Mission-Level Path Planning for Rover Exploration

Paul TOMPKINS, Anthony STENTZ, William WHITTAKER
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract. We present TEMPEST, an autonomous, energy-cognizant planner for long-range rover navigation. Its purpose is to plan sequences of actions that avoid large-scale obstacles, balance energy expenses against available resources, and satisfy operational constraints while attaining a distant goal. An algorithm called Incremental Search Engine (ISE) enables TEMPEST to search over high-dimensional spaces, and provides highly efficient re-planning in response to changes in state transition costs. Models of the world and rover adapt ISE to the rover navigation domain. We propose the concept of Mission-Level Path Planning, then describe how TEMPEST uses ISE to solve for mission-level plans in a spatial, temporal and energy space. A simple application problem aims to illustrate TEMPEST behavior under various time and rover locomotion power conditions.

Introduction

Work in navigational autonomy for planetary exploration rovers has focused on achieving safe and efficient motion through cluttered, rocky terrain. The fact that the NASA Mars Exploration Rovers will employ such a system in 2004 is testament to the growing maturity and reliability of this technology. However, as scientists push to extend the reach of rovers far beyond the landing site to investigate multiple distant targets, issues of large-scale terrain, the passage of time, and the collection and expenditure of energy will grow in importance.

We propose the concept of Mission-Level Path Planning to signify navigational autonomy that considers long-range effects of actions - for example planning paths that avoid distant hills or canyons, or scheduling traverses, battery charging and periods of hibernation to be most efficient under limited resources. Where local navigation considers spans of meters and minutes at high resolution, mission-level navigation considers hundreds of meters and hours at low resolution. It also bridges the gap between path planning and classical AI planning and scheduling, to address coupled problems affecting or affected by route, timing and energy availability, and to provide a framework and grounding from which to derive more detailed plans.

Towards this end, we have developed an experimental planner called TEMPEST. TEMPEST was originally developed for a specific problem in long-distance, energy-cognizant navigation called Sun-Synchronous Navigation [21][18]. Since then, our focus has been to develop a more general approach for use in wider variety of problems, most recently in the support of robotic search for life in the Life in the Atacama project [22].

We begin by describing how we produce long-range rover navigation plans. The goal of the paper is to illustrate the behavior of TEMPEST in a simple problem. Having introduced our approach, we compare and contrast a number of other approaches. We conclude by providing some high-level observations from results, and by setting a course for development. A companion paper in this conference details results from field experiments in which TEMPEST acted as an online mission-level planner and re-planner in Mars-analog terrain [19].

1. Approach

1.1. Incremental Search Engine (ISE)

ISE is the search algorithm that allows TEMPEST to reason about rover actions, achieving goals efficiently, and satisfying constraints. ISE is a graph-theory based, heuristic search algorithm optimized for planning and re-planning in high-dimensional spaces [17]. The algorithm is complete, and optimal to the resolution of the discrete state space and actions.

An ISE state space comprises two types of variables. IPARMS are independent parameters, whose values are discrete and generally coarse. DPARMS are arbitrarily fine dependent parameters, whose values are grouped according to equivalence classes. Actions define transitions between IPARMS values. For each action *a*, a user-defined state transition function defines how DPARMS change in response to changes in IPARMS. For example, an application might define two spatial IPARMS variables *(X, Y)*, and a DPARMS variable for time *(T)*. The state transition function would take the form $(\Delta X, \Delta Y) = f(a)$, and where $\Delta T = f(\Delta X, \Delta Y)$. An objective function accumulates costs for a sequence of actions, where costs are functions of IPARMS, DPARMS or other parameters. ISE derives paths that are of minimum cost according to the objective function.

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 actions in reverse from the state. It maintains constraints by preventing action-state combinations that violate them. Each resulting initial state becomes a new node in a directed graph. Each node is a state from which one of the goals is reachable. Eventually, branches of the graph may intersect the start state IPARMS. Some of these are judged "feasible" according to user-supplied DPARMS criteria. From these, ISE determines the optimal path - the least expensive path according to a user-supplied objective function.

ISE results are identical to those from A* [9] in an initial plan search with no constraints. However, once environment models have changed locally, ISE operates far more efficiently than A* in re-planning paths. This is a major benefit for TEMPEST, since environment models might evolve as the rover gains new information through sensing. 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 path within it. The algorithm is time efficient because it determines which portions of the search space are affected by new information and limits the re-computation to those portions. The algorithm is space efficient through the use of three mechanisms:

- <u>Dynamic State Generation:</u> ISE creates a state when it is needed and deletes it when it no longer serves a purpose. This feature precludes the need to allocate an entire multi-dimensional space even though only a small part of it may be searched.
- <u>State Dominance:</u> ISE determines when one state dominates another, through user-defined routines, and prunes the dominated state to minimize unnecessary state proliferation.
- <u>Resolution Pruning:</u> ISE reasons about DPARMS variable resolution and prunes the lesser states from a DPARMS resolution-equivalent class. This feature can dramatically reduce the number of states while still preserving resolution optimality.

ISE enables two modes:

- <u>BESTPCOST</u> 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.
- <u>BESTDPARMS</u> finds the "best" state solution below a maximum path cost. In this mode, the objective function serves only to measure path costs against the maximum. The user-defined "better" criteria evaluate DPARMS to prioritize plans that are equivalent in path cost to determine the optimum or "best".

ISE is a general-purpose, discrete space path search algorithm. To apply ISE to a specific problem, it requires a user to 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. The following sections describe how TEMPEST defines these domain elements for long-range navigation.

1.2. Domain Models

Table 1 summarizes 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.

Table 1: World and rover models used to define the mission-level navigation planning domain.

Domain Model	Description
Terrain	• Digital elevation model (DEM) • Slope derived from DEM • Map projection equations and ellipsoidal geodetic reference enable transformations between maps and Cartesian or polar planetographic coordinates.
Ephemeris	• CSPICE software [1] predicts the relative positions and orientations of Solar System bodies.
Lighting	• Offline ray-tracing projects light from the ephemeris-predicted sun position onto the terrain model. • Lighting maps are instantaneous sun angle-of-incidence on the local slopes. • Sequences of lighting maps, at even time intervals, represent time-varying light exposure.
Solar flux	• Peak estimate of solar flux F_{peak}, in W/m^2. $F_s = F_{peak} A \cos\theta$, where A is the flat area and θ is the angle between the sun direction and the surface normal.
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-hr)
Electronics	• Power load as constant value
Navigation Cameras	• Orientation and field-of-view for sun-in-camera constraint calculations.

1.3. State Space

The TEMPEST state space comprises four dimensions - two spatial dimensions, time and battery energy. The two DEM grid coordinates specify two position IPARMS variables in ISE, *X* and *Y*. The CSPICE basis time system, in integer seconds, specifies a third, DPARMS variable *T*.

The fourth dimension, battery energy *E*, is more complicated, both in its semantics and how it is represented in ISE. It is important to clarify that *E* does not represent the instantaneous energy in the battery, but rather the minimum battery energy required to reach the goal. Because ISE searches backwards, from goal to start, it requires a user-specified end-point goal state (x_g, y_g, t_g, e_g) from which to begin its search. Since there is no operational penalty in arriving at the goal with battery energy greater than e_g, e_g represents the minimum goal arrival energy. With this in mind, let us address how to interpret the effects of positive-energy actions and negative-energy actions.

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

Alternatively, actions with a net negative energy in the forward-time direction (*e.g.* nighttime locomotion) will tend to 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 1.6.

Path solutions from ISE are trajectories through this state space. Discrete points in the plans, “waypoints”, are 4-tuples of (x, y, t, e).

1.4. Actions, Constraints and the State Transition Function

TEMPEST specifies a list of basic, motion- and energy-relevant actions that coarsely describe the essential activities of rover navigation. Table 2 summarizes the actions used for recent tests, and their target changes in IPARMS and DPARMS. Stationary actions have fixed durations, but can be executed in series to provide longer actions.

Table 2: Actions used to coarsely describe activities relevant to rover navigation

Action	$\Delta X,\Delta Y$	ΔT	Active Subsystems and Description
Drive	$0/\pm1$, $0/\pm1$	$=f(\Delta X,\Delta Y)$	• Locomotor, Solar Array, Battery, Electronics, Nav. Cameras • Separate action to each of 8 adjacent grid cells
Charge	0,0	ΔT_c=900 sec	• Solar Array, Battery, Electronics • Rover heading dictated to optimal sun angle
Hibernate	0,0	ΔT_c=1800 sec	• Solar Array, Battery, Electronics • Low-power option for extended low solar energy conditions

A user can also specify action-specific constraints that specify state conditions under which an action cannot be executed. Table 3 lists some of these local constraints and how they might be applied to actions.

Table 3: Local constraints used to enforce the operational limitations of actions

Constraint	Description	Applied to
Slope	Prevents actions in areas that are too steep.	Drive, Charge, Hibernate
Daylight	Prevents actions at night.	Drive, Charge
Direct Sun	Prevents actions in shadows or at night.	Charge
Sun-In-Camera	Prevents actions if the sun is in the field-of-view of the navigation cameras.	Drive

Given an action and initial state, the TEMPEST state transition function uses a path integrator that calls on the world and rover models to compute the final state.

1.5. Start and Goal Specification

ISE must know the goal states to know where to begin its backwards-chaining search. For these experiments, TEMPEST specifies goal positions and battery energies, but leaves arrival time unconstrained. TEMPEST accomplishes this by utilizing the ISE multiple goal mechanism. TEMPEST estimates a likely range of arrival times - a "goal window" - based on best-case and worst-case projections on path duration. At even intervals within the goal window, TEMPEST designates separate ISE goal states. All have the same position and energy, but each has a different time value. Since the ISE search graph does not distinguish between paths growing from different goals, path solutions are free to terminate at any 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 grid cell center as the IPARMS position start (x_s, y_s). Since the search occurs on the coarse IPARMS discrete grid, it is reasonable to designate a specific point. However, time is an arbitrarily fine DPARMS variable in the search. A search is not likely find any paths that precisely match both the start position and a specific time, so TEMPEST designates an allowable "start window" that spans the current time $(t_{si} < t < t_{sf})$. Finally, TEMPEST treats the current battery energy e_s as upper bound on E for the start state.

1.6. Feasibility and Optimality

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

$$P = \{w_1, \ldots, w_n\} \text{ with } \mathrm{w}_i = (x_i, y_i, t_i, e_i),\ \mathrm{t}_i < t_{i+1} \qquad (1)$$

Given a start state $s = (x_s, y_s, t_x, e_s)$ and a start window time interval $[t_{si}, t_{sf}]$, a plan is feasible if and only if:

$$(x_1 = x_s) \wedge (y_1 = y_s) \wedge (t_{si} \le t_1 \le t_{sf}) \wedge (e_1 \le e_s) \qquad (2)$$

In the Life in the Atacama project, TEMPEST planned energy-optimal paths. Unlike path duration or distance, whose quantities increase monotonically as a plan gets longer, energy is non-monotonic - locomotion and other rover activities expend energy while solar energy and other power sources restore it. Standard heuristic search approaches to path planning would become "stuck" in states providing a net positive energy intake, and would never actually reach the designated goal.

To avoid 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 in these experiments, we define the quantity E_{max}:

$$E_{max} = \left| min_{\forall(s \in S,\, a \in A,\, \Delta e \le 0)} \Delta e = f(s, a) \right| \qquad (3)$$

In words, E_{max} is the absolute value of the greatest single-step negative cost (positive increment) to battery energy over all states and actions. The value of the TEMPEST objective function is the sum of two quantities:

$$L = nE_{max} \qquad (4) \qquad\qquad B = \sum_{i=1}^{n} \Delta e_i \qquad (5)$$

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 $\Delta e + E_{max} \geq 0$, the objective function increases monotonically over a path, and hence 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 $\boldsymbol{E}$ state variable. Further, since changes in the $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{T}$ variables are totally independent of $\boldsymbol{E}$ in all TEMPEST applications to date, $\boldsymbol{E}$ 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.

The basic TEMPEST search plans from a start to a single position goal. By chaining ISE searches in series, TEMPEST also enables planning to an ordered list of position goals $G = \{g_1, \ldots, g_n\},\ g_i = (x_i, y_i)$. In the interests of space, and given that multi-goal planning was not tested in the Atacama Desert, we elect not to describe that work here.

2. Simulation Results

We now show how TEMPEST generates mission-level path-plans in a simple hill traverse problem. We synthesized a 3.6 km x 3.6 km hypothetical model representing smooth Earth terrain at 0º latitude 0º longitude at 30 m resolution, and generated lighting data for it centered on North hemisphere summer solstice. A solar-powered rover sits on the West side of the hill, and must reach a goal on the East side with 10% of full charge remaining in its battery. Its solar array is mounted horizontally, and charges a battery. The robot cannot drive at night due to low light, but can hibernate at low power if necessary. We examine how varying start times and locomotion powers affect the optimal TEMPEST plan.

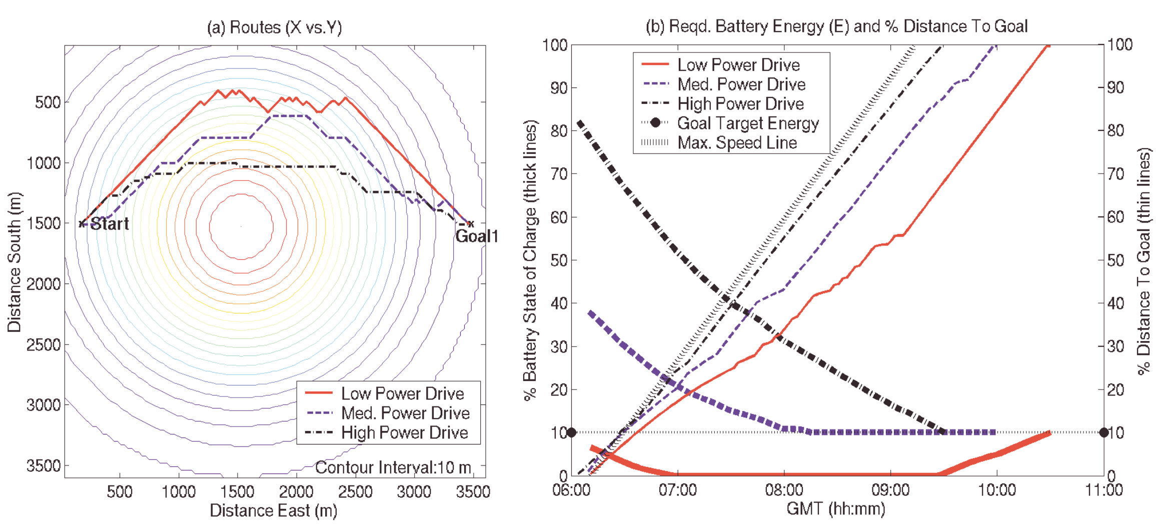

Figure 1: Early morning departure leads to benign power conditions. Rovers requiring higher locomotion power must start with more fully charged batteries to achieve the goal.

Figure 1 and Figure 2 summarize plans for two departure times assuming low, medium or high locomotion power. In each figure, the left plot depicts the hill with elevation contours, the rover start and goal positions to the left and right of the summit respectively, and plots of the *X* and *Y* waypoints. The right plot has two sets of curves: 1) distance travelled towards the goal (right axis, thin lines), and 2) battery energy *E* (left axis, thick lines). Both are plotted against local time of day. Note that the slope of the "distance travelled" lines represents the rover speed toward the goal.

Figure 1 looks at an early morning departure (immediately after sunrise) for low, medium and high locomotion power draw. One first observes that all three routes pass to the North of the hill which, given the summer season, allows greater solar array sun exposure than the southern slopes. One also notes that the cases requiring greater locomotion power take more direct routes, illustrating a tradeoff between longer sun exposure and shorter path length. In Figure 1b, we also see from the distance plots that plans dictate continuous driving (the "max speed line" shows the fastest possible route). The indirect, low-power route leads to a delay of roughly one hour relative to the direct high-power. The energy plots illustrate the behavior of the *E* state variable - the least energy-taxed case requires nearly zero battery charge for the entire traverse. Higher power cases require greater initial energy to reach the goal. Since sunlight is plentiful, this case is benign.

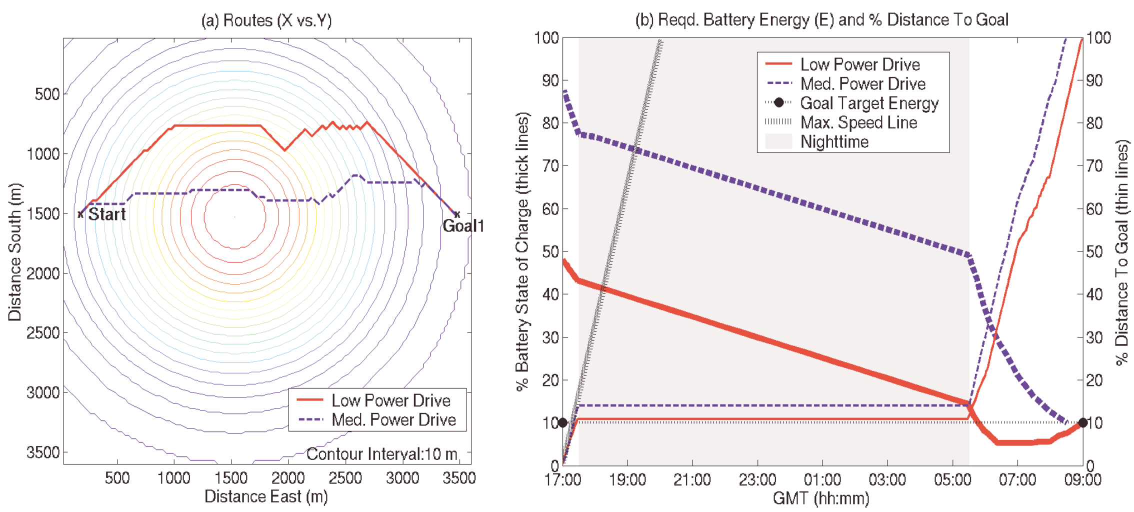

Figure 2: The late evening departure forces an overnight hibernation and battery drain. The highest-power case could not achieve the goal without overfilling the battery. Note the similarity between the battery profiles in the morning here and those in Figure 1.

Figure 2 shows corresponding cases for a late evening departure. The routes are similar to those in the first case. However, the late departure prevents the rover from reaching the goal before nightfall. At night, the rover is forced to hibernate, expending as little energy as possible until morning. The distance plot shows this long period of no motion. The energy plots indicate the batteries must be highly charged to power the robot during hibernation. Interestingly, the profiles from 06:00 to 09:00 for distance and energy are almost identical to those in the first case. As before, the more power-hungry rover requires a much greater initial battery charge. In fact, the highest-power case did not have sufficient battery capacity to complete the traverse - hence it is missing from the plots.

Planning and re-planning behave identically. In understanding how the above plans vary with time of day, one also begins to understand how schedule jumps or slips, caused by reshuffled events, unforeseen delays, or opportunistic science, might affect route, timing and resources.

As a result of the removal of the energy dimension from the search space, TEMPEST is over 100 times faster than the version used in Arctic experiments [18]. Search time and memory requirements are domain dependent. More involved solutions require significantly greater search time and effort. The above cases took just over 60 seconds to plan, with a search rate of 29 thousand states generated every second. However, cost pre-calculation is significant - TEMPEST generated 17.3 million cost vectors for each case, in an average of

15 minutes. Each cost vector requires 16 bytes, for a total of 278 Mbytes for the above cases, while the ISE graph requires 50 bytes per state searched, for 70 to 90 Mbytes to solve the first plan.

3. Related Work

For search, the D* algorithm [13][14][15] is similar to ISE in that it provides optimally efficient planning and re-planning on a graph. However, it does not enable planning under global path constraints, or planning in spaces of greater than two dimensions. The ABC algorithm [7] enables inequality and optimality constraints on a list of path costs, but lacks mechanisms for efficient re-planning. CD* [16] provides path planning under a single global constraint, and uses D* in a binary search to find a weighting factor for a single constrained variable in the objective function. However, it does only works in 2-D state spaces, and cannot track multiple global constraints.

Research into path planning for vehicles spans a wide array of thrusts, including local obstacle avoidance [4][6][10][20] and dynamics [5][12], but almost universally focuses on local route planning to avoid obstacles over short traverses. Techniques like that in [8] are promising for large-scale terrain evaluation, but avoid the temporal and energy issues we address here. The early work of Bobrow et al. [2] and others [3][12] addressed temporal path planning by first selecting a path to avoid static obstacles, then determining the acceleration profiles necessary to reach time goals or avoid moving obstacles. These approaches are limited in that they avoid the fully-coupled space-time problem. Finally, the literature is almost completely absent of work toward path planning under renewable resources. Shillcut analyzes several robot coverage patterns in terms of solar energy collection [11], but stops short of planning paths to optimize energy.

4. Conclusions

TEMPEST demonstrates a basic capability to plan routes and schedule energy-saving events in the face of complex geometry and changing conditions. The planning framework is flexible to accommodate a range of rover navigation problems, from planning the timing of polar circuits to planning linear traverses in mid-latitudes. In a current effort, we are redesigning the TEMPEST architecture to enable a much richer specification of models, actions, constraints and goals.

Specifically, TEMPEST is intended to fulfill mission-level path planning for science exploration rovers. Currently, neither science activities nor communications are represented in plans. Both are central to exploration and strongly affected by route selection, timing and energy. We plan to introduce goal-based actions to model position-specific science activities. In conjunction with current multi-goal planning, TEMPEST will be able to represent sequential science site investigations. The TEMPEST architecture is ideal for predicting communications line-of-sight geometry. The sun, solar array and battery might have communications analogs in orbiting relays, antennae and onboard memory. It is not clear how to interleave communications planning with energy planning while maintaining tractability.

Owing to the collapse of a dimension of the search space through an innovation in the ISE objective function, TEMPEST is far faster and less memory intensive than previously documented. This improvement allows the software to plan over substantially larger areas, and resulted in the first online use of TEMPEST in field experiments in Chile. We now seek improvements in state transition cost calculations and storage to compute only costs used in the search.

The range of problems TEMPEST can address is limited by basic structure and representation. Computation and memory are exponential in the number of state variables, preventing TEMPEST from addressing very high dimensional problems directly. The sequential action model prevents more generic partial-order plan solutions. In limited form, TEMPEST enumerates simultaneous actions as new actions, but clearly this approach suffers from combinatorial explosion of the action space, as well as the branching factor of the search.

This paper only addresses TEMPEST algorithms and basic behaviors. One of TEMPEST's greatest strengths is in online planning and re-planning. We invite readers to the companion paper in this proceedings [19] for more information on mission-level path planning in support of robotic astrobiology in the Atacama Desert.

References

[1] C. H. Acton Jr., "Ancillary Data Services of NASA's Navigation and Ancillary Information Facility", *Planetary and Space Science*, 44 (1):65-70, 1996.

[2] J. Bobrow, S. Dubowsky, J. Gibson, "Time-Optimal Control of Robotic Manipulators Along Specified Paths," *International Journal of Robotics Research, Vol. 4, No. 3,* Fall 1985.

[3] T. Fraichard, "Dynamic Trajectory Planning with Dynamic Constraints: a 'State-Time Space' Approach," *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 93)*, Yokohama, Japan, July 1993.

[4] S. Goldberg, M. Maimone, L. Matthies, "Stereo Vision and Rover Navigation Software for Planetary Exploration," *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[5] K. Iagnemma, H. Shibly, A. Rzepniewski, S. Dubowsky, "Planning and Control Algorithms for Enhanced Rough Terrain Rover Mobility," *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (I-SAIRAS 2001)*, St.-Hubert, Quebec, Canada, June 2001.

[6] S. Laubach, J. Burdick, "RoverBug: An Autonomous Path-Planner for Planetary Microrovers," *Sixth International Symposium on Experimental Robotics (ISER 99)*, Sydney, Australia, March 1999.

[7] B. Logan, N. Alechina, "A* with Bounded Costs," *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, 1998.

[8] D. Pai, L. Reissell, "Multiresolution Rough Terrain Motion Planning," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, February 1998.

[9] J. Pearl, <u>Heuristics: Intelligent Search Strategies for Computer Problem Solving</u>, Addison-Wesley, Reading, MA 1984.

[10] H. Seraji, A. Howard, "Behavior-Based Navigation on Challenging Terrain: A Fuzzy Logic Approach," *IEEE Transactions on Robotics and Automation*, 18(3), pp. 308-321, June 2002.

[11] 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.

[12] Z. Shiller, Y. Gwo, "Dynamic Motion Planning of Autonomous Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, April 1991.

[13] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning", *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, August 1995.

[14] A. Stentz, M. Hebert, "A Complete Navigation System for Goal Acquisition in Unknown Environments", *Autonomous Robots*, Vol. 2, No. 2, August 1995.

[15] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments", *International Journal of Robotics and Automation*, Vol. 10, No. 3, 1995.

[16] A. Stentz, "CD*: A Real-time Resolution Optimal Re-planner for Globally Constrained Problems," *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, Edmonton, AB, Canada, July 2002.

[17] A. Stentz, "Optimal Incremental Search for High-Dimensional, Constrained Path Finding Problems," Carnegie Mellon University Robotics Institute Technical Report, to be released.

[18] P. Tompkins, A. Stentz, W. Whittaker, "Mission Planning for the Sun-Synchronous Navigation Field Experiment," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA-02)*, Washington D.C., May 2002.

[19] P. Tompkins, A. Stentz, W. Whittaker, "Field Experiments in Mission-Level Path Execution and Re-Planning," *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS 2004)*, Amsterdam, Netherlands, March 2004.

[20] C. Urmson, M.B. Dias, R. Simmons, "Stereo Vision Based Navigation for Sun-Synchronous Exploration," *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 02)*, Lausanne, Switzerland, October 2002.

[21] D. Wettergreen, B. Dias, B. Shamah, J. Teza, P. Tompkins, C. Urmson, M. Wagner, W. Whittaker, "Experiments in Sun-Synchronous Navigation," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 02)*, Washington D.C., May 2002.

[22] D. Wettergreen, N. Cabrol, F. Calderon, M. Deans, D. Jonak, A. Luders, F. Shaw, T. Smith, D. Strelow, J. Teza, P. Tompkins, C. Urmson, V. Verma, A. Waggoner, M. Wagner, "Life in the Atacama: Field Season 2003 Experiment Plans and Technical Results," Carnegie Mellon University Robotics Institute Technical Report, CMU-RI-TR-03-50, October, 2003.