IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Terrain-Aware Path Planning and Map Update for Mars Sample Return Mission

Gabrielle Hedrick[†*], Nicholas Ohi[†], *Student Member, IEEE*, and Yu Gu[†*], *Member, IEEE*

***Abstract*—This work aims at developing an efficient path planning algorithm for the driving objective of a Martian day (sol) that can take into account terrain information for application to the proposed Mars Sample Return (MSR) mission. To prepare the planning process for one sol (i.e., with a limited time allocated to driving), a map of expected rover velocity over a chosen area is constructed, obtained by combining traversability classes, rock abundance and slope at that location. The planning phase starts offline by computing several potential paths that can be traversed in one sol (i.e., a few hours), which will later provide suitable options to the rover if replanning is necessary due to unexpected mobility difficulties. Online, the rover gains information about its environment as it drives and updates the map locally if major discrepancies are found. If an update is made, the remaining driving time along the different options is recalculated and the most efficient path is chosen. The online process is repeated until the rover has reached its daily goal. When simulated on different areas at Gusev Crater, Mars, the algorithm correctly captured changes of terrain initially not mapped, and rerouted the rover to a more efficient path when necessary, in which case it effectively complied with the time constraint to reach the goal.**

***Index Terms*—Space Robotics and Automation, Robotics in Hazardous Fields, Autonomous Vehicle Navigation, Mars Sample Return, Map Update.**

I. Introduction

FUTURE surface missions on Mars and other bodies in the Solar System will require autonomous rovers to face more difficult duties than previous missions have performed. Despite having remote sensing data for Mars, including 25 cm/pixel images (HiRISE, High Resolution Imaging Science Experiment [1]) it gives insufficient terrain information needed for planning purposes, as it does not capture current conditions at the surface [2] which translates into difficulty in preparing for day to day robotics operations. For example, Spirit encountered high deformable soil at the Columbia hills that had not been detected in orbital and ground images and made traversability a challenge [3]. Opportunity encountered high wheel sinkage situations at Endeavour Crater [4], and Curiosity experienced mobility difficulties with wheel damage (holes and dents) from roving on sharp rocks [5]. The importance of understanding terrain properties prior to driving has since been widely recognized, but for past and current Mars rovers, several limiting factors can be listed. First, the rover planners are limited to line-of-sight driving from the few images received, and limited communication [6] added to limited downlink data leads to few opportunities to plan drives, not scheduled every sol [2]. The maximum distance driven in one sol has never been greater than 143 m (Curiosity on sol 665 [7]) and a rover such as Opportunity drove an average of 3 km per Earth year [8]. To remedy this problem, "autonomous" navigation (AutoNav) has been implemented, but the distances covered are short. Indeed, AutoNav takes pairs of stereo images to map hazards or rough terrain with limited on-board computational power, leading to slow speeds, e.g., 10 m/hr to 36 m/hr for the Mars Exploration Rovers (MERs) [9]. To cover greater distances faster, the rover can be programmed to "blind" drive, i.e., without checking its surroundings (a.k.a "directed"), when the terrain is declared safe (from assessment of images by humans). For MERs, it enabled speed up to 124 m/hr [9], but it also implies that the drive remains within line-of-sight (i.e., path visible on pictures collected from previous sols).

NASA's proposed mission Mars Sample Return (MSR) would collect material to bring back to Earth and would be completed within one Martian year (687 days) [10]. With its main purpose being to retrieve samples up to 10 km away due to landing uncertainty [11], in order to maintain its mission timeline the MSR rover will require more autonomy to safely perform more frequent (potentially every sol) and longer drives than previous rovers. It would need more efficient measures to address unexpected conditions as well as a faster, more autonomous decision making process. MSR needs a different approach in path planning that can take into account the main limitations of current Mars rovers: the lack of information about the terrain, the need to drive further and faster without more frequent communication between Earth and Mars, the limited on-board computational resources, and the lack of human input on regions not covered by surface images from previous sols. Some of these challenges also apply to other planetary missions, such as Dragonfly to Saturn's moon Titan [12]), and solutions to address these problems on Mars could potentially be extended to other planetary rovers.

The rest of this letter is organized as follows: section II discusses previous work related to Mars rover path planning, terrain and traversability. Section III shows the problem statement and contribution of this paper. Section IV describes the technical approach; results are presented in section V. A conclusion and future work are discussed in section VI.

[†]Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, USA
[*]G. Hedrick: glc0006@mix.wvu.edu,
Y. Gu: yu.gu@mail.wvu.edu

II. Related Work

Planetary environments pose the problem of being partially unknown with uncertainty about the terrain. This creates difficulty in motion planning and mobility, and raises the question of traversability. Traversability is the ability for a terrain to support a driving vehicle without reaching the failure point, and depends on both the robot and the terrain. Being able to predict it is key to a successful surface mission and has been attempted many times by researchers with different approaches. For example, on most Mars rovers, AutoNav has been implemented to perform real-time checks of the environment in the immediate vicinity of the robot [13] before a field D* algorithm finds paths through a grid-based representation of the environment [14]. Additionally, research has been conducted to characterize and classify the terrain using a variety of on-board sensors [15], [16]. Some authors have proposed to incorporate terrain uncertainty directly into the planning phase by implementing a path with an adaptive cost that changes according to new information gathered along the drive (i.e., the cost changes as the environment is updated and uncertainty is reduced) [17]. Unlike AutoNav, this method can account for a budget (e.g., limited time to reach the goal each sol). Some other researchers have suggested to equip the rover with a spectrometer to gather information along the way, allowing the vehicle to update its route through Bayesian inference using the newly acquired knowledge of its environment [18]. This monitoring method might not be easy to implement for a sample return rover, as the systematic use of a spectrometer would be time consuming (about 3 hours each time, [19]) and could be computationally demanding, especially for a mission whose main purpose is not to conduct science experiments in real time. However, using a different instrument could potentially alleviate this difficulty, provided the hardware is easy to implement and deploy (e.g., shear vane [20]). This concept closely resembles a map update, where the rover gather information about its environment online to update its knowledge and plan its traverse accordingly. This has been demonstrated to work for local planning in an unknown, off-road environment [21], and for planning in hazardous environments containing radioactive materials [22]. However, the mapping and planning phases are both performed online, which might be a challenge for a planetary rover with limited on-board resources and time to perform computations. Another algorithm was suggested to help rover drivers plan paths, where ground images and topographical information would be merged to predict potential terrain hazards ahead of the vehicle [23]. A slightly different approach to this is the use of terramechanics (the science of wheel-terrain interaction) as a mean of assessing traversability. Slip has been widely chosen for this purpose [24] and has been cited as early as the foundation work of terramechanics [25], [26]. One proposed model uses on-board cameras to predict slip and consequently classify the terrain into three categories (traversable, untraversable, and uncertain) to then plan a path accordingly [27]. Additionally, some authors have developed a path planning algorithm that accounts for error in predicted slip [28]. However, slip prediction methods are often online and limited to line-of-sight estimations. Offline simulations of traverses with a software specifically developed for this application (ARTEMIS, Adams-based Rover Terramechanics and Mobility Interaction Simulator) have been researched as well [29], [30], validated by single-wheel experiments and used on the Mars rovers Opportunity and Curiosity [31]. Similarly to most slip prediction methods, this is also constrained to limited distances on terrain assessed via ground images. To address this issue, a different mobility model was developed that factors in terrain information such as types and rock abundance (Cumulative Fractional Area or CFA [32]) in addition to slope, initially used alone [33]. This classifier, the Mars Terrain Traversability Tool (MTTT), performs a traversability analysis and classifies terrain types into categories based on rover performance (trained with data from MSL, the Mars Science Laboratory) [23]. In this model, terrain types are estimated using the Soil Property and Object Classification (SPOC) software developed at the NASA Jet Propulsion Laboratory (JPL) [34]. MTTT is then used to compute optimal-time routes offline over extensive distances, but it is intended for scientific missions and therefore, the paths do not converge to a single goal [35] and it would not provide an effective replanning strategy for MSR. A recent planning approach worth mentioning consists of computing the entire traverse globally, before using a local planner with higher resolution to dynamically correct the path as obstacles are detected in real-time [36].

This paper proposes a new framework that leverages these existing methods to meet the needs of future planetary surface missions such as MSR: an effective path planning method that predicts rover traversability and limits the use of computational resources online; local map updates to assess the safety of the terrain ahead of the rover; and an efficient, autonomous replanning strategy when unexpected hazards are detected.

III. Problem Statement

The goal of this work is to develop a path planning and real-time map update algorithm to support the autonomous traverse of a sample return rover on partially unknown terrains for the driving objective of a sol on Mars. This process would be repeated each sol until the traverse from the landing site to the sample locations is completed. The information available to this research in the context of MSR are the raw orbital data (e.g., Digital Elevation Model or DEM) and processed orbital data (CFA and terrain types). It is assumed that the rover has reasonably accurate knowledge of its position at all times, that only the traversability class data set contains uncertainty and that the rover can continuously drive for up to eight hours per sol. Even though a solar powered rover such as MERs was designed to drive at most four hours by design and in practice even less [9], better performance is expected in the future [6]. The assumption is made that the computational resources are at least equivalent to those of Mars 2020 with a Virtex-5QVs

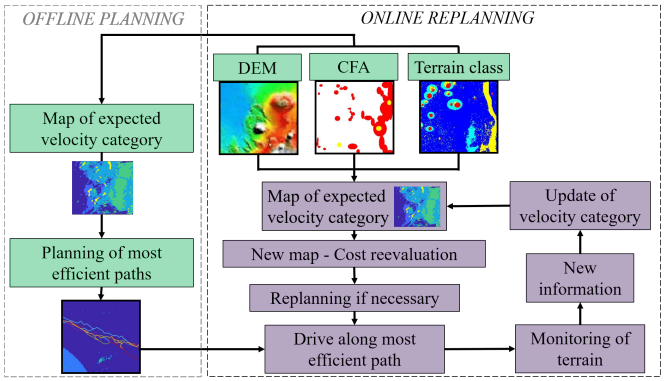


Fig. 1. Diagram showing the one-time offline planning and the online loop comprised of update and replanning.

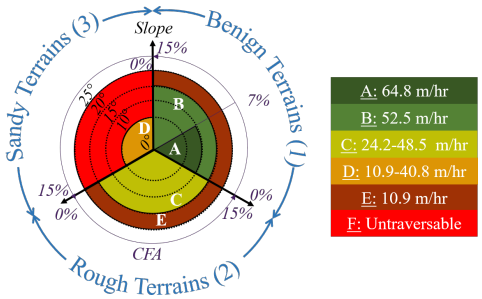


Fig. 2. Velocity categories, A to F (CFA >15% or slope >25° are in category F and are not shown).

Field Programmable Gate Array (FPGA) [37] and RAD750 Single Board Computer (SBC) [38]. Scheduler algorithms of complexity up to $O(TN_a^3)$ (T number of timelines and N_a number of activities) have been developed for such computers [38] and it is assumed that any algorithms of equal or lower complexity can therefore be handled by similar computational resources. Additionally, it is assumed that the rover can perform real-time terrain monitoring [16], [39] and obtain information about its type through such monitoring. Constraints associated with this work include a time budget (limited time each sol to perform a drive) and limited data sets available. That is, the amount and type of data used are specific to Mars and the planning algorithm was built to focus on future surface missions to the red planet.

The main contribution of this work is a novel terrain-aware planning framework that supports the MSR mission concept by enabling safe driving over significant distances. It includes: 1) a planning algorithm that can incorporate orbital terrain information to allow the rover to autonomously assess its environment prior to driving; 2) an algorithm that accounts for new information gathered by the rover and locally lowers the uncertainty about the terrain; 3) a planning process that gives the vehicle options to reroute if a more efficient path is found without requiring heavy real-time computation, resulting in faster decision making and replanning.

IV. Technical Approach

The path planning process is split into an offline path planning phase and an online map update and replanning loop. This approach was chosen to alleviate as much as possible the use of computational resources on-board. The offline planning creates paths for the rover to use during its traverse, whereas the online part is dedicated to monitoring of the terrain and map updates given new terrain information, with possible replanning if needed (i.e., terrain has changed significantly). The overall process is presented in Fig.1 and detailed in the next subsections.

A. Definitions

The following definitions are used throughout this letter:

- Terrain types: direct output of SPOC [34]. At the Columbia Hills, the following are found: SR - Smooth Regolith and SO - Smooth Outcrop, RR - Rough Regolith and RO - Rough Outcrop, SRF - Sparse Ripples Firm and SRS - Sparse Ripples Sandy, DLR - Dense Linear Ripples and PR - Polygonal Ripples.
- Traversability class: each class of traversability is comprised of two terrain types (see Table I). The classes are: benign (1), rough (2), sandy (3), no-AutoNav (4) and untraversable (5). At Gusev Crater, class 4 has not been found [34].
- Cumulative Fractional Area (CFA): fraction of area covered by rocks [32].
- Velocity categories: lettered A to F, they represent the expected rover velocity given the slope, CFA and traversability class [35], as shown in Fig. 2. This paper does not focus on possible variations of velocity within each categories but rather, takes the optimal value/range as published in [35] for each one.

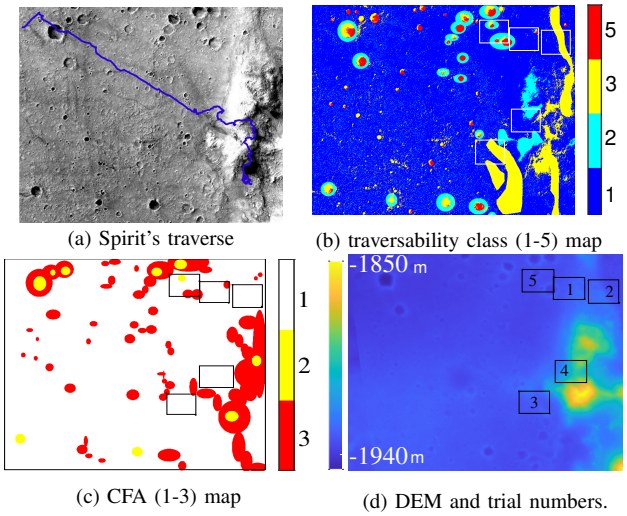
B. Offline Path Planning

This step focuses on generating offline a diverse set of paths for the driving objective of a sol based on terrain information. Spirit's landing site is chosen to simulate the algorithm since detailed terrain studies have been conducted there [34].

1) Velocity Categories: The path planning method uses expected velocity categories derived from combining three geological data sets [35]. Indeed, by overlapping the CFA, DEM and traversability class maps over each 1x1 m^2 cell, a geological context is retrieved that allows estimation of an expected speed of the vehicle across each square meter of the map, using the classification shown in Fig. 2 [35]

2) Map of Velocity: The map known to the rover prior to planning is a belief map, specifying the probability distribution over the expected velocity categories of each of the grid cells (see subsection IV-B1). This map is built over an area potentially covering the equivalent of the journey from landing to samples and Spirit's traverse was taken as a reference (Fig.3(a)). The belief map is initialized from the uncertainty contained in the terrain type map generated by the classifier as presented in [34].

To build the belief map, CFA, traversability classes and slope information are needed, but only the DEM is public information. New maps were therefore hand labeled using HiRISE images and DEM over the chosen area. Even if additional uncertainty could be present in the traversability



(a) Spirit's traverse (b) traversability class (1-5) map

(c) CFA (1-3) map (d) DEM and trial numbers.

Fig. 3. (a) HiRISE PSP0017771650REDA01 with Spirit's traverse in blue (Image: NASA/JPL/University of Arizona). (b) Traversability classes (1, 2, 3, 5). (c) CFA map; 1: CFA <7%, 2: 7%< CFA <15%, 3: CFA >15%. (d) DEM with subset trial maps (see Table II for numbering). Maps are 2915x3540 m^2, subsets are 301x401 m^2.

class map due to the fact that the map had to be manually labeled, it is assumed that the data set provides a reasonable approximation of the Mars conditions for evaluating the proposed planning algorithms, and that the uncertainty from the classifier used in [34] can be applied to this work. CFA is divided into three ranges (low or 1, <7%, medium or 2, between 7% and 15%, and high or 3, >15%) and slope is taken within five main ranges (< 10°, between 10° and 15°, between 15° and 20°, between 20° and 25°, and >25°) [35]. The CFA map was built based on visible geological features such as craters, assuming that rocks are more abundant around them (ejecta blanket, high CFA), a little less inside (medium CFA) and a lot less abundant everywhere else (low CFA) [40]. The traversability class map was obtained in a similar fashion: untraversable class was matched with craters, and sandy terrains were matched with ejecta blankets and dunes or ripples when they could be detected. Rough terrain was built using grey threshold values from the DEM and the rest of the map was set to benign terrain by default. Therefore, classes 1 and 2 (benign and rough) do not correspond to actual geological units, unlike classes 3 and 5. The maps are shown in Fig.3, with 3(a) having a resolution of 25 cm/px and 3(b), 3(b) and 3(d) having a resolution of 1 m/px. To test the performance of the algorithm, small portions of the map were selected, chosen so that they would cover the distance traversable in one sol (i.e., taking no more than eight hours), and are 301x401 m^2 in size (shown as rectangles in Fig.3).

3) Planning Approach: To reduce the computational cost online, several paths are computed offline prior to begin the traverse of a sol, to give the rover options to reroute in real time if necessary. This approach enables the rover to quickly replan its traverse without heavy computation involved, should the terrain be different than expected. Rerouting could be triggered by any significant change detected by the rover as it drives and it is assumed that various methods such as slip monitoring or instrument deployment can detect terrain variations. The start and goal locations are chosen so that the rover would not take more than eight hours to drive from one to the other under nominal conditions (Fig. 4). To plan the paths, a graph is first built using a Probabilistic Road Map (PRM) [41] and an A* algorithm [42] then computes paths from start to goal. The cost of the traverse is the sum of the cost of the edges along the path given by Eq. 1.

$$e_c = \frac{l_e}{V_e} \tag{1}$$

Where e_c is the edge cost (in hours), l_e is the length of an edge in meters and V_e is the maximum velocity encountered by the rover along the edge (in m/hr). Each cell corresponds to a pixel on the orbital images, with a resolution of 1 m/px. V_{px} is obtained from [34] by overlapping the slope, traversability class, and the rock abundance for each cell (see subsection IV-B1). The slope is derived from the DEM using Horn's method in the direction of driving (simplified to "up" or "down") as shown in [35]. The traversability class and CFA values are obtained from the maps made from HiRISE images as described in subsection IV-B2. To compute several paths to the goal, Yen's k-shortest paths algorithm [43] is implemented on the node following the spur node of the latest computed path. The algorithm was modified to incorporate the following: maximum cost of each path (i.e., time budget); and minimum distance from each node on previous paths to avoid computing paths too close to each other. The number of paths with this no-go corridor can be changed, and the width of the corridor is no longer than twice the maximum length of an edge. With this method, the maximum number of options generated is the number of edges of the second-to-last computed path, and it allows for both diversity and overlapping of the paths.

C. Local Map Update

The map update allows the rover to keep a local up-to-date velocity map at 1 m/px using constant monitoring of the terrain while driving. The initial map known to the rover contains uncertainty due to the terrain classifier errors [34]. For simulation purposes, the "measurements" resulting from terrain monitoring are taken from an altered map referred to as the "truth" map (1m/px). The alterations were made so that the map remains as close as possible to real situations: for example, class 1 has 10.5% chance to be mistaken for class 2, but class 6 can hardly be misclassified, with a prediction accuracy of 99.8% [34] (Table I). The initial belief states of each cell are derived from statistical proportions of each velocity category under different ranges of CFA and slope. The derivation of these matrices is presented below.

1) Class Confusion Matrix: The probability matrices are derived from the class confusion matrix (Table I), i.e., the matrix showing the performance of the terrain classifier in [34]. It is first adapted from types to classes with the following process:

- The proportion of each class present at the Columbia Hills was computed by simply adding the proportions of each terrain belonging to the class:

$$p_{c_i} = \sum_j p_{t_j \subset c_i} \tag{2}$$

where p_{c_i} is the proportion of class i and $p_{t_j \subset c_i}$ is the proportion of terrain type j included in class i.

- Each terrain proportion within a class is obtained by normalization of the proportions:

$$p_{t_i \subset c_j} = \frac{p_{T_i}}{p_{c_j}} \tag{3}$$

where $p_{t_i} \subset c_j$ is the proportion of terrain type i in class j, p_{T_i} is the proportion of terrain type i at the Columbia Hills as observed by [34] and p_{c_j} is the proportion of class j at the Columbia Hills as calculated previously.

- Each value within the confusion matrix was calculated based on the proportion of each terrain within each class and the probability given by the confusion matrix for that terrain in [34]:

$$P(c_j) = \sum_i p_{t_i \subset c_j} P(t_i) \tag{4}$$

where $P(c_j)$ is the performance of the model for class j (computed as explained above), $p_{t_i} \subset c_j$ is the proportion of terrain type i in class j and $P(t_i)$ is the performance of the model for terrain type i (from [34]).

TABLE I
CONFUSION MATRIX FOR TRAVERSABILITY CLASSES 1 TO 5.

		Prediction			
		SR+SO Class 1	RR+RO Class 2	SRF+SRS Class 3	DLR+PR Class 5
Ground Truth	SR+SO	84.0%	7.50%	0.200%	0%
	RR+RO	10.5%	81.3%	1.20%	0.200%
	SRF+SRS	5.20%	10.6%	98.4%	0%
	DLR+PR	0.300%	0.600%	0.200%	99.8%

2) Edge Potential Matrices: The edge potential matrices, representing the conditional relations between velocity categories in adjacent nodes, are built after the confusion matrix for classes shown in Table I. For each category (see Fig. 2), it is first determined what classes of terrain are included, what the proportions of each class within the category are, and finally, the probability that the prediction matches the ground truth is calculated. Each column of the edge potential matrices corresponds to the belief Probability Mass Function (PMF) of velocity categories for each combination of CFA and slope; each row of the matrices is the "true" PMF of categories (i.e., the "measurement"). To illustrate the derivation, an example is taken for CFA less than 7% and slope between 20° and 25°. This combination of CFA and slope gives only two possible categories of velocity: E (10.9 m/hr) and F (untraversable). E includes class 1 and 2. Moreover, within these ranges of CFA and slope, category F can only be class 3 or 5. To calculate the probability that category E is mistaken for F, the following steps are followed:

- The relative proportions of class 1 and class 2 within E are obtained, called $p_{c_1 \subset E}$ and $p_{c_2 \subset E}$, respectively:

$$p_{c_i \subset C_j} = \frac{\sum_k p_{t_k \subset c_i}}{\sum_i p_{c_i}} \tag{5}$$

where $p_{c_i \subset C_j}$ is the proportion of class c_i within category C_j, $p_{t_k \subset c_i}$ is the proportion of terrain type k within class c_i and p_{c_i} is the proportion of class i at the site of interest, calculated from Eq. 2.

- The relative proportions of class 3 and class 5 within F is calculated, called $p_{c_3 \subset F}$ and $p_{c_5 \subset F}$, respectively.
- The numbers in Table I corresponding to the performance of the model for each predicted class (1 and 2) compared against each true class (3 and 5) is taken, called $P(c_1|c_3)$ (probability of the model to give class 1 knowing that it is in fact 3), $P(c_1|c_5)$, $P(c_2|c_3)$ and $P(c_2|c_5)$.

The final equation is:

$$P(E|F) = p_{c_1 \subset E}(p_{c_3 \subset F} * P(c_1|c_3) + p_{c_5 \subset F} * P(c_1|c_5)) + p_{c_2 \subset E}(p_{c_3 \subset F} * P(c_2|c_3) + p_{c_5 \subset F} * P(c_2|c_5)) \tag{6}$$

Which gives, in this case: $0.54(0.436{*}5.2\%{+}0.564{*}0.3\%){+}0.46(0.436 * 10.6\% + 0.564 * 0.6\%) = 8.31\%$.
This number is highlighted in Table A4. The other cells are calculated using a similar process. All of the edge-potential matrices are presented in Appendix A[1].

3) Belief Propagation Framework: Each cell s in the map has an associated belief state of the velocity category of that cell $\phi(s)$. The initial belief state of a cell is derived from the PMF of velocity categories, by tuning the probabilities towards the most likely categories as seen from ground images assumed to be available to MSR from Perseverance. Here, for simulation purposes, the truth map is used to set the initial conditions. The map update problem is simplified by estimating the belief only at the nodes along the computed paths in the local area around the location where the measurement was taken. The graph used for belief propagation is found as a minimum spanning tree of the subgraph, consisting of the subset of nodes less than or equal to a graph distance of N away from the measured node. This minimum spanning tree of the local subgraph around the measurement is referred to as the *belief propagation tree*, the nodes of this graph are the set $\{x\}$, and the belief state of a particular node x_i is $\phi(x_i)$, representing a Markov Random Field (MRF). The belief of the rest of the cells in the local area of the map is then updated by linearly interpolating between nodes in the belief propagation tree. The edge potential functions $\psi(x_i, x_j)$ represent conditional relations between neighboring nodes in the graph and describe how the state of x_j should be updated, based on the state of x_i. A measurement taken at a particular node z_i directly results in a new belief state for that node $\phi(x_i|z_i)$. The objective of the map update is to propagate this new information to the rest of the graph and infer the marginal PMFs of each node. In other words, the algorithm updates the belief states of all other vertices, given the measurement, $\phi(x_j|z_i), \forall j \neq i$. The minimum spanning tree of the local subgraph is found (with a time complexity of $O(|E|log|\nu|)$, where $|E|$ is the number of

[1]When used as edge potentials in the map update framework, the columns are normalized, so each sums to 100%.

edges in the graph and $|\nu|$ the number of nodes), so that exact inference can be performed using the sum-product message passing algorithm [44]. The belief propagation in this work is implemented using conditional inference on trees, provided by the Undirected Graphical Model (UGM) Matlab Toolbox [45]. The belief propagation tree MRF is factorized and the marginal distributions of each node are computed using the following equations, as described in [44], with a complexity of $O(|\nu|^2)$ (ν vertices of the tree, in this work $|\nu| < 100$).

$$\phi(x_i) = \sum_{\{f\}} \mu_{f \to x_i}(x_i) \tag{7}$$

$$\mu_{f \to x_i}(x_i) = \sum_{\{x\} \setminus x_i} \left(f(n(f)) \prod_{x_j \in n(f) \setminus x_i} \mu_{x_j \to f}(x_j) \right) \tag{8}$$

$$\mu_{x_i \to f}(x_i) = \prod_{h \in n(x_i) \setminus f} \mu_{h \to x_i}(x_i) \tag{9}$$

Where in general, $\mu_{f \to x}(x)$ is the message passed from a factor node f to the variable node x and $\mu_{f \to x}(x)$ is the message passed from a factor node f to a variable node x.

The terrain monitoring on the nodes along the path leads to a "noisy measurement" of conditions corresponding to a specific PMF of velocity categories. It is assumed that such monitoring (e.g., slip) can estimate such PMF, sampled from the "truth" map for the simulations. The algorithm then generates the 1 m/px map through interpolating the information from the updated nodes. This paper does not address the real-time monitoring process (see section III).

D. Replanning

Replanning is used to optimize the remaining time to the goal. After each map update, if the most likely velocity category along the path the rover is driving on is different than the believed velocity, the costs of all the pre-planned paths are recalculated using a weighted average of belief cost of each cell crossed by the remaining edges of the path. The belief cost for a cell is given by:

$$c_b = \frac{l_c}{\boldsymbol{V} \cdot \phi(s)} \tag{10}$$

where $\boldsymbol{V}$ is a vector of velocity for categories A to F and $\phi(s)$ is the belief state PMF of categories in cell s. If there is an available path among the proposed options with a lower cost than the current path the rover is following, and if there is a connection, i.e., segments of pre-computed paths from the rover's location to the potential new path, this option is selected to continue the traverse. The complexity of the replanning algorithm is simply $O(nlog|\nu|)$ in time and $O(log|\nu)|$ in space, where ν are the vertices on the paths.

V. Results

Five simulations are performed, and the first one is detailed step-by-step in subsection V-A. The results of the other four simulations are summarized in subsection V-B. The start and goal locations are generated randomly with the budget constraints that the time it takes the rover to go from one to the other is no more than eight hours.

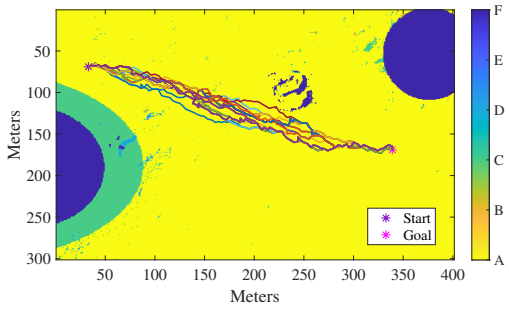


Fig. 4. Inital map of velocity categories A to F, with 37 paths.

A. Detailed example

For this particular example, there are 37 paths computed within the allotted time budget (costs are between 5.5 and 6.4 hours) with a path length between 354.5 m and 430.2 m. Monitoring is performed every four nodes starting on the first node of a chosen path, and the maximum length of a branch of the belief propagation tree is $N = 3$ nodes. Following a change of velocity from A to C (Fig. 5(a)) and a cost recalculation, the initial path chosen is no longer the fastest. The local update is shown in Fig. 5. A different option is found that minimizes the remaining time to the goal by avoiding the terrain in category C for as long as possible (Fig.6), thus gaining about 45 minutes. This example shows that the diversity of the paths is needed to offer less costly alternatives from the current route, but that having overlapping segments is necessary to allow for quick deviations from the original path. The rover gets to its goal in six hours and ten minutes in the simulation, and would have taken seven hours along the initial route.

B. Other trials

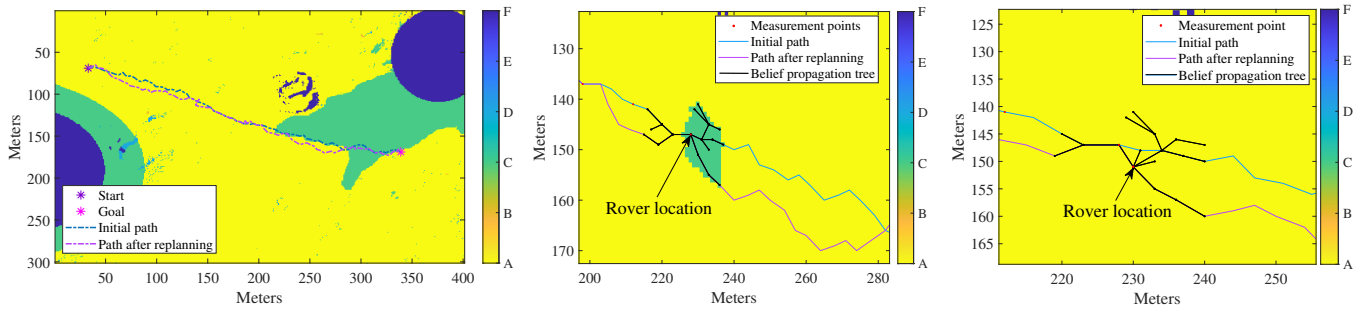
The algorithm has been tested on other subset maps shown in Fig. 3. The results are presented in Table II, with trial one being the case detailed above. These trials show that replanning is not always needed, despite the occurrence of map updates. If the remaining time to the goal after an update is still the shortest along the current path, the rover maintains its course.

TABLE II
TRIALS 1 TO 5 (SEE FIG.3(A) FOR LOCATION).

Trials, distance start-goal	1(344m)	2(283m)	3(186m)	4(344m)	5(231m)
Categories along path	A,C	A,C,D	A,C,D	A,B,C,D	A,C,D
Cost initial path	5.48hrs	7.32hrs	3.40hrs	7.13hrs	5.06hrs
Replanning events	1	0	0	1	0
New cost of initial path	6.99hrs	N/A	N/A	8.33hrs	N/A
Cost final path	6.17hrs	7.42hrs	3.82hrs	7.63hrs	5.12hrs

VI. Conclusion and Future Work

This work presents a planning framework that addresses challenges posed by Mars surface missions, proposing a method that incorporates terrain information in the planetary rover path planning and gives efficient replanning steps while keeping the on-board computational cost to a minimum. Offline, paths are planned to provide diverse options to the rover to divert from its original route if needed. Online, the map of expected velocity is kept up to date locally by monitoring the



(a) Truth category map. (b) Local update before replanning. (c) Local update after replanning.

Fig. 5. (a) Truth map of velocity category (A to F) showing the two paths included in the actual traverse. (b) Local update before a replanning event. (c) Local update after a replanning event. The initial path followed is shown in yellow and the new path followed in magenta (only those are shown for clarity purposes).

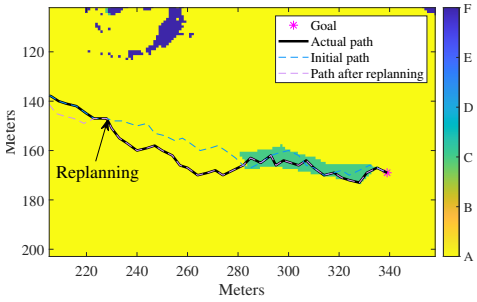


Fig. 6. Actual path with locally updated velocity categories and replanning with the two paths combined in the traverse.

terrain along the path. With several offline planned options at its disposal, it becomes easier for the rover to reroute after a significant update is made (e.g., change of velocity category). Future work includes research on integration of lower-level motion planning algorithms and focus on online risk assessment and decision making under unexpected situations. Moreover, the authors are working on the development of an algorithm that implements direct measurements (i.e., exact information gathering through deployment of an instrument) in addition to the noisy monitoring mentioned in this paper, as well as a method to obtain this exact terrain information. Lastly, these results will be tested on a physical robot in a Mars analog environment, at which point the "truth" map would be replaced by actual monitoring of the terrain.

Appendix A
Edge Potential Matrices

Tables A1 through A6 shows the edge-potential matrices. Rows and columns of zeros are not shown.

TABLE A1
CFA $\leq 7\%$ AND SLOPE $\leq 10°$.

		Prediction			
		A	C	D	F
Truth	A	84.0%	7.50%	.200%	0%
	C	10.5%	81.3%	1.20%	.200%
	D	5.20%	10.6%	98.4%	0%
	F	.300%	.600%	.200%	99.8%

TABLE A2
CFA $\leq 7\%$ AND $10° <$ SLOPE $\leq 15°$.

		Prediction		
		A	C	F
Truth	A	84.0%	7.50%	.0872%
	C	10.5%	83.3%	.636%
	F	2.44%	4.96%	99.2%

TABLE A3
CFA $\leq 7\%$ AND $15° <$ SLOPE $\leq 20°$.

		Prediction		
		B	C	F
Truth	B	84.0%	7.50%	.0872%
	C	10.5%	81.3%	.636%
	F	2.44%	4.96%	99.2%

TABLE A4
CFA $\leq 7\%$ AND $20° <$ SLOPE $\leq 25°$; AND FOR $7\% <$ CFA $\leq 15\%$ AND $20° <$ SLOPE $\leq 25°$. THE HIGHLIGHTED CELL COMES FROM EQ. 6.

		Prediction	
		E	F
Truth	E	82.8%	.340%
	F	8.31%	99.2%

TABLE A5
$7\% <$ CFA $\leq 15\%$ AND SLOPE $\leq 10°$.

		Prediction			
		B	C	D	F
Truth	B	84.0%	7.50%	0%	.200%
	C	10.5%	81.3%	0.200%	1.20%
	D	.300%	.600%	99.8%	.200%
	F	5.20%	10.6%	0%	98.4%

TABLE A6
$7\% <$ CFA $\leq 15\%$ AND $10° <$ SLOPE $\leq 15°$; AND FOR $7\% <$ CFA $\leq 15\%$ AND $15° <$ SLOPE $\leq 20°$.

		Prediction		
		B	C	F
Truth	B	84.0%	7.50%	.0872%
	C	10.5%	81.3%	.636%
	F	2.44%	4.96%	99.2%

For the cases CFA $\leq 7\%$ with slope $> 25°$, $7\% <$ CFA $\leq 15\%$ with slope $> 25°$ and CFA $> 15\%$ with any slope, category F is the only category possible.

Acknowledgment

The authors thank J. Beard, J. Strader, C. Kilic, J. Gross for their help. This research is supported by NASA EPSCoR Research Cooperative Agreement WV-80NSSC17M0053, B. Statler Fellowship, NSF GRFP Grant DGE-1102689, and NASA West Virginia Space Grant Consortium.

References

[1] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansen, W. A. Delamere, J. A. Grant, V. C. Gulick, K. E. Herkenhoff, L. Keszthelyi *et al.*, "Mars reconnaissance orbiter's high resolution imaging science experiment (hirise)," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

[2] D. Gaines, R. Anderson, G. Doran, W. Huffman, H. Justice, R. Mackey, G. Rabideau, A. Vasavada, V. Verma, T. Estlin *et al.*, "Productivity challenges for mars rover operations," 2016.

[3] J. B. Johnson, A. V. Kulchitsky, P. Duvoy, K. Iagnemma, C. Senatore, R. E. Arvidson, and J. Moore, "Discrete element method simulations of mars exploration rover wheel performance," *Journal of Terramechanics*, vol. 62, pp. 31–40, 2015.

[4] R. E. Arvidson, J. W. Ashley, J. Bell, M. Chojnacki, J. Cohen, T. Economou, W. H. Farrand, R. Fergason, I. Fleischer, P. Geissler *et al.*, "Opportunity mars rover mission: Overview and selected results from purgatory ripple to traverses to endeavour crater," *Journal of Geophysical Research: Planets*, vol. 116, no. E7, 2011.

[5] R. Arvidson, P. DeGrosse Jr, J. Grotzinger, M. Heverly, J. Shechet, S. Moreland, M. Newby, N. Stein, A. Steffy, F. Zhou *et al.*, "Relating geologic units and mobility system kinematics contributing to curiosity wheel damage at gale crater, mars," *Journal of Terramechanics*, vol. 73, pp. 73–93, 2017.

[6] M. Bajracharya, M. Maimone, and D. Helmick, "Autonomy for mars rovers: Past, present, and future," *Computer*, vol. 41, no. 12, pp. 44–50, 2008.

[7] The new york time - mars curiosity rover tracker. [Online]. Available: https://archive.nytimes.com/www.nytimes.com/interactive/science/space/mars-curiosity-rover-tracker.html#sol665

[8] C. Schroeder, "Life on mars: my 15 amazing years with oppy, nasa's record-breaking rover," 2019.

[9] J. Biesiadecki, C. Leger, and M. Maimone, "Tradeoffs between directed and autonomous driving on the mars exploration rovers," *The Internat. Journal of Robotics Research*, vol. 26, no. 1, pp. 91–104, 2007.

[10] MEPAG, "Science priorities for mars sample return," 2008.

[11] M. Golombek, R. Otero, M. Heverly, M. Ono, K. Williford, B. Rothrock, S. Milkovich, E. Almeida, F. Calef, N. Williams *et al.*, "Characterization of mars rover 2020 prospective landing sites leading up to the second downselection," in *Lunar and Planetary Science Conference*, vol. 48, 2017.

[12] R. Lorenz, E. Turtle, J. Barnes, M. Trainer, D. Adams, K. Hibbard, C. Sheldon, K. Zacny, P. Peplowski, D. Lawrence *et al.*, "Dragonfly: a rotorcraft lander concept for scientific exploration at titan," *Johns Hopkins APL Technical Digest*, vol. 34, pp. 374–387, 2018.

[13] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global path planning on board the mars exploration rovers," in *2007 IEEE Aerospace Conference*. IEEE, 2007, pp. 1–11.

[14] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field d* algorithm," *Journal of Field Robotics*, vol. 23, no. 2, pp. 79–101, 2006.

[15] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, no. 2, pp. 103–122, 2006.

[16] G. Reina, A. Milella, and R. Galati, "Terrain assessment for precision agriculture using vehicle dynamic modelling," *Biosystems engineering*, vol. 162, pp. 124–139, 2017.

[17] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environmental monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.

[18] A. Candela, D. Thompson, E. N. Dobrea, and D. Wettergreen, "Planetary robotic exploration driven by science hypotheses for geologic mapping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3811–3818.

[19] R. Gellert, J. Campbell, P. King, L. Leshin, G. Lugmair, J. Spray, S. Squyres, and A. Yen, "The alpha-particle-x-ray-spectrometer (apxs) for the mars science laboratory (msl) rover mission," in *The 40th Lunar and Planetary Science Conference*, vol. 1468, 2009, p. 2364.

[20] L. A. Rahmatian and P. T. Metzger, "Soil test apparatus for lunar surfaces," in *Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments*, 2010, pp. 239–253.

[21] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[22] F. Mascarich, C. Papachristos, T. Wilson, and K. Alexis, "Distributed radiation field estimation and informative path planning for nuclear environment characterization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2318–2324.

[23] M. Ono, B. Rothrock, E. Almeida, A. Ansar, R. Otero, A. Huertas, and M. Heverly, "Data-driven surface traversability analysis for mars 2020 landing site selection," in *2016 IEEE Aerospace Conference*. IEEE, 2016, pp. 1–12.

[24] R. E. Arvidson, "Roving on mars with opportunity and curiosity: terramechanics and terrain properties," in *Earth and Space 2014*, 2014, pp. 165–173.

[25] M. G. Bekker, "Introduction to terrain-vehicle systems.part i:the terrain.part ii:the vehicle," Michigan Univ. Ann Arbor, Tech. Rep., 1969.

[26] J. Wong *et al.*, *Terramechanics and off-road vehicles*. Elsevier, 1989.

[27] D. Helmick, A. Angelova, L. Matthies, C. Brooks, I. Halatci, S. Dubowsky, and K. Iagnemma, "Experimental results from a terrain adaptive navigation system for planetary rovers," in *9th Internat. Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS*, 2008.

[28] H. Inotsume, T. Kubota, and D. Wettergreen, "Robust path planning for slope traversing under uncertainty in slip prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3390–3397, 2020.

[29] F. Zhou, R. E. Arvidson, K. Bennett, B. Trease, R. Lindemann, P. Bellutta, K. Iagnemma, and C. Senatore, "Simulations of mars rover traverses," *Journal of Field Robotics*, vol. 31, no. 1, pp. 141–160, 2014.

[30] F. Zhou, R. Arvidson, and A. Zastrow, "Simulating mars science laboratory curiosity rover traverses using artemis," in *Lunar and Planetary Science Conference*, vol. 48, 2017.

[31] C. Senatore, N. Stein, F. Zhou, K. Bennett, R. Arvidson, B. Trease, R. Lindemann, P. Bellutta, M. Heverly, and K. Iagnemma, "Modeling and validation of mobility characteristics of the mars science laboratory curiosity rover," in *Proc. of the 12th Internat. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2014, pp. 17–19.

[32] M. Golombek and D. Rapp, "Size-frequency distributions of rocks on mars and earth analog sites: Implications for future landed missions," *Journal of Geophysical Research:Planets*, vol. 102, no. E2, pp. 4117–4129, 1997.

[33] M. Heverly, J. Matthews, J. Lin, D. Fuller, M. Maimone, J. Biesiadecki, and J. Leichty, "Traverse performance characterization for the mars science laboratory rover," *Journal of Field Robotics*, vol. 30, no. 6, pp. 835–846, 2013.

[34] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono, "Spoc: Deep learning-based terrain classification for mars rover missions," in *AIAA SPACE 2016*, 2016, p. 5539.

[35] M. Ono *et al.*, "Mars 2020 surface mission performance analysis: Part 2. surface traversability," in *AIAA Space Forum*, 2018, p. 5418.

[36] J. R. Sánchez-Ibánez, C. J. Pérez-del Pulgar, M. Azkarate, L. Gerdes, and A. García-Cerezo, "Dynamic path planning for reconfigurable rovers using a multi-layered grid," *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 32–42, 2019.

[37] A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, and J. Zheng, "The lander vision system for mars 2020 entry descent and landing," 2017.

[38] G. Rabideau and E. Benowitz, "Prototyping an onboard scheduler for the mars 2020 rover," 2017.

[39] C. Kilic, J. N. Gross, N. Ohi, R. Watson, J. Strader, T. Swiger, S. Harper, and Y. Gu, "Improved planetary rover inertial navigation and wheel odometry performance through periodic use of zero-type constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 552–559.

[40] G. S. Collins, H. J. Melosh, and G. R. Osinski, "The impact-cratering process," *Elements*, vol. 8, no. 1, pp. 25–30, 2012.

[41] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Unknown Publisher, 1994, vol. 1994.

[42] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," Carnegie-Mellon Univ., Tech. Rep., 1993.

[43] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of Applied Mathematics*, vol. 27, no. 4, pp. 526–530, 1970.

[44] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[45] M. Schmidt, "Ugm: Matlab code for undirected graphical models," *URL http://www. di. ens. fr/mschmidt/Software/UGM. html*, 2012.