Mission Planning for the Sun-Synchronous Navigation Field Experiment
Paul Tompkins, Anthony Stentz, William “Red” Whittaker
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
pauldt@ri.cmu.edu, tony@cmu.edu, red@ri.cmu.edu

Abstract This paper describes the development and testing of software to enable an experimental, solar-powered rover to reason about solar energy, rover power expenditure, terrain and time in creating extended route plans in a planetary analog environment. Unlike previous path planners, this new software solves the coupled path, path timing and resource management problem that will be critical for future planetary surface missions. The mission planner combines mission domain information, a set of environmental and rover models and the Incremental Search Engine (ISE), a new search algorithm that produces optimal paths subject to constraints on state variables. In July 2001, the planning software supported the Sun-Synchronous Navigation Field Experiment on Devon Island in the Canadian Arctic. Experiments indicate the planner was successful in selecting time-sequenced, closed-circuit paths that would enable a planetary exploration rover to traverse an area indefinitely with battery energy reserve. The field trials also suggested future work in mission replanning, multiple resource constraint analysis and improved speed and memory performance.

1. Introduction
CMU embarked on the Sun-Synchronous Navigation project in response to an emerging need for robots that focus on resource availability and expenditures when planning their actions. Rover power systems combining solar arrays and batteries continue to be principal design candidates for proposed Mars rover missions through the end of the decade. Building on the performance of Sojourner, these missions seek an ever-growing reach beyond the landing site, and demand increasing levels of survivability and operational autonomy. Solar powered robots that strategize for path selection and activity placement and timing in terms of energy availability might yield significant improvements over current rover concepts in science data return. Resource management becomes even more difficult in challenging environments such as canyons, mountain ranges and craters, where terrain inhibits sunlight, limits communications and complicates locomotion. Despite the difficulties, missions of the next decade will likely explore such difficult terrain where science data tends to be richest.

As a component of the Sun-Synchronous Navigation effort, we developed automated mission planning software to enable a solar powered rover to operate over long durations in a polar environment. The software produces time-sequenced, closed-circuit paths that maintain a positive energy balance for a rover that employs a mechanically simple, fixed orientation solar array. Though applied to the sun-synchronous navigation strategy, the mission planning framework is applicable to general resource management on planetary exploration robots.

1.1. Sun-Synchronous Navigation
Sun-synchronous navigation is an energy-cognizant strategy for planetary exploration in polar regions. It involves selecting cyclic routes whose timing and specific course maintain sun exposure on a rover’s solar array over an entire day of operations, yielding a greater battery energy level at the end of the traverse than at the start [12],[13]. Since energy reserves are replenished on each cycle, such routes could repeat on a daily basis, enabling long-term survival and extended area coverage. Sun-synchronous navigation must also evaluate paths and path timing based on their energy costs, balancing the benefits of solar panel sun exposure and minimum distance path with difficult terrain crossings and entry into shadows. With current robot capabilities, sun-synchronous navigation would
allow a rover to fully explore a small region, traversing the area once for each day of a summer season. With greater terrain-crossing and speed capabilities, the strategy might enable a rover to repeatedly circumnavigate the pole of a slowly rotating planet or moon (e.g. Mercury or the moon [footnote here on solar day duration], selecting the path and speed to follow the daylight and avoid the night.

Polar exploration entails low sun elevation angles, even in mid-summer, and hence solar-powered robots operating near a pole must incorporate nearly vertical solar arrays (see Figure 1). An actively-pointed solar array decouples sun pointing from rover heading, and so sun-synchronous Navigation simply involves avoiding shadows. However, rovers with fixed solar arrays, mechanically simpler and lighter than those with a full pointing capability, must synchronize the timing of moves with the motion of the sun in the sky to achieve solar array pointing without actuation. With a fixed solar array and operating on a planar surface, optimal paths are circular and timed to maintain direct sun on the solar array for the entire route. More realistically, terrain and intermediate, pre-selected subgoals (e.g. science targets) may prevent circular paths. Because non-circular routes prevent the solar array from rotating at a constant rate, the specific choice of route and timing must be carefully planned to maintain sun-synchrony.

2. Prior Work

Substantial research and development efforts have yielded effective strategies for rover path planning to enable autonomous 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 [6], also produced with Mars rover navigation in mind, is based upon the D* algorithm [7],[8],[9]. 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 [6], and in evolved form on the CMU Hyperion rover [11]. 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 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 shadowed, sloping terrain.

In the arena of grid search applied to path planning, the D* (Dynamic A*) algorithm [7],[8],[9] 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 initial 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 [5] analyzes how various rover terrain coverage patterns affect incoming sunlight, and provides motivation for planning under such considerations.

3. Method

The Sun-Synchronous Navigation mission planner solves for time-sequenced path plans, with minimum battery energy guidelines, that enable a rover to operate long-term under solar power and rechargeable batteries. The planner runs offline prior to rover operations, but represents the deliberative layer in the Hyperion rover software architecture (Figure 2). At the early stage of research culminating in the Devon Island field experiment, 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 [11]. Future developments will fully integrate the planner onboard a rover.

The core of the planner is the Incremental Search Engine (ISE). ISE provides efficient search through high-dimensional state spaces, manages constraints on the feasible set of solutions, and enables quick re-planning [10]. Modules specifying mission objectives, mission constraints, operating environment and rover adapt domain specifics into the ISE framework. We discuss ISE and the domain specifications in the following sections.

### 3.1. Incremental Search Engine

ISE is a heuristic search algorithm that allows the mission planner to represent and reason about paths in two spatial dimensions, time and battery energy level. ISE plans an initial path given all known information about the world that satisfies path constraints and is optimal. As world conditions change, either through a re-specification of parameters during a planning cycle, or 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 only used in evaluating alternative path options in parallel. Future work will seek to incorporate the same feature for online, on-the-fly re-planning.

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. The mission planner uses the second option, as described in the Path Search section. Finally, to prevent searches that exceed practical durations, ISE incorporates a search timeout feature that sets the maximum number of explored states prior to termination.

### 3.2. Mission Domain Formulation

The mission planner searches for routes passing through a pre-determined, ordered list of position subgoals. The subgoals can be targets for science investigation or guides to steer the rover away from hazardous terrain. Additionally, subgoals provide a loose structure to the path to reduce the size of the search space. Because the list of subgoals is ordered, no attempt is made to simultaneously address the Travelling Salesman Problem. We refer to the path segments between each pair of adjacent subgoals as “legs.”

The mission planner considers two actions to transition between states: 1) drive to one of the eight adjacent cells, and 2) point the solar array to perform a fixed-duration, stationary battery charge. Each action results in time and energy costs, though energy costs can be negative.

Path state is represented by position, time and battery energy. Path plan solutions are sequences of state parameters from the start position of the rover through all subgoals. Position is the DEM grid cell position of the rover. Path plans comprise a sequence of waypoints at 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. Time is JPL CSPICE ephemeris time [1], referenced as seconds from January 1, 2000.
time state variable is used directly to determine the position of the sun and to reference lighting maps in the precalculated lighting sequence. Each waypoint in a path plan is tagged with the recommended arrival time of the rover at the waypoint. Battery energy represents the minimum battery energy required to complete the path from the current state, and therefore is a specification of lower bounds for a resulting path.

Through ISE, the mission planner enforces a number of constraints on plans. Fixed constraints prevent battery energy from violating battery charge limits, prevent moves out of bounds on the DEM and prevent path start times that fall outside the designated start time search window. Optional constraints impose maximum allowable slope and minimum allowable sun angle, and prevent travel through predicted shadows and operator-designated stay-out zones. The planner treats constraints as obstacles in the (x, y, time, energy) search space, preventing travel through restricted regions.

3.3. Path Search

The mission planner uses cumulative traverse time as the objective function to optimize paths. This allows both drive and charge transitions to be considered in equivalent cost terms, and still promotes minimum distance paths indirectly. Rather than using ISE to globally optimize over all path subgoals, the mission planner finds locally optimal solutions for each path leg. ISE uses a backward-chaining search and therefore the operator must specify all parameters of the goal state. The goal position is that of the final subgoal. Specification of goal battery energy and arrival time are more complicated.

Goal battery energy is arbitrarily set at halfway between the maximum and minimum charge limits. The desire to make sun-synchronous routes repeatable might suggest targeting the maximum energy level at the goal to ensure it exceeds the required minimum on the next repetition. However, forcing the battery to be fully charged at the goal would mandate that the final action have a negative cost (e.g. stationary charging), since a positive energy cost move would require the battery energy to be above the maximum in the previous state. Selecting an intermediate value seeks a significant goal energy level while not unnecessarily limiting move options.

Setting the goal arrival time is key to sun-synchronous navigation planning. The path start and goal arrival time are unknowns; both must be solved to yield a route that is synchronized with the sun. Additionally, given that the path distance is not known a priori and the rover speed is modeled as constant, the planner cannot calculate the precise duration of the traverse. Instead, it defines a day-long window of path start times, then based on estimated path duration with uncertainties, derives a broader window of possible goal arrival times (see Figure 3). Arrival times are selected throughout the window at even intervals, and considered separately in the ISE search.

The mission planner utilizes the quick re-planning feature of ISE to consider multiple goal arrival time cases simultaneously. The earliest goal arrival time is fed to ISE with the fixed goal position and energy values. ISE searches for a path from the previous subgoal to the current goal, populating a search graph with backpointer data in the course of this first iteration. Once it discovers a solution, or the timeout is reached, the planner replaces the goal arrival time with the next time in the window, and calls ISE again. Updating only those cells affected by the change, ISE returns almost immediately with results for this new case. The process repeats until ISE has considered every arrival time. Those cases yielding solutions are survivors. Their resulting start states are passed on as goal states for the previous path leg, chaining the solutions from later legs to earlier leg searches. The mission planner continues until one or more chained solutions survives at the start, that no paths survive, or that the search timeout is reached with no paths found.

![Figure 3: Depiction of Backward-Chained Plan Search. ISE enables consideration of many goal arrival times with little extra cost](image)

Often, more than one chained solution survives to the start. The mission planner uses three criteria to break ties among multiple survivors. First, the planner selects the solution with the minimum initial energy, requiring the least battery energy to complete the path. If more than one solution still remains, the mission planner selects the path with the lowest maximum battery energy over the entire path. Finally, if ties still exist, the planner selects the earliest opportunity. If the rover misses the first opportunity, the remaining opportunities are still valid.
3.4. Environment Modeling

The mission planner utilizes a number of environment models to calculate the costs of actions and to track constraint satisfaction during path search (see also Figure 4):

Terrain Models: Encoding elevation values to grid cells, Digital Elevation Models (DEM) form the basis for local slope estimates and line-of-sight occlusion for surface lighting maps. The spatial resolution of the DEM (cell size) is the same resolution at which the planner designates paths. DEM's used during the Haughton Field Experiment were 25 m resolution. The software 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 for rover orientation calculations.

Planetary Ephemeris: The mission planner uses the JPL CSPICE software [1] to determine the relative location and orientation of Solar System bodies. For the Sun-Synchronous Navigation project, CSPICE was used to determine sun azimuth and elevation.

Lighting Maps: Derived from terrain and planetary ephemeris data, lighting maps encode sun angle of incidence and shadow to cell position. The mission planner pre-calculates lighting maps at regular time intervals over one or more daylight cycles to create a dynamic lighting sequence. The mission planner refers to this sequence to determine lighting exposure and shadowing for cost and constraint satisfaction calculations.

Solar Insolation: The mission planner uses a constant value model for solar flux, representing the mean over one day, to enable solar energy collection calculations.

3.5. Rover Modeling

Rover models used by the mission planner are simplified but capture basic physical behaviors without the computational expense of using higher resolution models. The rover is assumed as a point for position, as the smallest terrain unit considered is the individual DEM grid cell. However, the planner represents the orientation of the rover solar array with respect to the rover driving direction and the local, slope-defined ground plane. Rover average speed is assumed constant for all drive state transitions. This speed attempts to take obstacle avoidance delays into account, and is typically substantially less than the rover operating speed on flat and level terrain.

The mission planner also uses simple models for energy collection and power consumption. The solar array is modeled by array area and an overall efficiency. Locomotion power is calculated by multiplying the rover mass, an effective coefficient of friction and rover speed. Slope further contributes by decreasing the locomotion friction force and adding a direct gravity force component. Because background electronics power consumption varied insignificantly during rover tests, the planner assumes a constant value. The rover battery model assumes an upper and lower bound on charge, but does not currently have limits on charge or discharge rates. Energy collected in the battery is simply the difference between energy collected and energy consumed. Future refinements could easily add charge rate limits to prevent invalid scenarios.

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 several 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. This paper details results from the first experiment. Table 1 summarizes the experiment as it applied to the planner.

<table>
<thead>
<tr>
<th>Table 1: Summary of Experiment 1 Plan and Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quantity</strong></td>
</tr>
<tr>
<td>Distance (m)</td>
</tr>
<tr>
<td>Duration (hr)</td>
</tr>
<tr>
<td># of initial subgoals/ # planned actions</td>
</tr>
<tr>
<td>% within 10 deg of optimal pointing</td>
</tr>
</tbody>
</table>
Figure 5 depicts the paths, both planned and executed, for Experiment 1. The shape of the path indicates the degree to which terrain prevented an ideal circular path. Streambeds ran along the outside of both diagonal legs, and a rocky promontory rose to the 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 offpointing.

A measure of the mission planner’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 mission planner’s standard solution tie-breaking strategy. If the planner finds several time-sequenced paths that are equivalent in terms of performance metrics, it selects the earliest path opportunity. For clockwise paths in the northern hemisphere, earlier opportunities tend to 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 [11].

In its current form the mission planner requires significant computational and memory resources. The mission planner was run exclusively offboard for the field experiments. Several laptops with Pentium III processors were used over the course of the experiment, the fastest machine with a 900 MHz clock speed and 256 MB of RAM. The planner required 4 to 8 hours to generate plans, depending primarily on the number of subgoals, which were typically spaced 10 to 20 cells apart. The mission planner’s speed increases substantially by reducing the resolution at which time and battery energy are represented and employing ISE resolution pruning. The increase in speed comes with a penalty in completeness. Meanwhile, ISE planning graphs often exceeded 60 MB, also dependent on time and energy resolution.

5. Future Work

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: Through a combination of intelligent use of variable state space resolution and software engineering, we expect to substantially increase software speed and memory efficiency.

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 different elements of the traverse depending on current mission objectives, or enable emergency survival responses.

6. Conclusion

We have developed software to autonomously plan sun-synchronous routes through natural terrain. The planner was employed in a field experiment in the Canadian Arctic where it planned routes for the solar array and battery powered Hyperion rover. It successfully solved for routes that allowed Hyperion to traverse multi-kilometer routes in two continuous 24-hour experiments. ISE, the planner's underlying search algorithm, provides a general framework for optimal path planning, constraint management and rapid re-planning. Future research will exploit these capabilities to enable general, online, resource-cognizant path planning for planetary rovers.

7. Acknowledgements

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

8. References


