Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 54 (2006) 174–183
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Mission-level path planning and re-planning for rover exploration

Paul Tompkins*, Anthony Stentz, David Wettergreen

The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15232, United States

Received 9 March 2005; accepted 13 September 2005
Available online 9 January 2006

Abstract

The Life in the Atacama (LITA) project seeks to develop technologies for robotic life detection and apply them to the investigation of the Atacama Desert. Its field investigation in 2005 will demonstrate highly autonomous robotic science and daily multi-kilometer traverses over several weeks. A key component is mission-level path planning, which combines large-scale route selection, path and activity timing, and predictive energy management. Its purpose is to yield high-level plans for locomotion, solar charging and hibernation. We describe this new level of robotic autonomy and illustrate its performance from the field experiments in 2003.

Keywords: Autonomous navigation; Planetary exploration; Planning; Execution; Re-planning

1. Introduction

The Life in the Atacama (LITA) project seeks to develop technology in support of robotic astrobiology for NASA while conducting useful Earth science in the Atacama Desert of northern Chile [17]. The Atacama is one of the driest places known, and has been described as the most lifeless place on Earth. The project intends annual field seasons of rover testing and biological investigation in this environment. To stress robot autonomy for this and future planetary missions, scientists in the United States will conduct their investigation remotely and will be limited by communication opportunities and bandwidths analogous to a Mars mission. They will characterize the presence and distribution of microorganisms over more than 100 km of travel in an effort to better understand the limitations of life.

The LITA project seeks to multiply the science data return possible in future planetary missions by enabling rovers to autonomously investigate multiple widely-distributed science targets in a single command cycle, for example in one Martian sol. Toward this end, a central problem is to automatically solve for plans that meet navigational, scientific, temporal and resource requirements, and to repair plans if necessary, to derive the greatest return from the rover despite unexpected events that arise during execution. In contrast to the NASA Mars Exploration Rover (MER) mission which requires many command cycles to achieve each science target, the LITA survey concept entails navigation distances of kilometers and science observations at multiple locations in each cycle. LITA's greater emphasis on mobility prompted an approach rooted in path planning rather than more traditional constraint-based planning and scheduling techniques (for example [1]).

Most research into path planning on natural terrain has focused on providing safe, goal-directed travel over short distances amidst obstacles of roughly rover scale [3–5,7,10,15]. All of these works are enabling of global navigation in that local navigation for obstacle avoidance is needed before global path planning. Some techniques generate time-optimal paths, typically to satisfy vehicle kinematic and dynamic constraints. Given the slow pace of planetary vehicles, dynamics were considered to be out of scope. Kobilarov and Sukhatme illustrate the strength of randomized search as applied to natural terrain navigation, at the expense of optimality [2]. Research into energy-optimal path planning is rare. Shillcut [8] analyzed the motion of the sun to select amongst coverage patterns, but did not incorporate these analyses into path planning. Sun and Reif present a technique for generating minimum energy cost paths through models of natural terrain [12], but do not consider the problem where stored energy is non-monotonic. None of the time- or energy-optimal approaches emphasize efficient

* Corresponding author.
E-mail addresses: pauldt@email.arc.nasa.gov (P. Tompkins), tony@ri.cmu.edu (A. Stentz), dsw@ri.cmu.edu (D. Wettergreen).

0921-8890/$ - see front matter
doi:10.1016/j.robot.2005.09.027

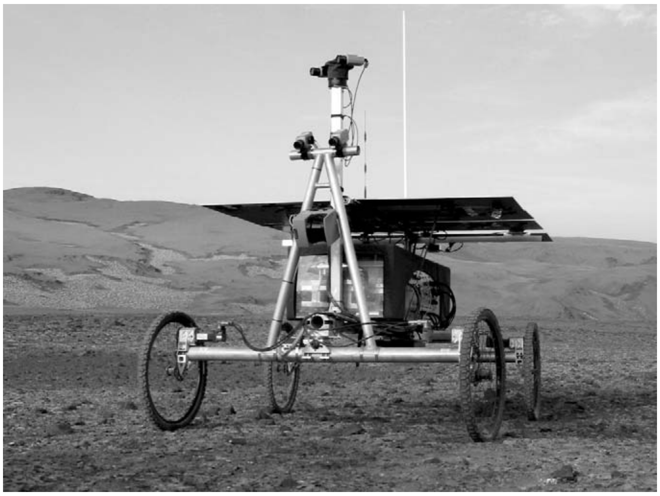

Fig. 1. The Hyperion robot, Atacama Desert, Chile, April 2003.

re-planning. Aspirations for long-distance autonomous traverse motivate us to extend the prior work.

Our approach, called *mission-level path planning*, operates at a higher level of abstraction than previous navigation planning approaches. It models locomotion amidst large-scale terrain like hills and valleys rather than over and between rocks, but at a resolution of tens of meters rather than tens of centimeters. In addition, it considers parameters that do not factor heavily in local navigation but are critical to mission success at the mission level—time, resources, and operational constraints. Since commands interleave long traverses with science and other activities, the mission-level path planning approach must also take into account the temporal and resource requirements of non-locomotion activities. A primary goal is to bridge the gap between path planning and task planning and scheduling.

In April 2003, building upon technology developed in the Sun-Synchronous Navigation project [16], the first LITA field experiment used the Hyperion robot (Fig. 1) to test a number of technologies including TEMPEST, a mission-level path planner. Over eight days, TEMPEST provided plans and re-plans for Hyperion that avoided hazardous terrain, considered the solar energy available to the robot, and satisfied operational constraints.

In this article we begin by describing the mission-level path planning approach (Section 2), explain the current implementation (Section 3) and then examine planning results from Atacama field experiments. We compare our work to other relevant robot field work (Section 4) and conclude with a critique of this year's accomplishments and suggestions of future work (Section 5).

2. Planning approach using TEMPEST

TEMPEST is a mission-level path planner [13,14]. At its core, the planner relies on an algorithm called Incremental Search Engine (ISE). TEMPEST applies models of the world, rover capabilities, rover actions, action constraints, and mission objectives. It translates these into a state space, state transition and cost functions, an objective function and the specification of the start and goal states understood by ISE. ISE then finds optimal solutions to the problem presented by TEMPEST.

2.1. Incremental Search Engine (ISE)

TEMPEST uses a search algorithm called Incremental Search Engine (ISE) to search for plans. ISE, a generalization of the D* algorithm [9], is a graph-theory based, heuristic search algorithm optimized for globally constrained planning and re-planning in high-dimensional spaces. The algorithm is complete, and resolution optimal. The algorithm is general and abstracts away domain-specific information. For further information see [11].

ISE encodes its state space using two types of discrete variables. IPARMS are independent parameters. DPARMS are dependent parameters represented at two resolutions: the values are at sufficiently high resolution to model the state, but are also grouped into low resolution equivalence classes. An IPARMS–DPARMS pair collectively defines a state. For example, the LITA TEMPEST planner defines two spatial IPARMS variables (X, Y) at 30 m resolution, and a DPARMS variable for absolute time (T) at one second resolution and with equivalence classes defined in 30 min bins.

ISE performs a backwards-chaining search, beginning from one or more goal states. Using a best-first approach directed by the objective function and an admissible heuristic, ISE prioritizes the states to "expand". ISE expands a state by simulating all possible arcs to neighboring IPARMS states, in reverse time, to generate ancestor states. For each arc a, an application-specific state transition function defines the change in DPARMS in response to changes in IPARMS. Each resulting state becomes a new node in a directed graph, from which one of the goals is reachable. For each transition, ISE enforces local constraints by preventing illegal action-state combinations, and global constraints by rejecting states whose DPARMS or objective functions exceed thresholds on cumulative path effects. For LITA, the state transition function computes the duration for various actions (e.g. Drive, Charge) and takes the form $\Delta T = f(\Delta X, \Delta Y)$, where $\Delta X = f(a)$ and $\Delta Y = f(a)$. All arcs result in a positive cost. A monotonically increasing objective function, associated with each state, accumulates the cost. This might be a traditional distance metric or an energy function. ISE derives paths that are minimum cost according to the objective function. If a solution exists, the search will produce states whose IPARMS match those of the start state (defined by the state of the robot). The optimal path begins at the lowest-cost feasible start state, as defined by the objective function.

In problems without global constraints, ISE and A* [6] yield similar plans (with the same cost but subject to variations in tie-breaking). In an initial search, they also display similar computational performance. However, if transition costs change locally, ISE operates far more efficiently than A* in re-planning paths. Where changes would force A* to re-build its search from scratch, ISE uses incremental graph theory techniques to repair both the feasible set of solutions and the optimal

Table 1
World and rover models used to define the mission-level navigation domain for LITA

Domain model	Description
Terrain	• Terrain elevation and slope. • Map projection and ellipsoidal geodetic reference enable transformations between maps and Cartesian coordinates or latitude/longitude.
Ephemeris	• Ephemeris software predicts the relative positions and orientations of Solar System bodies.
Lighting	• Lighting maps encode instantaneous sun-incidence on the terrain. • Sequences of lighting maps represent time-varying lighting on terrain.
Solar flux	• Prediction of solar power (W/m^2) as a function of time.
Locomotor	• Power load computed from mass, maximum speed, effective coefficient of friction, and drive train efficiency.
Solar array	• Power source computed from area, cell efficiency, and orientation.
Battery	• Energy storage limited between minimum and maximum charge (W h).
Electronics	• Power load as constant value.
Navigation cameras	• Orientation and field-of-view for sun-in-camera constraint calculations.

path within it. Efficient re-planning makes ISE very well suited for robot navigation planning in partially-known environments, where information is added incrementally.

ISE enables two types of re-planning: state update re-planning and model update re-planning. In response to updates in initial state, ISE extends its existing search graph to the new IPARMS state, and re-applies its feasibility criteria to candidate solutions arriving there. If underlying models change, the original search graph is invalid. ISE propagates the new costs, as derived from the model changes, and repairs the graph to yield a new optimal plan. The algorithm is time efficient because it determines which portions of the search graph are affected by new information and limits the re-computation to those portions. The algorithm is space efficient through the use of three mechanisms:

Dynamic State Generation: ISE creates a state when it is needed and deletes it when it is pruned. This precludes the need to allocate an entire multi-dimensional space when only a small part of it may be searched.

State Dominance: ISE prunes dominated states to minimize state proliferation. Dominance exists when a state can be proved to be always less expensive than another state, in terms of the objective function. ISE employs user-defined state dominance relationships between states that share the same IPARMS.

Resolution Pruning: Within each coarse resolution DPARMS equivalence class, ISE prunes all but the least expensive states, according to the objective function. This feature can dramatically reduce the number of states while still preserving resolution optimality.

ISE applies one of two search modes:

BESTPCOST finds the minimum cost path. In this mode, the feasible plan with the lowest path cost, as defined by the objective function, is the optimal solution.

BESTDPARMS finds the best state solution, as defined by a user-defined "better" function, below a maximum path cost. The objective function serves only to measure path costs against the maximum. The "better" function evaluates DPARMS to prioritize plans that are equivalent in path cost to determine the best solution.

ISE is a general-purpose, discrete-path search engine. To apply ISE to a specific application, a developer must define the domain in terms of the state space S, the actions A, the state transition function $S \times A \rightarrow S$, the start and goal states, and the conditions for feasibility and optimality.

2.2. *Domain models*

TEMPEST comprises domain models, summarized in Table 1, that are the foundation of the domain definition models of the world and the rover that portray the relevant characteristics of the mission-level path planning problem.

2.3. *State space*

For LITA the TEMPEST state space comprises four dimensions—two spatial, one time, and one energy dimension. The two terrain map coordinates specify two position IPARMS X and Y, in units of map grid cells, and absolute time T, in seconds, specifies a DPARMS variable.

The fourth dimension, battery energy E, expressed in Watt hours, is more complicated both in its semantics and how it is represented. It is important to clarify that E does not represent the instantaneous energy in the battery, but rather the *minimum required* energy (state of charge) to *reach the goal*. Because ISE searches backwards, it requires an exact goal state (x_g, y_g, t_g, e_g) from which to begin its search, where e_g is the minimum goal energy. How must energy in ancestor states be interpreted?

Actions that accumulate energy in the forward-time direction (e.g. solar charging) decrease the value of E in the backwards search. One or more successive "positive energy" actions, run backwards during the search, might drive E to zero. Herein lies the subtlety: $e = 0$ does *not* indicate that ISE should abandon the path instance. Instead, $e = 0$ indicates the goal could be reached from the current position and time, *even starting with no stored energy*. During the backwards search, ISE prevents E from dipping below zero—there is no physical meaning to an energy state with "less-than-empty" conditions.

Alternatively, actions that discharge the battery in the forward-time direction (e.g. nighttime locomotion) will tend to

Table 2
Actions coarsely describe activities relevant to LITA mission-level path planning

Action	$\Delta X, \Delta Y$	ΔT	Active subsystems and description
Drive	$0/\pm 1, 0/\pm 1$ grid cells	$\Delta T_D = f(\Delta X, \Delta Y)$	• Locomotor, Solar array, Battery, Electronics, Nav. cameras • Separate action for each of 8 neighboring cells
Charge	0,0 cells	$\Delta T_C = 900$ s	• Solar array, Battery, Electronics • Assumes a heading to achieve the optimal sun angle on array.
Hibernate	0,0 cells	$\Delta T_H = 1800$ s	• Solar array, Battery, Electronics • A low power option for extended low solar energy conditions.

Table 3
Local constraints enforce operational limitations on LITA actions

Constraint	Prevents	Applied to action
Slope	Actions on terrain that is too steep.	Drive, Charge, Hibernate
Daylight	Actions at night.	Drive, Charge
Direct sun	Actions in shadows or at night.	Charge
Sun-in-camera	Actions when the sun appears in the camera field-of-view.	Drive

cause E to increase in a backwards search. This could cause the value of E to exceed battery capacity. Perhaps counter-intuitively, ISE should abandon such a path—it requires a "more-than-full" charge to achieve the goal. Put simply, low E is good, and high E is bad.

A further complication with E is in its representation within ISE. In contrast to position and time, it is not represented as either an IPARMS or DPARMS variable, but within the objective function. We describe how in Section 2.6.

2.4. Actions, constraints and the state transition function

Developers define a set of coarse sequential actions for TEMPEST—driving, rover maintenance and survival, and science investigation activities—that have an effect on rover position and resources at the mission level of granularity. Table 2 summarizes the actions used for LITA, and resulting IPARMS and DPARMS state transitions. Mobile actions specify a change in position—the duration is a function of this motion. Stationary actions have fixed durations, but can be executed in series to provide longer actions. Path solutions from ISE are fully-ordered sequences of state-action pairs, or waypoints.

The developer must also specify a set of local constraints, defined as functions of the world and rover models, which limit actions to a subset of the state space. For example, a constraint to prevent the sun from blinding the camera combines a sun ephemeris model and a model describing the rover camera mounting angles and field-of-view. Table 3 lists some of these local constraints and how they might be applied to actions.

Each action and its associated constraints define a state transition function. The function uses a path integrator that calls on the world and rover models to compute new states. Changes in the world, rover, action or constraint models during plan execution require re-planning. Typically, only changes to the world model are localized. Global changes require repairs to the entire ISE graph, and therefore are not handled efficiently.

2.5. Start and goal specification

In backwards chaining, the goal states must be fully specified before search begins. Goal positions and energies are fixed by the mission objectives, but arrival time is often unconstrained. TEMPEST accommodates this by utilizing the ISE multiple goal mechanism. TEMPEST estimates a likely range of arrival times—a "goal window"—based on optimistic and pessimistic projections on plan duration. At uniform time intervals in this window, TEMPEST designates separate ISE goal states that share common position and energy values. The ISE search does not distinguish between paths growing from different goals. Path solutions are free to terminate at any time interval within the goal window.

TEMPEST must also specify the start state so that ISE can determine which paths are feasible solutions. It designates the nearest cell in the terrain model as the IPARMS position start (x_s, y_s) and an allowable "start window" that spans the current time. TEMPEST must ensure that a plan starting energy, that is, the required energy to reach the goal does not exceed the current robot battery energy e_s.

2.6. Feasibility and optimality

Using this specification of the start and goal states, ISE searches for feasible start states from which to select an optimal start state and associated plan. A plan P is composed of N waypoints w_i as follows:

$$P = \{w_1, \cdots, w_N\} \text{ where } w_i = (x_i, y_i, t_i, e_i) \text{ and } t_i < t_{i+1}.$$

Given a start state $s = (x_s, y_s, t_s, e_s)$ and a start window time interval $t_s \in [t_{wi}, t_{wf}]$, a start state is feasible if and only if:

$$(x_1 = x_s) \wedge (y_1 = y_s) \wedge (t_{wi} \le t_1 \le t_{wf}) \wedge (e_1 \le e_s).$$

For LITA, TEMPEST plans are optimal in combined terms of battery energy and plan length. Unlike path duration or distance, whose quantities increase monotonically as a plan gets longer, energy is non-monotonic: locomotion and other activities expend energy while solar energy and other power

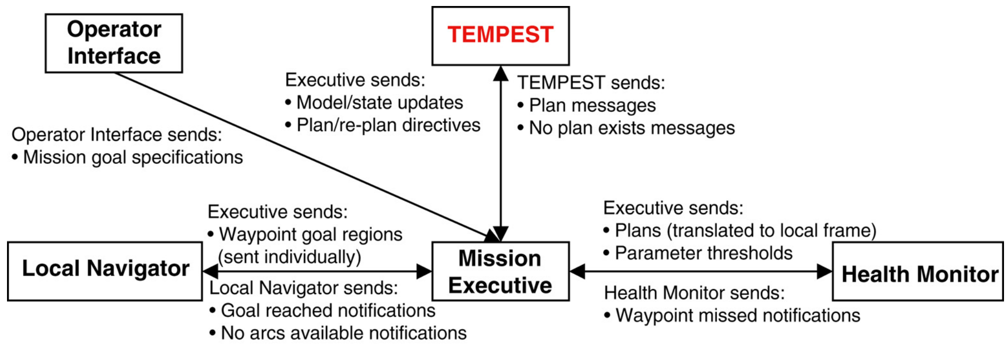

Fig. 2. The mission-level planning and execution modules for Hyperion.

sources restore it. Standard heuristic search approaches to planning would become "stuck" in states providing a net positive energy intake, and would never actually reach the designated goal.

To prevent this behavior, TEMPEST uses ISE in its BESTPCOST mode and a specialized objective function that yields energy-optimal paths and also collapses the search space from 4-D to 3-D. ISE enables a composite objective function consisting of two or more elements that can be tracked and manipulated separately, but that collectively contribute to a single objective function cost. To describe the specific composite objective function used for LITA, we define the quantity $E_{\max}$:

$$E_{\max} = \left| \min_{\forall (s \in S, a \in A, \Delta e \le 0)} \Delta e = f(s, a) \right|.$$

In words, $E_{\max}$ is the absolute value of the greatest single-action negative cost (positive increment) to energy over the entire state and action space. The value of the TEMPEST objective function is the sum of two quantities:

$$f = L + B \qquad \text{where } L = nE_{\max} \text{ and } B = \sum_{i=1}^{n} \Delta e_i$$

L is a measure of plan length in increments of $E_{\max}$, and B is the sum of energy costs (positive and negative) over the path. Since over any possible action $E_{\max} + \Delta e \ge 0$, the objective function increases monotonically over the path, and so is legal under heuristic search. Using this objective function under the BESTPCOST mode, ISE produces paths that are optimal in combined terms of minimum plan length and energy cost.

The B term in the objective function can be used to track the E state variable. Further, since changes in the X, Y and T variables are totally independent of E in all TEMPEST applications to date (i.e. $\Delta X \ne f(\Delta E)$, $\Delta Y \ne f(\Delta E)$, $\Delta T \ne f(\Delta E)$), E can be removed from the search space. Collapsing the four-dimensional search to three dimensions drastically reduces the computation and memory for search, thereby enabling TEMPEST to run online and to solve larger planning problems.

3. Experimental system

Hyperion is a solar-powered robot originally designed for sun-synchronous navigation in polar latitudes [16]. The Atacama Desert differs from the Arctic in a number of respects, most notably in latitude, which affects sun elevation angles, diurnal lighting and the navigation strategy. For the Atacama, Hyperion's solar array was oriented horizontally to best collect energy from the overhead sun. Even so, its batteries last only two hours while driving without the sun's input, so it is critically dependent on its environment.

The Hyperion autonomy software comprises several modules. TEMPEST is the sole mission planner, and provides plans comprising Drive, Charge and Hibernate actions that guide execution in terms of route, event timing and minimum battery charge. The Navigator (NAV) avoids local obstacles while seeking globally-referenced positions defined by TEMPEST waypoints. A Health Monitor (HM) identifies abnormal state conditions while operating. A rudimentary mission executive (ME) coordinates mission-related data passing between these modules, and receives and distributes traverse commands from the Operator Interface (OI). Fig. 2 illustrates the basic set of inter-module communications relating to mission-level path planning and execution.

Upon receiving a traverse command from the OI, the ME determines the current robot position and time state, and then requests a plan from the initial state to the specified goal state. TEMPEST finds and returns an optimal plan to the ME. The ME initiates and oversees each of the actions in the TEMPEST plan.

For Drive actions, the ME computes parameters for a 10 m by 30 m goal region surrounding the next globally-referenced position waypoint (Fig. 3), and sends them to the Navigator for execution. Using the goal region as its global goal, the Navigator uses an arc evaluation/incremental search algorithm [9] to pursue this region while avoiding obstacles it detects using stereo vision. Once Hyperion is within the region, as determined by fusing GPS with inertial and wheel odometry measurements, the Navigator signals its arrival, completing the action. For Charge and Hibernate actions, the ME stops the rover and then waits for the target duration before signaling completion.

During plan execution, at the scheduled arrival time for each TEMPEST waypoint, the HM performs a check to confirm the robot is on time. If the rover is more than a fixed distance from the waypoint at the scheduled arrival time, the HM indicates the waypoint was missed and requests a re-plan. The ME

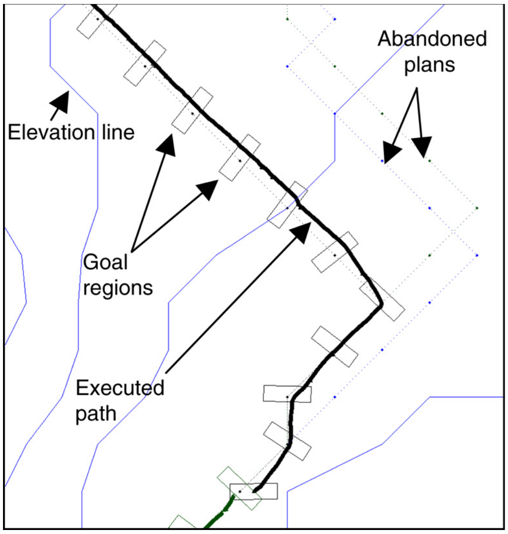

Fig. 3. Plans and executed paths, with local navigation goal regions.

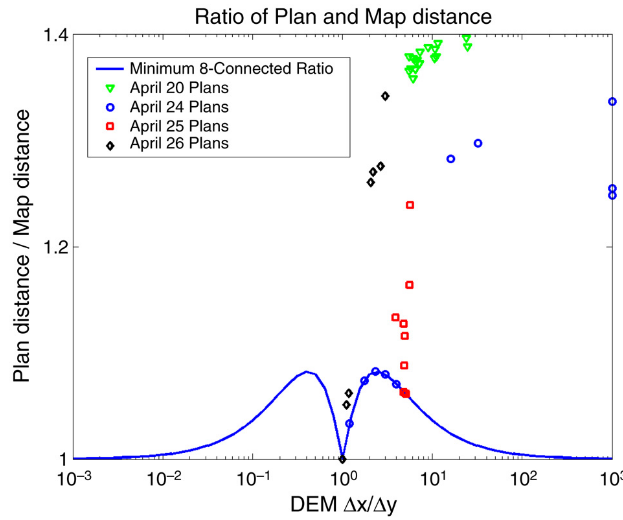

Fig. 4. The eight-connected path artifact was seldom a dominant effect in path length.

terminates the execution of the current plan, and TEMPEST uses the current rover state and cached goal to find a new plan.

In earlier field experiments [13], TEMPEST was run offline to produce a single plan for a day's activities. For LITA in 2003, as a first step toward comprehensive re-planning, TEMPEST uses ISE state-update re-planning to revise plans as the state evolves during plan execution. Re-planning and software performance enhancements transform TEMPEST into an online planner. Previously, plans were generated once per traverse, and relied on perfect execution to remain valid. For LITA, TEMPEST repairs plans as the execution evolves.

4. Field experiment

Over April 17–20 and 24–26, TEMPEST generated 27 plans and 83 re-plans. Our goal was to evaluate TEMPEST in terms of resulting path length, terrain avoidance, energy and operational utility.

4.1. Path length

The ISE grid representation of position interferes with the ability to always produce shortest-distance paths. The grid enforces motion on an eight-connected graph connecting each cell to its nearest neighbors. The minimum eight-connected plan distance between two points depends on the ratio of their relative distance in the x-coordinate (Δx) and y-coordinate (Δy). When the start and goal lie along the horizontal ($\Delta x/\Delta y$ = infinity), vertical ($\Delta x/\Delta y = 0$) or principal diagonal axes of the graph ($\Delta x/\Delta y = 1$ or $\Delta x/\Delta y = -1$), the minimum eight-connected plan distance is equal to the Euclidean distance between points. Since the eight-connected path cannot assume arbitrary headings, other ratios of Δx-to-Δy produce minimum path lengths that exceed the distance between the points.

In Fig. 4 we examine plans generated during the field experiment to determine the degree to which eight-connectedness was dominant in extending path length beyond the Euclidean distance. The horizontal axis spans the range of the absolute value of $\Delta x/\Delta y$, representing an East–West heading on the left, a Northeast–Southwest or Northwest–Southeast heading at the center, and a North–South heading on the right. The vertical axis spans a range of the plan distance divided by the Euclidean or map distance. The solid curve shows the minimum plan-to-map distance ratio for the range of ratios of Δx and Δy. The markers indicate the relationship between grid distance ratio and plan-to-map ratio for the plans from April 20 through April 26. Observe that all the plans fall to the right of center, confirming that routes traveled mainly in a North–South direction. Interestingly most plans do not fall on the minimum curve indicating that the eight-connected effect was not the principal contributor to path length extension for most plans. For paths not on the minimum eight-connected curve, obstacle avoidance, energy minimization and constraint satisfaction contributed to path length extension, often significantly.

4.2. Large-scale terrain avoidance

TEMPEST demonstrated large-scale hazard avoidance on several occasions. The planning for April 25 shows subtlety. Fig. 5(a) shows the sequence of plans and executed paths for the day. At first glance, the initial northeast heading taken by the plans is mysterious. Why did the planner detour rather than a more direct route to the goal? The answer appears to lie in slope avoidance. By plotting the same path over a contour map of the magnitude-of-gradient (slope) field (Fig. 5(b)), we observe that the path avoids steeper slopes to its left, and then turns toward the goal at a break in the slope.

The most interesting, yet greatest failure in terrain avoidance occurred on April 18 (Fig. 6). In a plan to travel to the southern end of the area of operations, TEMPEST dictated a traverse near the rim of the large fault running in a primarily North–South direction to the West of base camp. According to observers near the robot on this day, waypoint goals dictated travel down precariously steep slopes on the West side of the

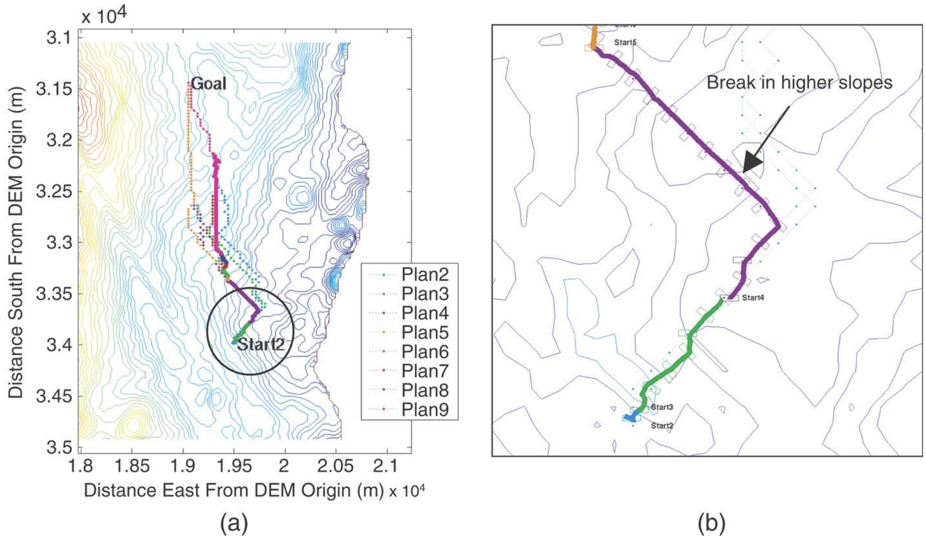

Fig. 5. (a) Plan and re-plan routes from April 25 on an elevation contour map; and (b) a close-up with contours of constant slope. The initial plans seem to have sought a break in steeper slopes.

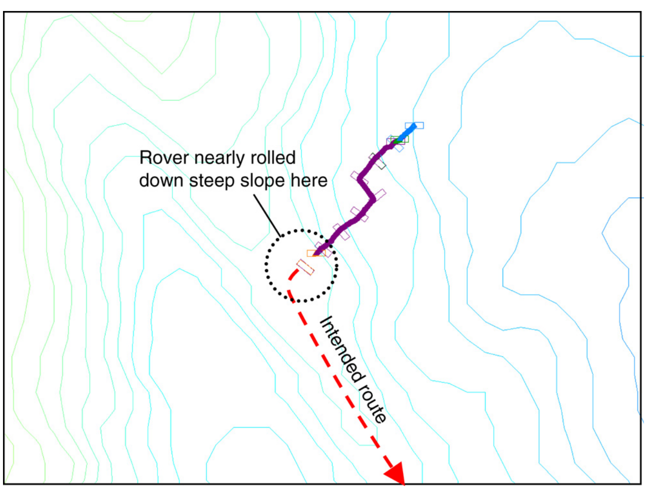

Fig. 6. A failure in terrain hazard avoidance was caused by map registration errors.

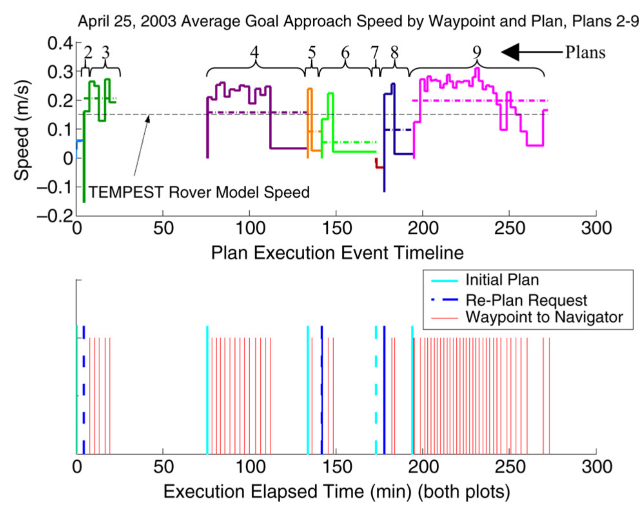

Fig. 7. Rover average speed (top) and planning events (bottom) versus time.

fault ridge. This motion prompted the team to abort autonomous travel at this point. Data supports that the map, created using photogrammetry on satellite imagery, was incorrectly registered with the terrain by several tens of meters.

4.3. Energy efficiency

In contrast to previous planning experiments in the Arctic [13], evaluating LITA TEMPEST plan energy efficiency was difficult. For Atacama experiments, Hyperion's solar array was horizontal which decouples the direction of travel and solar power, allowing the rover to fall behind schedule with little penalty until late in the day. Furthermore, the solar flux in the Atacama was sufficiently high in April, during daylight, to sustain the highest-power operations indefinitely. Shadows only occurred very near sunset, so only intersected paths when operations were coming to a close. The planning models verify this—mid-day TEMPEST plans never included Charge or Hibernation actions, and ambitious late-day plans were often infeasible on the basis on insufficient energy.

4.4. Plan monitoring and re-planning

A primary goal of the field experiment was to test re-planning in the context of rover operations and plan stability. As mentioned earlier, TEMPEST called upon ISE state update re-planning (updating a plan in response to a new initial state). The Health Monitor provided simple execution monitoring, and was the trigger for re-planning.

As position estimation was quite accurate, the major cause of re-plan requests was deviation of average rover speed from the rover model, shown in Fig. 7 for plans executed on April 25. The figure illustrates the connection between rover speed and re-planning triggers. The top plot in Fig. 7 plots speed as a function of time. The TEMPEST rover model speed is the constant dashed black line. The solid traces show the average rover speed for each executed Drive action. The dash–dot traces

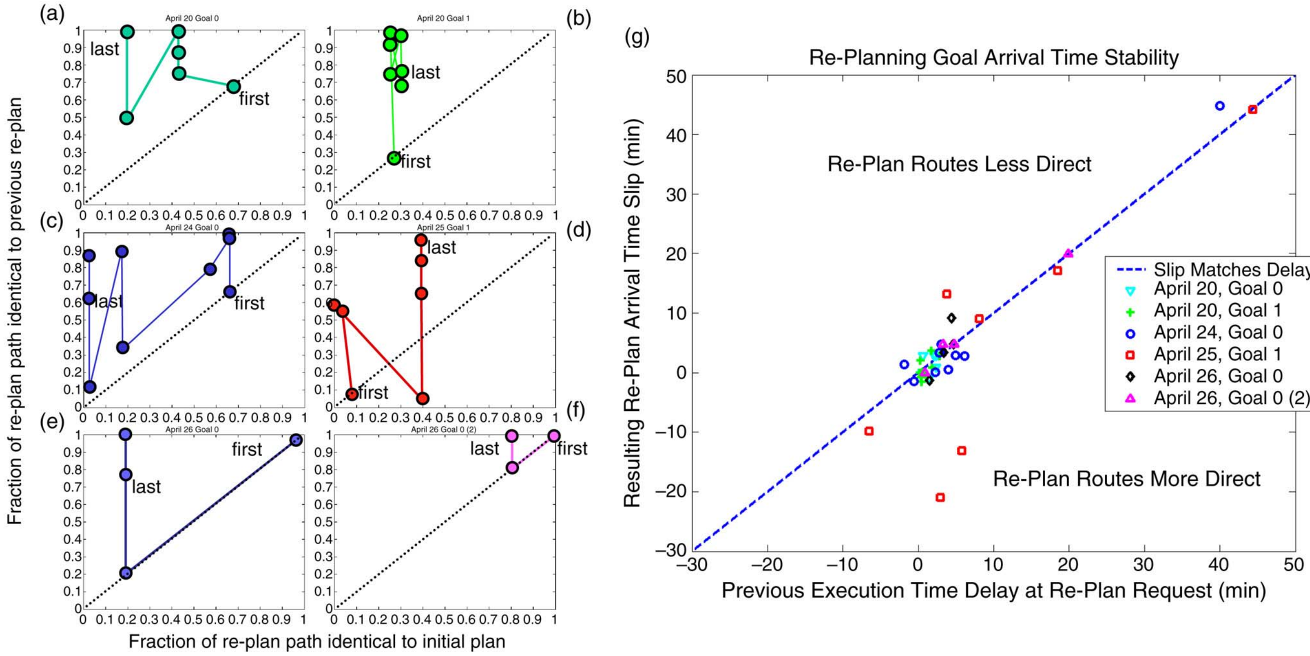

Fig. 8. Route stability (a)–(f) and arrival time stability (g) for re-planning.

show the average rover speed over the entire plan execution. Meanwhile, the bottom plot in Fig. 7 shows the timing of important plan execution events. TEMPEST was terminated several times during the day after long operational delays (after Plan 3, Plan 4, Plan 6 and Plan 8). However, it is clear from Plan 2, Plan 5 and Plan 7 that re-plans correlate well with periods of slow average driving speeds. The figure also underlines a logic error in the Health Monitor that overlooked faster-than-expected rover speed for re-planning. In no case does faster-than-predicted rover speed trigger a re-plan (see Plan 9).

4.5. Plan stability

We determined the degree to which re-planning was stable, both in terms of route selection and goal arrival time. The degree to which routes are stable in successive re-plans determines whether scientists can rely upon visiting specific interesting areas en route to the final goal. If re-plans vary significantly with respect to schedule, it may be impractical to project a detailed command schedule for the entire plan.

Fig. 8 illustrates route stability for the field experiment. For a given re-planning sequence, the horizontal axis shows the fraction of the re-plan waypoints that are identical to the initial plan (F_i). The vertical axis shows the fraction of the re-plan waypoints that are identical to those from the previous plan (F_p). Each sub-plot represents a series of re-plans to a single goal. The markers on the traces correspond to results from a specific re-planning instance. The markers labeled "first" indicate the first re-plan; subsequent points indicate later re-plans in chronological order. It follows that all traces begin on the line $F_i = F_p$ (shown as a diagonal line), since for the first re-plan, the *previous* plan is also the *initial* plan. We observe that for traces (a), (b), (c), (e) and (f), the endpoint falls generally left and above the starting point. One can infer that for these cases, re-plans are initially unstable but grow gradually more stable as plan execution progresses. This makes intuitive sense—a greater distance between start and goal entails greater freedom of choice. ISE finds a number of plans of similar cost but with differing routes and switches between them in re-plans. Subtle changes in initial conditions can cause substantial route variations. As the distance to the goal shrinks, the freedom is reduced, causing greater stability.

The exception is the plan sequence (d) from April 25, whose first re-plan shares fewer than 10% of the initial route's waypoints. Successive re-plans deviate even more from the initial plan at first, but then return to match about 40% of the remaining plan. Fig. 5(a) may help clarify what is happening in this case. The plans alternate between a right and left route over the last half of the traverse. Plan 2 (the initial plan shown) takes the right fork, Plan 3 (the first re-plan) the left. In the first half of the route, Plan 2 and Plan 3 are almost entirely distinct, but run very close to each other. In later planning instances, the plans settle on a variation of the right fork, increasing the fraction of the plan that is identical to Plan 2.

We also investigate plan arrival time stability. We wanted to determine whether projected goal arrival time slips were explained well by the delays induced by lower-than-anticipated rover speeds. In Fig. 8(g), we plot arrival time slip against operational delays. Each marker corresponds to a different plan instance. The dashed line falls where schedule slip would exactly match rover delays. Re-plans falling above the dashed line are less direct than their predecessors, while re-plans below the line are more direct. Aside from a few outliers, the data suggests a one-to-one mapping between operational delays and schedule slips. We must note that all experiments were carried out well within daylight hours. Operational delays that force a plan to conclude in vastly different power or lighting conditions, for example nighttime, might cause far greater slips in goal arrival time. For example, a robot that cannot drive at night would be forced to loiter until sunrise to resume progress toward an unattained goal.

5. Conclusions

We believe the greatest contribution of this work to be the operational proof-of-concept for online mission-level path planning. The two-level navigation hierarchy allows TEMPEST to work at an abstract level, at large scale, without knowledge of rover-scale terrain features, and allows the Navigator to seek TEMPEST waypoints and avoid rocks without knowledge of the global terrain or mission. State update re-planning enabled TEMPEST to re-synchronize plans naturally and efficiently in response to delays.

Rover plan execution could benefit immediately by incorporating sensor measurements of terrain into TEMPEST planning. Terrain avoidance failures might have been sidestepped with improved ability to perceive, identify and model mid-scale features such as extended slopes and rocky outcrops, and to plan new routes around them. This could be addressed. Dynamically-updated terrain maps would enable TEMPEST to respond to map registration problems using model update re-planning.

The Atacama's high solar flux and the insensitivity of Hyperion solar power to orientation might point to de-emphasizing energy optimization for future years. However, future experiments will entail multi-day continuous autonomous operations. Dawn, dusk and nighttime activities may force extended battery charging and low-power hibernation. From the perspective of planning it will be important to add a battery charge estimator and to develop a strategy that considers battery charge when executing plans.

Evidence suggests the rover should be more robust to sources of model error and uncertainty. Rover model speed errors prevent accurate projections of schedule, which in turn reduces the accuracy of energy projections. It is not clear TEMPEST will be able to adequately compensate for the full range of possible operational delays seen thus far. More realistically, TEMPEST might be able to plan for the speed uncertainty caused by unknown local obstacle density. Further, machine learning might be used to adapt the TEMPEST rover model to changing conditions.

In LITA 2003, science activities were totally decoupled from TEMPEST planning. Science actions were interleaved manually with plans, but TEMPEST could not predict their impact on schedule or energy resources ahead of time. Work has begun to consider science activities both in planning and execution.

References

[1] A. Jonsson, J. Frank, A framework for dynamic constraint reasoning using procedural constraints, in: Proceedings of the European Artificial Intelligence Conference, 2000.

[2] M. Kobilarov, G. Sukhatme, Time optimal path planning on outdoor terrains for mobile robots under dynamic constraints, in: 2004 International IEEE/RSJ Conference on Robotics and Autonomous Systems, IROS '04, Sendai, Japan, 2004.

[3] S. Lacroix, A. Mallet, D. Bonnafous, G. Bouzil, S. Fleury, M. Herrb, R. Chatila, Autonomous rover navigation on unknown terrains: functions and integration, International Journal of Robotics Research 21 (10–11) (2002) 917–942.

[4] S. Laubach, J. Burdick, Roverbug: an autonomous path-planner for planetary microrovers, in: Sixth International Symposium on Experimental Robotics, ISER 99, Sydney, Australia, March 1999.

[5] S. Moorehead, R. Simmons, D. Apostolopoulos, W. Whittaker, Autonomous navigation field results of a planetary analog robot in antarctica, in: 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS 99, June 1999.

[6] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading, MA, 1984.

[7] H. Seraji, A. Howard, Behavior-based navigation on challenging terrain: a fuzzy logic approach, IEEE Transactions on Robotics and Automation 18 (3) (2002) 308–321.

[8] K. Shillcut, Solar based navigation for robotic explorers, Ph.D. Thesis, Carnegie Mellon University Robotics Institute Technical Report CMU-RI TR-00-25, October 2000.

[9] A. Stentz, The focused D∗ algorithm for real-time replanning, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI '95, Montreal, Canada, August 1995.

[10] A. Stentz, M. Hebert, A complete navigation system for goal acquisition in unknown environments, Autonomous Robots 2 (2) (1995).

[11] P. Tompkins, Mission-directed path planning for planetary rover exploration, Ph.D. Thesis, Carnegie Mellon University Robotics Institute, December 2004.

[12] Z. Sun, J. Reif, On energy-minimizing paths on terrains for a mobile robot, in: 2003 IEEE International Conference on Robotics and Automation, ICRA '03, Taipei, Taiwan, 2003.

[13] P. Tompkins, A. Stentz, W. Whittaker, Mission planning for the sun-synchronous navigation field experiment, in: 2002 IEEE International Conference on Robotics and Automation, ICRA '02, Washington D.C., May 2002.

[14] P. Tompkins, A. Stentz, W. Whittaker, Mission-level planning for rover exploration, in: 8th Conference on Intelligent Autonomous Systems, IAS 04, Amsterdam, Netherlands, March 2004.

[15] E. Tunstel, T. Huntsberger, H. Aghazarian, P. Backes, E. Baumgartner, Y. Cheng, M. Garrett, B. Kennedy, C. Leger, L. Magnone, J. Norris, M. Powell, A. Trebi-Ollennu, P. Schenker, FIDO rover field trials as rehearsal for the NASA 2003 mars exploration rovers mission, in: 9th International Symposium on Robotics & Applications, 5th World Automation Congress, Orlando, FL, 2002.

[16] D. Wettergreen, B. Dias, B. Shamah, J. Teza, P. Tompkins, C. Urmson, M. Wagner, W. Whittaker, Experiments in sun-synchronous navigation, in: IEEE International Conference on Robotics and Automation, ICRA'02, Washington D.C., May 2002.

[17] D. Wettergreen, First experiments in the robotic investigation of life in the Atacama desert of Chile, in: IEEE International Conference on Robotics and Automation, Barcelona, May 2005.

Paul Tompkins is a Research Scientist with QSS Group Inc., a contractor to NASA Ames Research Center. His research focuses on planetary rover temporal and resource-based navigation planning. Mr. Tompkins was a member of the Life in the Atacama research team, and is the lead designer of TEMPEST. He received his B.S. in Aeronautics and Astronautics from MIT in 1992, an M.S. in Mechanical Engineering from Stanford University in 1997, and defended his doctoral thesis in Robotics from Carnegie Mellon in December 2004. Between 1992 and 1998 he worked at Hughes Space and Communications Company as a mission analyst and orbital operations team leader for commercial satellite programs. He is a recipient of the Hughes Electronics Graduate Fellowship, the Pennsylvania Space Grant Fellowship and was a NASA Graduate Student Research Program Fellow.

Dr. Anthony Stentz is a Research Professor at the Robotics Institute, and Associate Director of the Robotics Institute's National Robotics Engineering Consortium. He is a pioneer in the development of algorithms for incremental path search as the developer of the D∗ algorithm, which is extended in the Incremental Search Engine. Dr. Stentz' research spans unmanned ground and air vehicles, dynamic planning, multi-vehicle coordination, navigation perception, and AI for field-worthy systems. He received his Ph.D. in Computer Science from Carnegie Mellon University in 1989, his M.S. in Computer Science from CMU in 1984, and his B.S. in Physics from Xavier University in 1982. Dr. Stentz received the 1997 Alan Newell Award for Research Excellence.

Dr. David Wettergreen is an Associate Research Professor at the Robotics Institute, since 2000. He has held a faculty position at the Australian National University and an NRC Research Associateship at NASA Ames Research Center. He received his Ph.D. in Robotics from Carnegie Mellon in 1995. Dr. Wettergreen's research expertise is in field-deployable mobile robots and his work spans concept formulation through synthesis to field experimentation. He is perhaps most well known for deploying robots around the world in volcanoes, polar environments, deserts, and other locations that compel scientific investigation without human presence. Dr. Wettergreen leads the Life in the Atacama investigation, which supports this research under NASA contract NAG9-12890.