

Mission Planning for the Sun-Synchronous Navigation Field Experiment

Paul Tompkins Anthony Stentz William “Red” Whittaker

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes TEMPEST, a planner that enables a solar-powered rover to reason about path selection and event placement in terms of available solar energy and anticipated power draw. Unlike previous path planners, TEMPEST solves the coupled path, path timing and resource management problem. It combines information about mission objectives, operational constraints, the planetary environment and rover performance, and employs the Incremental Search Engine (ISE), a new search algorithm that produces optimal paths through high-dimensional spaces. In July 2001, TEMPEST supported the Sun-Synchronous Navigation Field Experiment on Devon Island in the Canadian Arctic. The planner successfully selected time-sequenced, closed-circuit paths that enabled a solar-powered planetary rover prototype to traverse a multi-kilometer path over 24 hours with battery energy reserve. The field trial results motivate future work in mission re-planning, multiple resource constraint analysis and improved speed and memory performance. Our objective is to fulfill a need for resource-cognizant autonomy that is critical for future long-distance planetary surface missions.

1 Introduction

Future Mars rover exploration missions grow increasingly ambitious in terms of science data diversity and coverage, requiring greater rover mobility and autonomy [5]. As reach is extended further beyond the landing site, a new host of issues become pertinent to the path planning problem. Regional terrain may inhibit travel to distant science targets. Sunlight availability changes with the presence or absence of occluding hills or mountains, prompting careful planning of the timing of solar battery charging, driving, communications and science. Even the power output of candidate rover nuclear generators may not sustain continuous driving over long distances, forcing well-timed, periodic battery charging during inactive time periods. In either case, extending rover resource-cognizant autonomy might significantly improve the science data return for planetary surface missions.

The Sun-Synchronous Navigation Project sought to investigate several technologies to enable long-duration, solar-powered exploration of the poles of planets and moons of the inner solar system [13]. TEMPEST (TEmporal Mis-

sion Planner for the Exploration of Shadowed Terrain) was a key development of this project: a step towards greater levels of autonomy through resource-cognizant path planning. The software produces plans whose path and timing are synchronous with planetary rotation, and hence with the sun. Though applied to the sun-synchronous navigation strategy, the mission planning framework is applicable to general resource management on planetary exploration robots.



Figure 1: The Hyperion rover, for which TEMPEST generated sun-synchronous routes. Note the vertical solar array and long polar shadow

Sun-synchronous navigation is an energy-cognizant strategy for planetary exploration in polar regions [12],[13]. Sun-synchronous routes, as the name implies, are synchronized with the motion of the sun in the sky. In the summer at polar latitudes of Mercury, Earth, the Moon and Mars, the sun remains above the horizon, at low elevation angles, for months. A rover could maintain sun exposure on a vertically-mounted solar array (see Figure 1) simply by driving in a circular path and turning at a rate that matches the motion of the sun. This strategy achieves sun pointing without the mechanical complexity and mass of a motor-driven array. Given proper solar array sizing, such routes guarantee a greater battery energy level at the end of the cycle than at the start. Daily recharge permits repetition, and with a sufficient path radius, enables coverage of large areas.

Circular sun-synchronous paths are idealized. In reality, terrain may interfere with driving and cast shadows, science targets may not lie on a circular path, and mission-critical activities or unanticipated delays may force a rover to deviate from a nominal schedule. Sun-synchronous route planning must weigh potential paths and timing based on energy availability and expected energy costs, balancing the benefits of solar panel sun exposure and minimum distance path with difficult terrain crossings and entry into shadows. Using sun-synchronous strategies, current robots might be able fully explore a small region, traversing the area once for each day of a summer season. With greater terrain-crossing and speed capabilities, robots might one day repeatedly circumnavigate the pole of a slowly rotating planet or moon (e.g. Earth's moon or Mercury¹), selecting the path and speed to enable persistent solar power and to follow benign thermal conditions.

2 Relevant Work

Substantial research and development efforts have yielded path planners that enable autonomous rover travel in natural terrain. One example is RoverBug [4], which utilizes local tangent graphs to construct minimum-distance paths about obstacles detected by rover sensors with limited range and fields-of-view. Using a dual strategy of "motion-to-goal" and "boundary following", the algorithm has successfully demonstrated path planning and execution aboard the JPL Rocky 7 Mars rover prototype. A CMU path planner [7], also produced with Mars rover navigation in mind, calls upon the D* algorithm [8],[9],[10]. Using a grid-based approach, D* uses sensor information to populate cells with traversability data, and plans paths that avoid hazards and that are distance-optimal under current world knowledge. This scheme has been successfully demonstrated on a CMU ATRV robot [7], and in evolved form on the CMU Hyperion rover [13]. Each path planner minimizes distance and avoids obstacles in producing paths, but ignores other potential factors (e.g. slope, sun position, shadow) and associated costs and benefits over the traverse. Both path planners are designed to permit autonomous navigation over distances on the scale of 100 m, in accordance with near-future Mars rover requirements. Neither addresses the coupled path planning and resource management problem anticipated during autonomous travel on the scale of 10's of km in partially shadowed, varied terrain.

In the arena of grid search applied to path planning, the D* (Dynamic A*) algorithm [8],[9],[10] was developed to balance the look-ahead capability of deliberative planning with the rapid response of reactive behavior. Like A*, D* operates on a map of cost values and finds the lowest cost path from the start to the goal. Employed on a vehicle in unknown terrain, D* enables incremental changes to ini-

tial plans as new information is collected. As cost values in the map are modified, D* computes a new optimal path from the vehicle's current location to the goal. D* uses incremental graph theory techniques to continually "repair" the path and efficiently produce a new optimal path to the goal, based on all information learned and aggregated to that point. For large maps, D* is hundreds of times faster than re-planning from scratch using A*, yet it produces the same results. However, as formulated, D* does not have mechanisms for dealing with the proliferation of search states that typically arise from high-dimensional, constrained path finding problems. For example, in addition to minimizing the length of a traverse, it may be important to constrain the feasible paths by terrain difficulty, driving time, energy expended, risk of accident, sensor coverage of areas of interest, and time spent out of communication. Each constraint increases the size and dimensionality of the search space and thus the time and memory required to find a path. Without careful management of the search, even small problems can become intractable.

Automated activity planning and scheduling software has been successfully deployed on spacecraft and prototype planetary rovers, including Remote Agent [2] and ASPEN [3]. In particular, the ASPEN system and the derivative CASPER system were separately integrated onto the JPL Rocky 7 rover and used to produce coordinated activity schedules based on science and engineering team requests. The activity planners enabled plan repair and reformulation in response to changing goals and other unexpected events. Rover activity scheduling experiments considered resources and environment effects (e.g. day/night cycle, sun angle), but demonstrated only loose coupling to path planning, focusing primarily on conflict resolution through event rescheduling or reordering.

Shillcut [6] analyzes how various rover terrain coverage patterns affect available solar power, and provides motivation for planning under such considerations.

3 Method

TEMPEST solves for mission plans that combine time-sequenced path plans, minimum required battery energy guidelines, and a schedule for battery recharge periods. The planner runs offline prior to rover operations, but represents the deliberative layer in the Hyperion rover software architecture (Figure 2). At this early stage of research, resulting plans are sent to a user interface and transmitted to the rover, action-by-action over the course of a traverse. A local navigation system guides the rover between waypoints in the plan, which are separated by 25 to 100 meters [13]. Future developments will fully integrate the planner onboard a rover.

TEMPEST plans paths in two stages. It plans path segments between pre-designated subgoals using the Incremental Search Engine (ISE), which provides efficient

1. The moon's and Mercury's solar days are 29.5 and 175.9 days respectively, enabling slow-speed sun-synchronous driving at polar latitudes

search through high-dimensional state spaces, manages constraints on the feasible set of solutions, and enables quick re-planning [11]. Modules specifying mission objectives, operational constraints, planetary environment and rover adapt domain specifics into the ISE framework. Resulting path segments are chained into full paths and evaluated on their global merit of minimal energy requirements and timeliness with respect to the mission objectives. The following sections describe TEMPEST planning in detail.

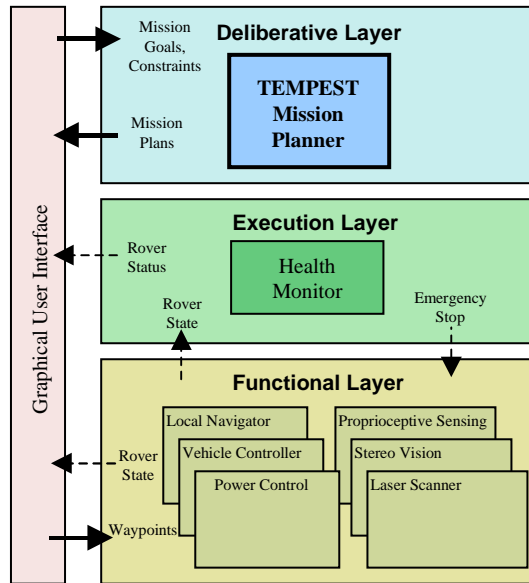


Figure 2: TEMPEST's role in the Hyperion architecture. The planner was run offline, and resulting plans fed to the rover through the GUI

3.1 Mission Domain Formulation

TEMPEST defines the mission domain in terms of the planetary environment, rover state and actions, rover performance, mission objectives and operational constraints:

Terrain: Digital Elevation Models (DEM) encode elevation values to grid cells. The planner designates paths at the spatial resolution of the DEM (cell size). DEM's used during the Houghton Field Experiment were of 25 m resolution. TEMPEST maps the elevation data to one of many available geodetic datum ellipsoids. Using the elevation data, the planner estimates local slopes, used to calculate rover locomotion energy costs and rover orientation.

Planetary Ephemeris: TEMPEST uses the JPL CSPICE software [1] to determine the relative position and orientation of Solar System bodies. For the Sun-Synchronous Navigation Project, CSPICE was used to determine sun azimuth and elevation from the local horizontal.

Lighting Maps: Using terrain and planetary ephemeris data and ray-tracing calculations, TEMPEST calculates

maps of lighting exposure at given times. Each lighting map represents a snapshot in time. When collected in a sequence, they represent the dynamic lighting on the surface over time. TEMPEST considers planetary curvature and so predicts horizon rolloff effects in the maps. TEMPEST refers to a lighting map sequence over one or more daylight cycles to determine lighting and shadowing for solar power and constraint satisfaction calculations.

Solar Flux: TEMPEST uses curve-fits of predicted available power per unit area at a particular global location as a function of time. For the Arctic field experiments, the mission planner used a constant value model for solar flux, representing the mean over one day, to enable solar energy collection calculations.

Rover State and Actions: Rover state is represented by x- and y-position, time and battery energy. TEMPEST searches on this four-dimensional space for plan solutions.

In the current formulation, TEMPEST considers two actions: *Drive* and *Charge*. The *Drive* action moves the rover between adjacent cells on the 8-connected map, and allows solar energy collection while in motion. The *Charge* action points the rover in the optimal sun direction and charges the batteries for a fixed duration. Each action results in time and battery energy costs, though energy costs can be negative if incoming collected energy exceeds expended energy.

Rover Performance: A kinematic model describes the orientation of a fixed solar array relative to rover driving direction. Rover speed is assumed to be constant. Other models estimate rover locomotion power as a function of speed, slope and direction, background electronics power, solar array power as a function of array area, efficiency and orientation, and battery energy accumulation as a balance between rates of energy intake and expenditure.

Mission Objectives: TEMPEST allows human operators to specify simple mission objectives in terms of goal position and minimum goal battery level, the position and order of intermediate subgoals, the start position and battery level, and an allowable window of start times. Selecting position subgoals allows operators to promote travel through safe locations, to specify locations for intermediate science surveys, and in the case of sun-synchronous navigation, to enforce cyclical paths by placing the final subgoal at the start position. Future versions will enable an operator to define dwell times and arrival time windows for each subgoal, enabling higher-fidelity modeling for science surveys, or other activities. Subgoal lists are ordered; no attempt is made to simultaneously address the Traveling Salesman Problem.

Operational Constraints: TEMPEST imposes mandatory constraints on minimum and maximum battery charge, prevents start times outside the allowable start time window, and prohibits paths that exceed map dimensions. A

human operator can select optional constraints to impose a minimum allowable sun angle or maximum allowable slope, prevent entry into any shadowed region, or disallow paths that enter user-defined stayout zones. The planner treats constraints as obstacles in the (x, y, time, energy) search space, preventing travel through restricted regions.

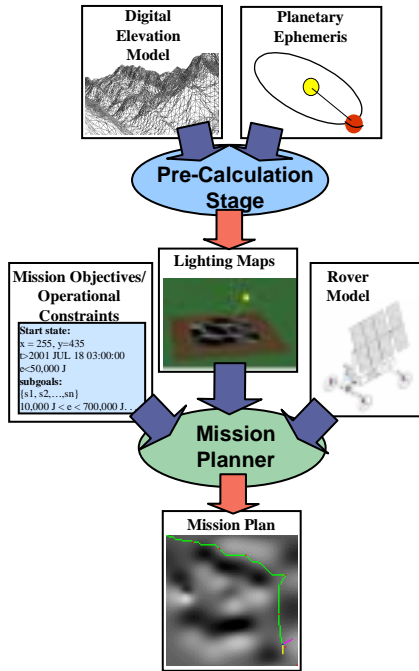


Figure 3: TEMPEST pre-calculates lighting maps, which are stored for later use. Planning combines objectives, constraints, lighting maps and the rover model to produce mission plans

3.2 Incremental Search Engine

ISE is a graph-theory based search algorithm optimized for planning and re-planning in high-dimensional spaces [11]. It allows TEMPEST to represent and reason about paths in two spatial dimensions, time and battery energy level. In initial planning, ISE performs a heuristic search to plan a specific path, given all known information about the world, that satisfies path constraints and is optimal. During this process, ISE populates the search space graph with pointers specifying the optimal path from any point to the goal. As world conditions change, for example through rover sensing in the course of execution, ISE can re-plan a new path in real time that is both feasible and optimal. For the Sun-Synchronous Navigation Field Experiment, the rapid re-planning feature was not used. Future work will focus this re-planning capability.

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 (e.g. sensor data, new goal parameters) and limits the recomputation 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 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.

State Dominance: ISE determines when one state dominates another, through user-defined routines, and prunes the dominated state to minimize unnecessary state proliferation.

Resolution Pruning: ISE reasons about parameter resolution and prunes the lesser states from a resolution-equivalent class. This feature can dramatically reduce the number of states while still preserving resolution optimality.

ISE provides functionality in goal and start state specification that eases the task of sun-synchronous navigation planning. First, ISE enables planning to more than one goal state from a common start state. If more than one goal state is specified, ISE will return the optimal path to the least expensive goal, as defined by the objective function. Secondly, ISE can efficiently plan paths to a new start state in subsequent calls by performing only those calculations necessary to extend the search to the new region of interest.

ISE operates in one of two search modes: 1) Find minimum cost path, and 2) Find best state solution below a specified maximum path cost. TEMPEST uses the second option, as described in the following section.

3.3 Planning

TEMPEST plans paths between the start and goal, through all the subgoals. Early experiments showed it was not practical to globally optimize entire paths through the enormous four-dimensional spaces in a single search (typically $> 5 \times 10^7$ states). However, local optimization ignores the global effects of time-varying sunlight on path energy that are so crucial to this application. TEMPEST compromises by planning many path options spread over a range of times, then selects the optimal path from the group. This process is described in detail:

Path Generation: TEMPEST plans paths in reverse order from goal to start, subdividing its search by locally optimizing path segments between adjacent subgoals. The algorithm, depicted graphically in Figure 4, proceeds according to the following steps:

1. Based on projections of possible bounds on average rover speed and path distance, and the allowable start time window specified in the mission objectives,

estimate a window of likely final subgoal arrival times. Create a list of goal states spaced evenly over the window, but co-located at the final subgoal position.

2. Using similar projections, estimate a window of likely start times for the path segments leading from the next earliest subgoal. Define equally-sized, adjacent start time intervals over the start window.
3. For each start time interval in the start window, call ISE. ISE searches for an optimal path segment between the current time interval and any of the goal states. Optimal segments are defined to have the minimum battery energy profile beneath a maximum threshold path time cost. Save path segment solutions, and abandon the goal states to which feasible solutions cannot be found.
4. If the goal position for the path segment solutions is the first subgoal (meaning the segment start position is the overall path start), record all chains of path segments as full path options. Else, replace the goal state list with a new list whose states are the start states for all path segment solutions planned in the previous round, and return to Step 2.

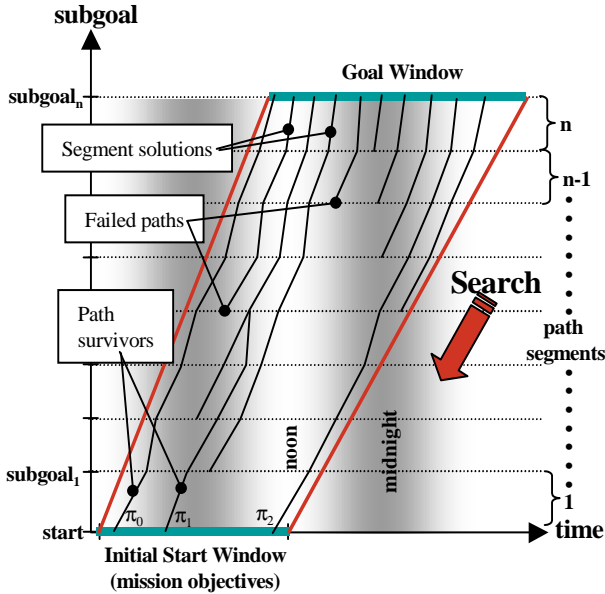


Figure 4: Depiction of backward-chained plan search. TEMPEST defines start and goal windows, and calls ISE to search for optimal path segments spread over time. By passing the start states of later legs as goals for earlier legs, TEMPEST builds complete paths for subsequent heuristic evaluation.

In this way, TEMPEST builds complete paths in reverse order, from goal to start. Path solutions are ordered lists of states, called waypoints, representing the rover state at each discrete step from the start position of the rover through all subgoals:

$$\pi \equiv \{w_0, w_1, \dots, w_{n-1}\}$$

Each waypoint is defined by four state parameters and one action parameter:

$$w_i \equiv \{x_i, y_i, t_i, e_i, a_i\} \quad \text{where}$$

- $x_i \equiv$ x-coordinate of waypoint cell in DEM
- $y_i \equiv$ y-coordinate of waypoint cell in DEM
- $t_i \equiv$ arrival time at waypoint (absolute reference)
- $e_i \equiv$ minimum allowable battery energy at waypoint
- $a_i \equiv$ principle action at waypoint, starting at t_i with
- $a_i \in \{\text{Drive, Charge}\}$

Waypoint position is that of the center of each DEM grid cell along the path. Therefore, resulting paths travel through an 8-connected grid at the scale of the DEM spatial resolution. Meanwhile, the action parameters distinguish plans as mission plans because they define events over the traverse upon which path timing and energy profile depend. TEMPEST derives a list of m surviving plans that satisfy all specified mission constraints, each corresponding to a different start interval:

$$\Pi_{\text{surv}} \equiv \{\pi_0, \pi_1, \dots, \pi_{m-1}\}$$

Path Selection: TEMPEST applies one or more of the following evaluation heuristics on path state values to prune Π_{surv} and find the best path or paths for execution. The first two heuristics return a list of all tied paths. The final heuristic returns a unique path:

Get Minimum Initial Required Energy Paths: Given the definition of energy e_i at each waypoint, a low initial energy corresponds to the least initial battery charge required to achieve the goal state.

$$\Pi_{ie}^* = \operatorname{argmin}_{\pi}(e_0)$$

Get Minimum Peak Required Energy Paths: The peak energy in a path represents the greatest demand on the battery over the traverse. Finding the minimum peak energy minimizes the peak demand on battery reserves.

$$\Pi_{pe}^* = \operatorname{argmin}_{\pi}(\max(e_i))$$

Get Earliest Start Time Path: Given a list of available start times, in many cases it is operationally sound to select the first opportunity path, leaving fallback options in the event of failure.

$$\Pi_{st}^* = \operatorname{argmin}_{\pi}(t_0)$$

TEMPEST applies these selection heuristics in series to achieve a composite selection criterion. The Sun-Synchronous Navigation Experiment used the following:

$$\Pi_{execute}^* = \Pi_{st}^*(\Pi_{pe}^*(\Pi_{ie}^*(\Pi_{surv}^*)))$$

4 Field Experiment Results

During July 2001, a CMU research team conducted the Sun-Synchronous Navigation Field Experiment on Devon Island in the Canadian Arctic [13]. Using the solar and battery powered Hyperion rover, the team oversaw two 24-hour experimental traverses to test technologies for sun-synchronous operation, including mission planning. In both cases, the objective was to operate over long distances under solar power, with minimal human intervention, and to return to the start position 24 hours later with higher battery charge than at the beginning. Tests indicated Hyperion's batteries would provide roughly two hours of continuous locomotion in the absence of solar power. This paper details results from the first experiment. Table 1 summarizes the experiment as it applied to the planner.

Table 1: Summary of Experiment 1 Plan and Execution

Quantity	Planned	Executed
# of subgoals	19 (last subgoal pertained to start pos.)	
# of waypoints	197 (183 Drive/14 Charge)	
Distance (km)	5.6	6.1
Duration (hr:mn)	23:45	23:45
% within 10 deg of optimal pointing	32.5	31.3

The five frames of Figure 5 depict snapshots of the path plan and projected lighting on the terrain at various times over the 24-hour Experiment 1 traverse. The path proceeds in a clockwise sense through all the marked subgoals, and returns to the start position. The shape of the path illustrates how hazardous terrain can prevent an ideal circular path. Streambeds ran along the outside of both diagonal legs, and a rocky promontory rose just West of the northwest end of the route. Pre-designated subgoals steered clear of these terrain hazards. However, because of the elongated shape of the traverse, the planner had to reason about how to best time the route to minimize inevitable solar array mispointing.

The varying shading of the background reflects the changing average sun angle of incidence on the terrain, from roughly local noon in Figure 5a to just before midnight in Figure 5d. Predicted shadows appear as black regions in Figure 5d. Vectors representing the sun direction and solar array normal emanate from the rover position in each.

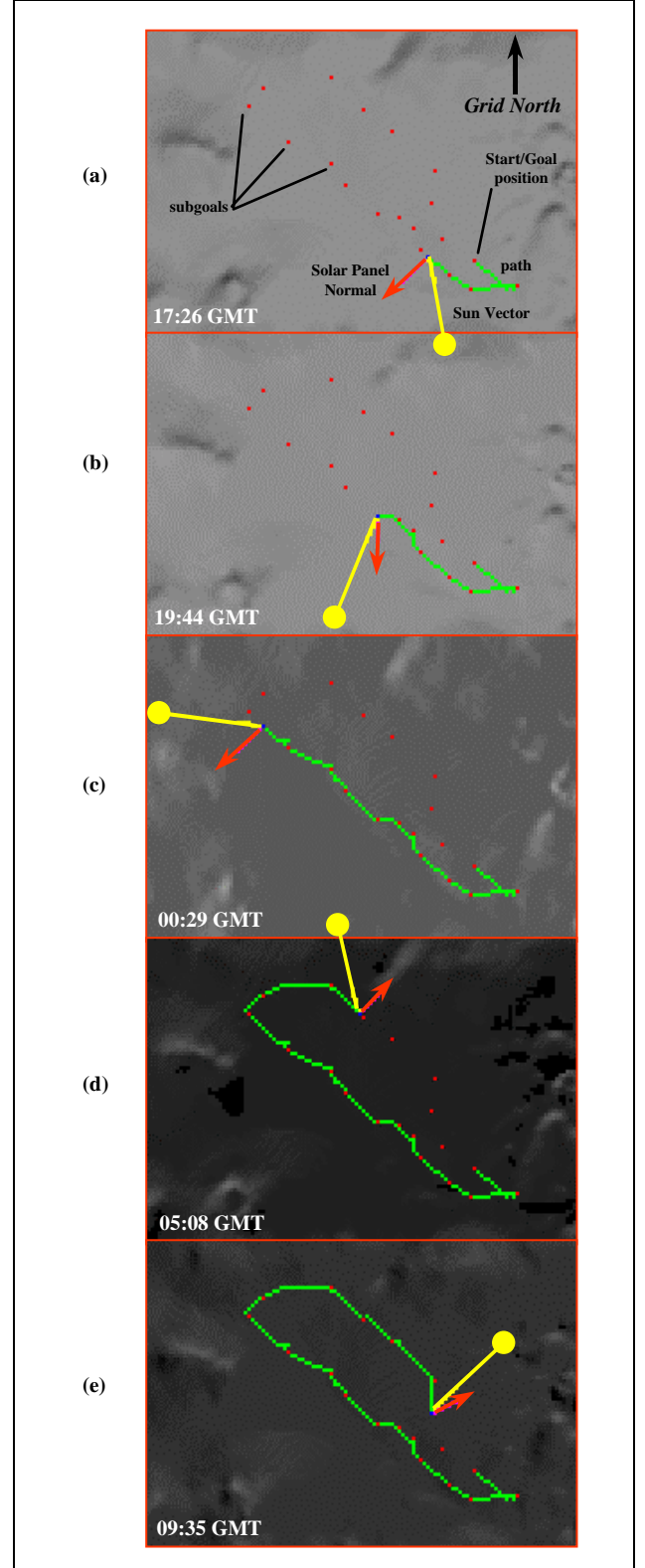


Figure 5: Depiction of the Experiment 1 path plan. TEMPEST biased the initial sun angle of each major diagonal of the path to ensure an optimal average sun angle over the day. Each frame is 6.00 km x 4.25 km.

frame. Note that the solar array normal points 90 degrees to the left of the driving direction, as it does for Hyperion (see Figure 1).

Figure 5a shows an early snapshot of the path. As with Hyperion, the solar array normal vector points to the left of the driving direction. Note how the sun direction is aft of the solar array normal. Meanwhile, in Figure 5c, the sun direction is forward of the solar array normal. These snapshots reflect TEMPEST's ability to schedule the path to achieve the best average sun angle, considering the entire path rather than favoring any path segment. The length of the northwest leg in these frames, coupled with a limitation in rover top speed, prevents an ideal sun angle over the course. Therefore, TEMPEST biases the sun aft at the beginning of the leg, in anticipation that the sun will overtake the rover near the center of the leg (Figure 5b), and be biased ahead of the rover by an equivalent angle at the end. A similar behavior occurs in Figure 5d and e on the return, southeast leg of the traverse.

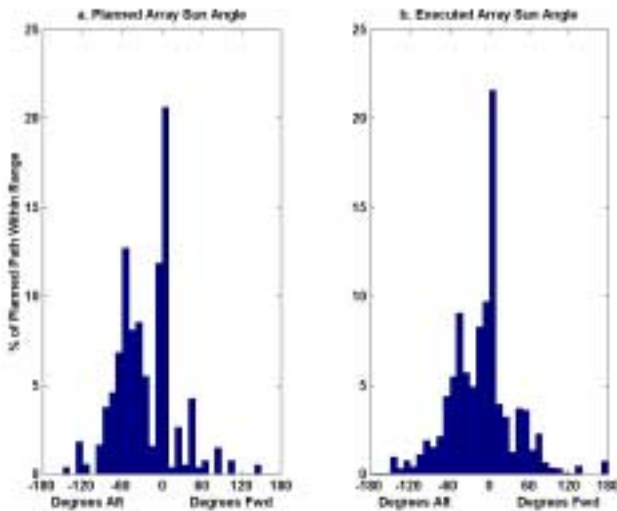


Figure 6: Histograms of Experiment 1 Solar Array Sun Angle (a) Planned, and (b) Executed

A measure of TEMPEST's ability to maintain sun exposure on Hyperion's solar array is shown in the histogram in Figure 6a. The histogram depicts the angle, in the plane of local horizontal, from the sun to the solar array normal resulting from the mission plan for Experiment 1. Zero degrees indicates optimal pointing, while negative and positive values indicate sun-aft and sun-forward conditions, respectively. The histogram indicates over 32% of the route is spent within 10 degrees of optimal solar array pointing. We attribute the bias in sun angles toward the aft of the rover to the Get Earliest Start Time Path heuristic. Following the application of energy-based heuristics, TEMPEST selects the earliest path opportunity. For

clockwise paths in the northern hemisphere, earlier opportunities will bias sun angles aft, as the Earth has not progressed as far in its daily rotation.

A second histogram in Figure 6b depicts the same quantities for the executed Experiment 1. Qualitatively, one can immediately see the similarities between the profiles, indicating integrity of the execution to the mission plan. Differences in these profiles are attributed to off-pointing due to specific actions taken by the local navigation system to avoid obstacles [13].

During the field experiment, TEMPEST required significant computational and memory resources. The mission planner was run exclusively offboard Hyperion for the field experiments. Experiment 1 took 5 hours 48 minutes to plan on a laptop with a 400 MHz Pentium II with 128 MB of RAM. Meanwhile, ISE planning graphs often exceeded 60 MB. Planning duration and memory requirements depend principally on search space size, defined by the dimensions of the four state variable axes, and the resolution at which each is specified. Decreasing resolution reduces the search space size for given problem, but negatively impacts search completeness and optimality.

5 Future Directions

The ultimate goal of this initial work is to make resource-cognizant mission planning an effective, fully-integrated component of a planetary rover autonomy architecture. We envision several parallel tracks of research and development to achieve our goal:

Performance: Performance enhancements are a critical step in putting TEMPEST onboard a rover. Preliminary improvements to cost calculations and ISE state hashing have resulted in a 3.75x increase in speed. Use of a monotonic, energy-based composite objective function will enable us to track energy and eliminate the energy dimension in the search space, with expected dramatic improvements in speed and memory usage. We may also trade some efficiency in ISE re-planning efficiency for speed in initial planning.

Probabilistic Plan Evaluation: Effective path planners must be robust to the high uncertainty of planetary environments and rover performance. Quantifying potential sources of error in terms of probability density functions will allow repeated simulation of path options and an estimate of a path's probability of success. This figure of merit might be used to select best path options, or as a means of monitoring plan progress during execution.

Re-planning/Contingencies: We intend to exploit ISE's capability for fast re-planning to perform online plan updates and pre-emptive contingency explorations. This will require a much greater integration with the rover to interpret the impact of mission progress and current rover state on planning.

Onboard/Offboard Architecture: We will explore performing the initial ISE search offboard, and then transferring the populated ISE search space to the rover for minimal computation re-planning and contingency operations. In a planetary exploration scenario, this architecture would make maximum use of ground assets, while enabling quick response autonomous planning on modest rover computers.

Domain Extension: We would like to develop a richer framework to simultaneously consider several constraints along with energy, and to implement multiple strategies depending on the rover's situation. Examples of constraints include communications and onboard data storage, or maintaining line-of-sight to ground targets. Alternate strategies might optimize the traverse based on different objective functions depending on current mission goals, prioritizing science, basic safety or emergency survival.

6 Conclusion

We have developed the TEMPEST software to autonomously plan long-distance, sun-synchronous routes through natural terrain. Unlike previous rover path planners, TEMPEST solves the coupled path, path timing and resource management problem. Deployed in the field, TEMPEST supplied plans that enabled the solar-powered Hyperion rover to traverse multi-kilometer paths in two continuous, 24-hour tests. Using the knowledge of planetary motion, terrain and the rover's fixed-orientation solar array, the plans maintained substantial battery energy reserves for the duration. This initial conceptual success in the off-board planner promotes further development towards a robust, on-board planning system adequate for planetary rovers. We seek substantial performance enhancements to minimize the operational overhead of planning, and will move to researching strategies for rapid re-planning, contingency analysis and probabilistic plan evaluation to contend with the uncertainty of operations in the planetary environment.

Acknowledgements

This research was funded under the NASA Sun-Synchronous Navigation project (contract NAG9-1256) and the NASA Graduate Student Research Program.

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] D. Bernard, G. Dorais, E. Gamble, B. Kanefsky, J. Kurien, G. Man, W. Millar, N. Muscettola, P. Nayak, K. Rajan, N. Rouquette, B. Smith, W. Taylor, Y. Tung, "Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment", *Proceedings of the AIAA 1999*, Albuquerque, NM.
- [3] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran, "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," *SpaceOps 2000*, Toulouse, France, June 2000.
- [4] S. Laubach and J. Burdick, "RoverBug: An Autonomous Path-Planner for Planetary Microrovers," *Sixth International Symposium on Experimental Robotics (ISER 99)*, Sydney, Australia, March 1999.
- [5] P. Schenker, T. Huntsberger, P. Pirjanian, E. Baumgartner, "Planetary Rover Developments Supporting Mars Science, Sample Return and Future Human-Robotic Colonization," *Proceedings of the IEEE 10th International Conference on Advanced Robotics*, Budapest, Hungary, August 2001.
- [6] K. Shillcut, "Solar Based Navigation for Robotic Explorers", Ph.D. thesis, technical report CMU-RI-TR-00-25, October 2000.
- [7] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, K. Schwehr, "Recent Progress in Local and Global Traversability for Planetary Rovers", *Proceedings from the IEEE International Conference of Robotics and Automation*, 1999.
- [8] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning", *Proceedings of IJCAI-95*, August 1995.
- [9] A. Stentz, M. Hebert, "A Complete Navigation System for Goal Acquisition in Unknown Environments", *Autonomous Robots*, Vol. 2, No. 2, August 1995.
- [10] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments", *International Journal of Robotics and Automation*, Vol. 10, No. 3, 1995.
- [11] A. Stentz, "Optimal Incremental Search for High-Dimensional, Constrained Path Finding Problems," *Carnegie Mellon Robotics Institute Technical Report*, to be released 2002.
- [12] D. Wettergreen, B. Shamah, P. Tompkins, R. Whittaker, "Robotic Planetary Exploration by Sun-Synchronous Navigation", *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 01)*, Montreal, Canada, 2001.
- [13] D. Wettergreen, B. Dias, B. Shamah, J. Teza, P. Tompkins, C. Urmson, M. Wagner, W. Whittaker, "Experiments in Sun-Synchronous Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington D.C., May 2002.