) $\neq$ in_plan_exec $\vee$ kind(task($j$, $t_k$)) $= -1$,

and $t_{\mathrm{end}(i)} > t_{\mathrm{start}(i)}$.

### Time in Scheduled Manual Operations

The time a person spends performing scheduled manual tasks is computed by summing the time intervals when the PlanState equals in_plan_exec and the active task is designated a manual task (that is, the task kind is –1). This time is updated when a contiguous phase of scheduled manual operations ends.

$$\mathrm{TimeInSchedManualOps}(t) = \sum_{i=1} t_{\mathrm{end}\,(i)} - t_{\mathrm{start}\,(i)}$$

where kind(task($j$, $t_k$)) is the type of task $j$ at $t_k$ and –1 corresponds to a wait task,

$t_{\mathrm{start}(i)}$ is the time when PlanState($t_k$) = in_plan_exec $\wedge$ kind(task($j$, $t_k$)) $\neq -1$,

$t_{\mathrm{end}(i)}$ is the time when PlanState($t_k$) $\neq$ in_plan_exec $\vee$ kind(task($j$, $t_k$)) $\neq -1$,

and $t_{\mathrm{end}(i)} > t_{\mathrm{start}(i)}$.

### Time in Unscheduled Manual Operations

The time a flight controller spends performing unscheduled manual operations is computed by summing the time intervals when a task in the robot's plan is paused. This approach ignores the time spent taking manual action outside the plan.

$$\mathrm{TimeInUnschedManualOps}(t) = \sum_{i=1} t_{\mathrm{end}\,(i)} - t_{\mathrm{start}\,(i)}$$

where status($j$, $t_k$) is the execution status of task $j$ at $t_k$ and –2 corresponds to a pause status,

$t_{\mathrm{start}(i)}$ is the time when $\exists j$ : status($j$, $t_k$) = –2, $t_{\mathrm{end}(i)}$ is the time when $\forall j$ : $\neg$status($j$, $t_k$) = –2,

and $t_{\mathrm{end}(i)} > t_{\mathrm{start}(i)}$.

This algorithm was used for consistency with our definition of nominal operations being structured by the plan.

### Team Productivity

The productivity of the human-robot team is measured as the percentage of operations time spent by the robot and flight control team operating within the plan at some level of autonomy. Productive time—time spent directly accomplishing mission objectives—is measured as the sum of the time spent in autonomous, scheduled manual, and unscheduled manual operations:

$$\begin{aligned}\mathrm{ProductiveTime} &= \sum_{i=1} \mathrm{TimeInAutoOps}(t_i) \\ &+ \mathrm{TimeInSchedManualOps}(t_i) \\ &+ \mathrm{TimeInUnschedManualOps}(t_i)\end{aligned}$$

Overhead time is defined as the time doing activities other than tasks in the plan and, thus, time spent without progress on mission objectives. This includes the time when the rover has no plan, the plan is inactive, or the robot is waiting to start a plan that has been uploaded.

### Task Success

The ability of the human-robot team to complete assigned tasks indicates the team's progress on achieving mission objectives. We measure the percentage of assigned tasks completed successfully:

$$\begin{aligned}\%\mathrm{TasksComplete} &= (\mathrm{NumberOfSuccessfulTasks}\,/ \\ &\quad \mathrm{TotalNumberOfTasks}) * 100\end{aligned}$$

where

$$\mathrm{NumberOfSuccessfulTasks} = \sum \mathrm{task}(j, t_i)$$
$$\forall j : \mathrm{status}(j, t_i) = 0\,[\mathrm{succeed}]$$
$$\mathrm{TotalNumberOfTasks} = \sum \mathrm{task}(j, t_i) \quad \forall j$$

## Field Test Results

The protocol for human intervention in autonomous robotic operations at the beginning of the BPLF field test was to monitor for situations where intervention was needed and to intervene reactively with the goal of returning to autonomous operation as soon as possible. When quick emergency response was needed, the rover was emergency stopped (an EStop). If time permitted, the plan was paused while the ground operator took manual action.

Over the course of the field test, protocols for planned human intervention evolved from best practice. These protocols used adjustable autonomy to reduce risks to the robot (robot safety) and to improve the likelihood of mission success. For example, scheduled manual operations were used in areas with terrain that either represented a safety threat to the robot or that slowed the robot significantly when performed autonomously. For this protocol, the robot would automatically navigate to a location near the difficult terrain (such as steep ledges or very rocky surfaces). The ground operator would then manually control the robot's camera to visually select a site for imaging and teleoperate the robot to move it to the desired location. When out of the difficult area, autonomous operations would resume.

Flight controllers also used adjustable autonomy to develop workaround protocols during the field test when full autonomy was not possible. For example, it was planned to keep the Lidar powered up throughout the day and take Lidar scans as specified in the plan. However, during the field test, the Lidar instrument was sensitive to high temperature and tended to overheat at BPLF if left running throughout operations. The workaround developed during the field test was to constrain the amount of time that the Lidar was powered up to minimize the chance of overheating. Since the Lidar instrument did not support autonomous power-up and shut down for this test, a scheduled manual operation was required to power-up and shut down the instrument. In the short run, this use of adjustable autonomy let Lidar data be collected on high temperature days, which would have been impossible following the original operational protocol. It also points out a long-term need to adapt the Lidar instrument to support more autonomous operation. Without this flexibility, Lidar data would not have been collected on two of the three days K10 operated in the north region of BPLF.

The objective of measuring team performance during robotic reconnaissance was to characterize typical performance for operations using adjustable autonomy. For the field test, this performance baseline consists of mission averages for team productivity (Figure 3a), team workload (Figure 3b), and task success (Figure 3c).

## Team Productivity

The average team productivity for the field test was 39 percent of the total operating time spent in productive activities and 61 percent spent in overhead. Productive time consisted of 34 percent of the autonomous operating period, and 5 percent manual. The robot spent nearly twice as long in overhead as in productive time, indicating a need to improve robot utilization. Overhead time consisted of 34 percent out of plan, 22 percent inactive in a plan handling anomalies, and 5 percent waiting to start an uploaded plan. Because the robot had no plan available to execute for a third of the operating time, one way to improve robot utilization would be to reduce the time to generate new plans. The remaining portion of the overhead time was spent either waiting to start an uploaded plan or handling anomalies that suspended the plan. Figure 3a summarizes the percentage of time the robot and flight control team spent in productive and overhead activities for the entire field test.

## Team Workload

A significant percentage of the productive time (85 percent) was spent in autonomous operations, indicating the robot was able to perform assigned tasks independently most of the time. The types of autonomous tasks include driving to data collection sites and taking Lidar measurements, pancam pictures, or micro-images (MI). The largest percentage of productive time (39 percent) was spent taking pancam, and the smallest percentage of productive time (3 percent) was spent taking MI. Manual operations were performed only 5 percent of the total operating time, or 15 percent of the productive time. Scheduled manual operations were done on seven of the 10 days of operation, corresponding to 4.3 percent of the total operations time and 12 percent of the productive time. Such planned intervention was successful 75 percent of the time, indicating that the manual protocols used at BPLF were effective.

Unscheduled manual operations were performed on three days, corresponding to an average of 1.2 percent of the total operations time and 3 percent of the productive time. These unplanned interventions were successful 40 percent of the time, consistent with the reactive nature of these tasks.

Figure 3b shows the average team workload during the field test.

## Task Success

The ability of the human-robot team to accomplish tasks is measured as the percentage of assigned tasks that are completed successfully. On average for the field test, the human-robot team successfully completed a significant percentage of the planned tasks (72 percent). Of the remaining 28 percent of the tasks, 2 percent were attempted but not completed and 26 percent were never attempted. This corresponded to completing 324 of 449 tasks and aborting 115 tasks before they were started. Only 10 tasks were attempted but not completed. Attempting but not completing a task reduces team effectiveness because time is expended without achieving the intended task objective. Tasks might never be attempted because they cannot be performed or because they are contingent upon what was observed during earlier tasks.

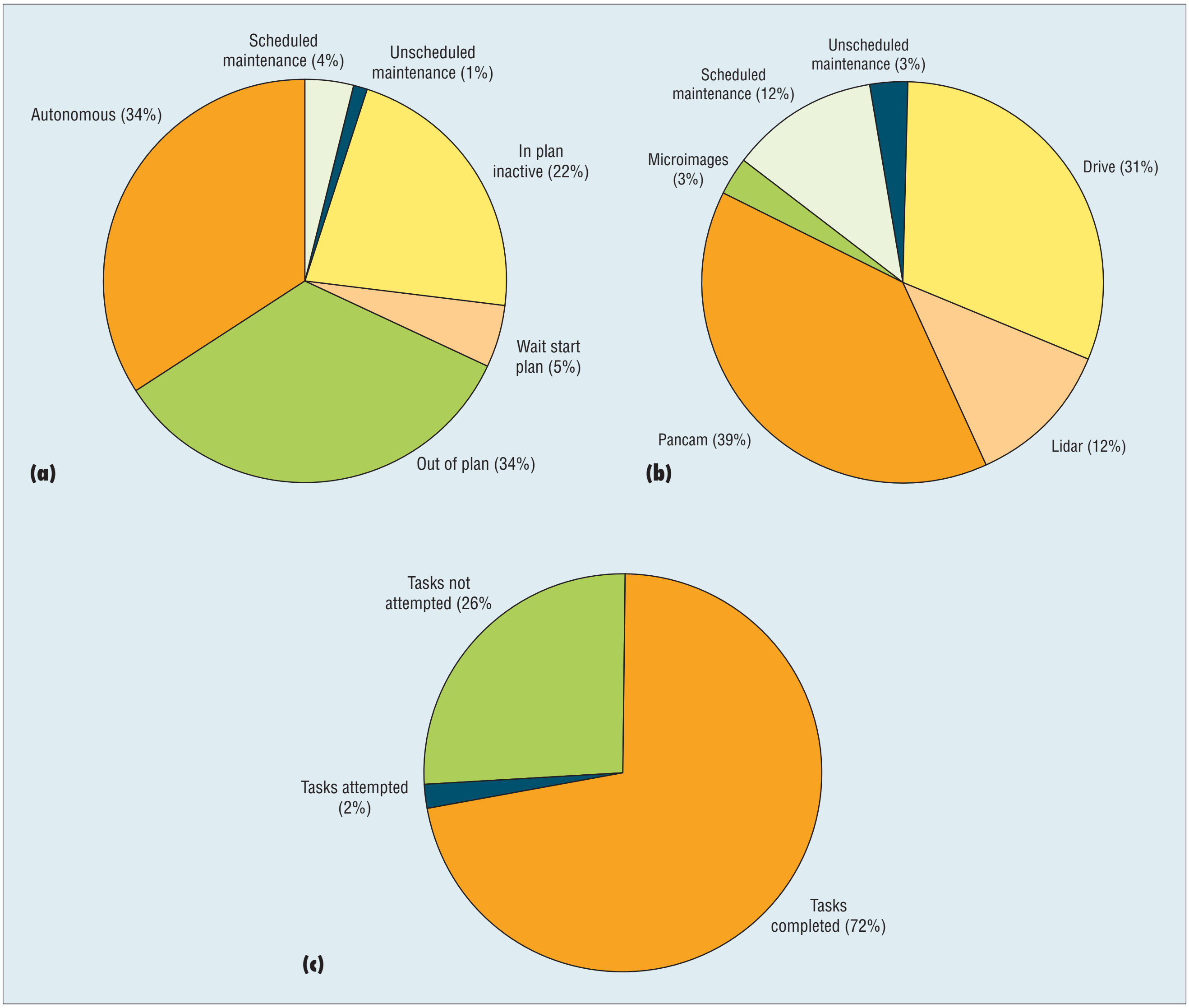


**Figure 3. Field-test results for the Black Point Lava Flow, Arizona. We measured the (a) average team productivity (15–26 June), (b) average team workload (17–26 June), and (c) average task success (15–26 June) while testing the K10 robot.**

Figure 3c summarizes the average progress made on planned tasks by the human-robot team for the field test.

The daily averages for team productivity and task success indicate that human-robot performance at BPLF was highly variable. The minimum team productivity and task success occurred on 19 June. With 5.4 percent productivity, the daily productivity was 34 percent below the average for the field test. Only 7.7 percent of planned tasks were completed successfully this day, more than 60 percent below the average for the field test.

The highest productivity of 64.9 percent was observed on 25 June. This is significantly more than the average productive time of 39 percent for the field test. The maximum observed task success was 100 percent on 16 and 26 June. Tables 1 through 3 summarize the average, minimum, and maximum values for daily productive time, overhead time, and task success measures.

The environment at BPLF impacted the ability to perform tasks and contributed to the high percentage of time spent in overhead (61 percent average overhead for the field test). Three types of environmental effects were observed during the field test: bad weather, high temperatures, and challenging terrain. Generally these environmental effects increased the time spent inactive in the plan or the time out of plan.

The terrain was particularly difficult to traverse on 18 and 20 June. On these days, the robot spent between 45 and 55 percent of its time with an inactive plan. This is more than twice the amount of time spent with an inactive plan on the remaining days of the field test (ranging from 4 to 17 percent).

Physical location was also correlated to degraded communication quality, which significantly reduced team productivity. The time in LOS ranged from 20 to 40 percent from 18 to 22 June, when operating in the western area of BPLF. Correspondingly, the percentage of productive time on all four days was below average.

**Table 1. Daily productive time.**

| Productive time (PT) | Average (%) | Minimum (%) | Maximum (%) |
|---|---|---|---|
| Auto operations | 34 | 2.7 | 57.5 |
| Scheduled manual | 4 | 0 | 11.6 |
| Unscheduled manual | 1 | 0 | 4.8 |
| Total PT | 39 | 5.4 | 64.9 |

**Table 2. Daily overhead time.**

| Overhead time (OT) | Average (%) | Minimum (%) | Maximum (%) |
|---|---|---|---|
| Out of plan | 34 | 11.7 | 86.6 |
| In plan inactive | 22 | 0.6 | 51.5 |
| Wait start plan | 5 | 0.3 | 9.7 |
| Total OT | 61 | 35.0 | 95.0 |

**Table 3. Daily task success.**

| Task success | Average (%) | Minimum (%) | Maximum (%) |
|---|---|---|---|
| Tasks completed | 72 | 7.7 | 100 |
| Tasks failed | 2 | 0 | 40.0 |
| Tasks abandoned | 26 | 0 | 88.5 |

The team performance was measured during robotic reconnaissance both to characterize the typical performance for such operations and to detect departures from typical performance. The mission averages computed during the field test at BPLF for task success, team productivity, and team workload represent a step toward establishing a performance baseline. Because performance measures are computed in real time, they can be compared to this baseline to identify degraded performance during operations and to determine the performance effects of adjusting operational protocols. Future mission planning could also use this baseline to establish performance expectations.

Equally important is characterizing the circumstances that can degrade team performance from this baseline. This includes both identifying the conditions causing the degradation and determining how frequently these conditions occur. Conditions at BPLF that reduced team productivity included difficult terrain and bad weather as well as delays in the availability of plans from the science team (the rover spent an average of 24 minutes waiting for a plan). High temperatures impacted the ability to complete operations such as Lidar tasks. The most significant impact to operations at BPLF was poor communication quality. Overhead time increased by up to 35 percent when time in LOS exceeded 20 percent. Such degraded communication quality occurred when operating in the western area of the BPLF. This indicates a need to revisit protocols for robot autonomy during periods of poor communication to investigate robot utilization when out of communication with the ground team.

The novel aspect of our approach is that metric values are computed inline while the team is performing activities, which makes them available for use during operations. Comparing current performance to typical performance can help detect anomalies requiring an operational change or workaround. For example, more frequent reactive interventions by the ground team can indicate a need to change an autonomous protocol. Inline processing of metrics also eliminates the significant task of postprocessing the terabytes of data collected during a mission to compute summaries of mission performance.

A challenge in automating the computation of performance metrics is determining when to compute metric values. The operational events that provide this context for computation are often more complex than the metrics algorithms. Part of this complexity arises from missing information, requiring that important context be deduced from patterns in telemetry. For example, the K10 rover does not have telemetry indicating when it is being teleoperated. This must be inferred by monitoring for patterns of robot motion when there is no autonomous task causing that motion. Part of this complexity results from the need to handle data imprecision. For example, the K10 robot's motion can be detected by monitoring changes in robot pose. To accurately determine when the rover is in motion, however, it is necessary to look for changes in pose that exceed the

## THE AUTHORS

**Debra Schreckenghost** is a senior scientist with TRACLabs in Houston and a principal investigator of NASA projects developing new technology for remote human-robot operations. Her research interests include adjustable autonomy, intelligent software agents, and human-robot interaction. Schreckenghost has an MS in electrical engineering from Rice University. Contact her at ghost@ieee.org.

**Tod Milam** is a research scientist and the lead software engineer for TRACLabs' DCI agents for human-automation interaction in mission operations. He is currently developing new technology to support remote interaction with NASA robots. Milam has a BS in computer science from Drake University. Contact him at tmilam@traclabs.com.

**Terrence Fong** is the director of the Intelligent Robotics Group at the NASA Ames Research Center. His research interests include field robotics, human-robot interaction, virtual reality user interfaces, and parallel processing. Fong has a PhD in robotics from Carnegie Mellon University. Contact him at terry.fong@nasa.gov.

noise level on the data. And part of this complexity results from inconsistent operations. That is, a K10 plan is considered done when there is no task executing, paused, or pending. The protocol for aborting a plan is to abort the remaining tasks within the plan. It is possible, however, to simply uplink and start a new plan without following this protocol. Thus, to accurately detect the end of a plan, it is necessary to monitor for a plan uplink as well as the status of tasks in the current plan.

For this study, overhead time is characterized with respect to the science plan—time spent waiting to start a plan, suspended within an active plan, or outside of the plan. Better algorithms are needed for measuring what the human-robot team is doing during overhead time. An improved understanding of the activities conducting during overhead time should help us identify ways to reduce the overhead time. In particular, techniques are necessary for tracking the robot's manual operation outside the plan, including techniques to determine when the robot is being teleoperated and whether the teleoperation originates at the test site or remotely. ■

## Acknowledgments

NASA funded this research under the Small Business Innovative Research, Exploration Technology Development Program (Human-Robotic Systems), and Exploration Systems Mission Directorate Analogs programs. We thank Hans Utz, David Lees, and Mark Allan for their assistance in integrating the performance monitoring software with the K10 robot and ground data system. We also thank Estrellina Pacis for her assistance in defining metrics for operations assessment and Matt Deans for his help in understanding the robotic reconnaissance planning process.

## References

1. S. Chien et al., "The Future of AI in Space," *IEEE Intelligent Systems*, vol. 21, no. 4, 2006, pp. 64–69.
2. T.B. Sheridan, "Allocating Functions Rationally between Humans and Machines," *Ergonomics in Design*, vol. 6, no. 3, 1998, pp. 20–25.
3. F. Heger and S. Singh, "Sliding Autonomy for Complex Coordinated Multi-Robot Tasks: Analysis & Experiments," *Proc. Robotics: Systems and Science*, 2006; www.roboticsproceedings.org/rss02/p03.html.
4. D. Schreckenghost et al., "Remote Task-Level Commanding of Centaur over Time Delay," *Proc. Space Technology Applications and Int'l Forum 2008*, AIP, 2008, pp. 784–791; http://scitation.aip.org/dbt/dbt.jsp?KEY=APCPCS&Volume=969.
5. P. Scerri et al., "A Prototype Infrastructure for Distributed Robot-Agent-Person Teams," *Proc. 2nd Int'l Joint Conf. Autonomous Agents and Multiagent Systems*, ACM Press, 2003, pp. 433–440.
6. D.J. Musliner and K.D. Krebsbach, "Adjustable Autonomy in Procedural Control for Refineries," *Proc. AAAI Spring Symp. Adjustable Autonomy*, AAAI Press, 1999, pp. 81–87.
7. R.P. Bonasso, D. Kortenkamp, and C. Thronesbery, "Intelligent Control of a Water Recovery System: Three Years in the Trenches," *Artificial Intelligence*, vol. 24, no. 1, 2003, pp. 19–44.
8. D. Schreckenghost et al., "Adjustable Autonomy with NASA Procedures," *Proc. 9th Int'l Symp. Artificial Intelligence, Robotics and Automation in Space* (i-SAIRAS), 2008; http://robotics.estec.esa.int/i-SAIRAS/Proceedings.html.
9. J. Crandall et al., "Validating Human-Robot Interaction Schemes in Multitasking Environments," *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 35, no. 4, 2005, pp. 438–449.
10. J. Crandall and M.L. Cummings, "Developing Performance Metrics for the Supervisory Control of Multiple Robots," *Proc. ACM/IEEE Int'l Conf. Human-Robot Interaction*, ACM Press, 2007, pp. 33–40.
11. T. Fong et al., "Field Testing of Utility Robots for Lunar Surface Operations," *Proc. AIAA Space*, AAAI Press, 2008; www.ri.cmu.edu/person.html?type=publications&person_id=388.
12. D. Schreckenghost et al., "Measuring Robot Performance in Real-time for NASA Robotic Reconnaissance Operations," *Proc. NIST PerMIS Workshop*, NIST special publication, 2009; www.ri.cmu.edu/person.html?type=publications&person_id=388.