A Risk-Constrained Multi-Stage Decision Making Approach to the Architectural Analysis of Planetary Missions

Yoshiaki Kuwata, Marco Pavone, and J. (Bob) Balaram

Abstract— **This paper presents a novel risk-constrained multi-stage decision making approach to the architectural analysis of planetary rover missions. In particular, focusing on a 2018 Mars rover concept, which was considered as part of a potential Mars Sample Return campaign, we model the entry, descent, and landing (EDL) phase and the rover traverse phase as four sequential decision-making stages. The problem is to find a sequence of divert and driving maneuvers so that the rover drive is minimized and the probability of a mission failure (e.g., due to a failed landing) is below a user-specified bound. By solving this problem for several different values of the model parameters (e.g., divert authority), this approach enables rigorous, accurate and systematic trade-offs for the EDL system vs. the mobility system, and, more in general, cross-domain trade-offs for the different phases of a space mission. The overall optimization problem can be seen as a chance-constrained dynamic programming problem, with the additional complexity that 1) in some stages the disturbances do not have any probabilistic characterization, and 2) the state space is extremely large (i.e, hundreds of millions of states for trade-offs with high-resolution Martian maps). To this purpose, we solve the problem by performing an unconventional combination of average and minimax cost analysis and by leveraging high efficient computation tools from the image processing community. Preliminary trade-off results are presented.**

I. INTRODUCTION

Future planetary missions, such as those involving any potential Mars Sample Return (MSR), would be expected to employ rovers to reach scientifically interesting sites after landing. In order to minimize the risk of mission failure, it is critical to land the rover in a place that is safe and in close proximity to the science targets.

Traditionally, the entry, descent, and landing (EDL) problem (e.g., where to place the landing ellipse) [1] and the mobility problem (e.g., how to drive to scientifically interesting sites, and how long it would take) are studied independently, and only in the late development phase is the overall performance of the integrated system analyzed [2]. Without being able to perform system-level trade-offs or know the overall system performance at an earlier phase, such approach could lead to highly suboptimal design decisions [3].

Y. Kuwata and B. Balaram are with Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA. Email: {Yoshiaki.Kuwata, J.Balaram}@jpl.nasa.gov

M. Pavone is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA. Email: pavone@stanford.edu.

Government sponsorship acknowledged.

The goal of this paper is to develop a systematic approach to perform quantitative mission-level trade-offs of improved EDL systems vs. improved mobility systems, and, more in general, cross-domain trade-offs for the different phases of a planetary rover mission. Given a fixed funding to invest in various technologies, and given a fixed landing mass that the spacecraft can carry, is it better to carry more fuel to land the rover closer to the target, or design a better and potentially heavier mobility system that can go over larger rocks or go around hazards? In order to answer such questions, it is important to address key couplings between different domains. What makes this problem even more challenging is that landing a rover on a planet or a moon is not a deterministic operation due to various sources of uncertainties, such as dispersions in the entry states (position and velocity), initial attitude uncertainty, vehicle aerodynamics, navigation error build-up from inertial sensor noise, and variability in atmospheric density and winds.

In the proposed approach, we model the EDL phase and the rover traverse phase as four sequential decision making stages. The problem is to find a sequence of divert and driving maneuvers so that the rover drive is minimized and the probability of a mission failure (e.g., due to a failed landing) is below a user-specified bound. By solving this problem for several different values of the model parameters (e.g., divert authority), this approach enables rigorous, accurate and systematic trade-offs for the EDL system vs. the mobility system. The optimization problem can be seen as a *chance-constrained* dynamic programming problem (see, e.g., [4] and [5] for the closely related problem of chance-constrained model predictive control), with the additional complexity that 1) in some stages the disturbances do not have any probabilistic characterization, and 2) the state space is extremely large (i.e, hundreds of millions of states for trade-offs with high-resolution Martian maps). To this purpose, we solve the problem by performing an unconventional combination of average and minimax cost analysis and by leveraging high efficient computation tools from the image processing community, such as morphological dilation and erosion. By using this approach, we accurately characterize the inherent trade-offs between, e.g., the power descent guidance control authority and the hazard detection avoidance control authority, or between spacecraft divert and rover drive distance.

In summary, the contribution of this paper is threefold. First, we present a novel risk-constrained multi-stage decision making approach to the architectural analysis of planetary rover missions. This approach can be seen as a multi-stage development of authors' previous work [6], where the objective was to compute trade-offs between the location of the landing ellipse and rover's driving capabilities. Second, we show how this approach can be made computational efficient by using image processing techniques. Third, we compute several trade-offs instrumental to the development of the mission architecture for a 2018 Mars rover concept.

The rest of the paper is organized as follows. Section II presents a more detailed formulation of the problem and description of the approach. Section III presents the algorithmic flow to solve the problem, and Section IV presents several trade-offs relevant to the 2018 Mars rover concept. Finally, in Section V, we draw our conclusions and discuss directions for future research.

II. System Modeling

In this section we first describe the sequence of control actions during the terminal phase of a mission to Mars; then, we consider the problem of finding a sequence of divert and driving maneuvers so that the rover drive is minimized and the probability of a mission failure is below a user-specified bound. Finally, we show how the results of such optimization problem can be used to perform an architectural analysis for a mission to Mars (or to other planetary bodies).

A. Mission Sequence

Figure 1 shows the mission sequence of the aforementioned 2018 Mars rover concept. Assume for now that the nominal landing target location has been selected. During the inter-planetary guidance phase, several trajectory correction maneuvers (TCMs) are performed, which guides the spacecraft to enter the Martian atmosphere with the planned pose and velocity. During the entry phase, the spacecraft controls its bank angle to maintain its course to the target. Then, it jettisons the heat shield and deploys the parachute. During the parachute descend, the spacecraft drifts significantly due to wind. Once it reaches an ignition altitude, the spacecraft ignites thrusters to initiate the terminal descent phase, which brings the velocity down to zero at the surface.

Recent efforts [7] have looked into a way to divert the spacecraft closer to the final target during the terminal descent phase. In this paper, we refer to this technology as Powered Descent Guidance (PDG). The two primary purposes of the PDG are to bring the lander closer to the target and to avoid landing hazards. The PDG divert makes a decisions using a map generated offline and stored onboard. For missions to Mars, images taken from the HiRISE (High Resolution Imaging Science Experiment) camera on the Mars Reconnaissance Orbiter can provide Digital Elevation Maps (DEMs) at about 1m resolution [8].

Near the final stage of landing, on-board sensors start to detect small hazards that could not be detected from the orbital imageries. The lander can make a final divert to avoid these hazards. This stage is called Hazard Detection and Avoidance (HDA) [9].

Throughout the entire EDL phase, the spacecraft is subject to uncertain disturbances, such as execution errors. In Figure 1, the planned trajectory at each stage of the EDL phase is represented with a solid line, and the actual trajectory is represented with a dashed line.

Once the vehicle has landed, the rover starts driving to a science target. Depending on the landing location, the best science target is selected from a set of candidate targets.

B. Multi-stage Decision Making Formulation

In this section, we aim to simultaneously choose a science target and design an EDL maneuver so as to minimize the rover drive distance while ensuring a probability of mission failure below a given threshold. One could model the detailed vehicle dynamics and optimize the decisions over the entire mission in continuous time, from atmospheric entry to reaching a science target. Such formulation however would lead to a prohibitively complex trajectory optimization problem. Instead, we abstract the entire EDL phase into a small number of *key* sequential stages. More specifically, we subdivide the mission sequence presented in Figure 1 into four stages: placement of ignition target, placement of PDG target, placement of HDA target, and rover path planning to the nearest science target.

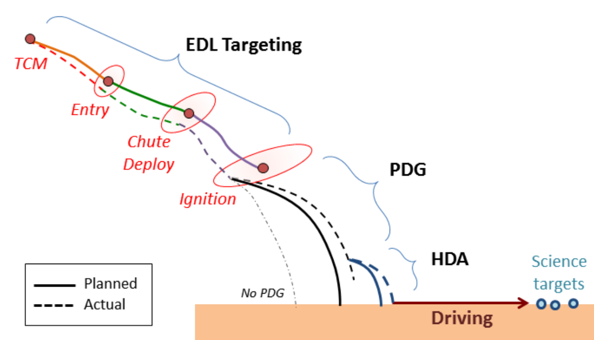


Fig. 1. Multi-stage model for the EDL phase of 2018 Mars rover concept.

Accordingly, the system evolves according to the following discrete-time equation, where $k = 0, 1, 2, 3$:

$$x_{k+1} = u_k + w_k, \tag{1}$$
$$u_k \in U_k(x_k), \tag{2}$$
$$w_k \sim P_k. \tag{3}$$

At each stage k, x_k denotes the state (a 2D position) and u_k denotes the control, which is constrained to be in a set $U_k(x_k)$. In our model, after each control action is exerted, a disturbance w_k drives the system away from the nominal trajectory, as shown in Figure 1. The uncertain disturbance w_k is modeled as additive, and follows a probability distribution denoted by P_k. The distribution P_k is assumed to have compact support, denoted by W_k. Note that, according to (3), the disturbance is independent of x_k and u_k. (The extension to the general case is possible, but leads to a deterioration in computation time.) The following subsections specify state, control, and disturbances at each stage, and also the stage and terminal costs. In the following, let $\mathcal{M} \subset \mathbb{R}^2$ be a (rectified) map of a landing site on Mars.

1) Initial Entry to Ignition ($k = 0$): The state x_0 is the position of the spacecraft before the entrance into the Mars atmosphere, and is assumed to be given. The control $u_0 \in \mathcal{M}$ corresponds to the placement of the chosen ignition target. The disturbance w_0 mainly models the effect of winds during the parachute descent phase, although it could include other uncertainty sources such as IMU errors and attitude initialization errors of the entry state.

2) PDG ($k = 1$): The state $x_1 \in \mathcal{M}$ is the position of the spacecraft at an altitude right before the ignition of the rockets for the PDG phase. The control $u_1 \in \mathbb{R}^2$ corresponds to the placement of the chosen PDG target, or, equivalently, to the foreseen landing site assuming nominal behavior (i.e., in absence of subsequent control actions and disturbances). The control u_1 is constrained to take values in a non-empty set $U_1(x_1) \subset \mathcal{M}$, which models the limited divert maneuver capability. The disturbance w_1 models the execution error of the PDG divert maneuver.

3) HDA ($k = 2$): The state $x_2 \in \mathcal{M}$ is the position of the spacecraft at an altitude right before the ignition of the rockets for the HDA phase. The control $u_2 \in \mathbb{R}^2$ corresponds to the placement of the chosen HDA target or, equivalently, to the foreseen landing site assuming nominal behavior (i.e., in absence of subsequent control actions and disturbances), and is constrained to take values in a non-empty set $U_2(x_2) \subset \mathcal{M}$.

When the vehicle lands on the surface, the mission could fail due to landing hazards $X_{\text{HazLander}} \subset \mathcal{M}$. We model this situation with an additional state F, and the dynamics (1) for $k = 2$ are replaced with the following

$$x_{k+1} = \begin{cases} F & \text{if } u_k + w_k \in X_{\text{HazLander}} \\ u_k + w_k & \text{otherwise.} \end{cases} \tag{4}$$

4) Driving ($k = 3$): The state $x_3 \in \mathcal{M}$ is the landed position. The control $u_3 \in \mathbb{R}^2$ corresponds to the location of the science target that the rover will visit. The constraint $u_3 \in U_3(x_3)$ models the maximum drive distance constraint, and given a set of science target regions $X_S \in \mathcal{M}$, we have $U_3(x_3) \subset X_S$.

There are three cases where this stage results in a failure state F: the vehicle has already failed due to a catastrophic landing in stage $k = 2$; no feasible path exists from its landing position x_k to the desired science target u_k; and the rover fails to reach any science target due to a mobility failure, modeled with w_k. For $k = 3$, (1) is replaced with

$$\begin{aligned} x_{k+1} &= \begin{cases} F & \text{if } \begin{cases} x_k = F, \\ x_k \neq F \text{ and } l(x_k, u_k) = +\infty, \text{ or} \\ x_k \neq F \text{ and } u_k + w_k \notin X_S \end{cases} \\ u_k + w_k & \text{otherwise.} \end{cases} \\ &\triangleq f_3(x_3, u_3, w_3) \end{aligned} \tag{5}$$

where the function $l(x_k, u_k)$ returns the shortest path length from x_k to u_k that avoids rover hazards, denoted by $X_{\text{HazRover}} \subset \mathcal{M}$, and returns $+\infty$ if there is no feasible path from x_k to u_k. The function $l(x_k, u_k)$ is computed via the Dijkstra's algorithm that starts from all states in X_S. For the case of visiting $n(> 1)$ science targets, $l(x_k, u_k)$ returns the path length of the Prize Collecting Traveling Salesman Problem (PCTSP) [10], where the TSP tour starts from x_k and ends at u_k while visiting a subset of the target regions in X_S. Indeed, the analysis and results presented in this paper consider the case where w_3 is identically equal to zero, i.e., we neglect rover's failures (the techniques presented in this paper easily extend to this case, which will be the subject of a future trade-off analysis).

The terminal state $x_4 \in \mathcal{M}$ is the final location of the rover.

Note that once the system enters the failure state F, it will remain F throughout, i.e., if $x_{k_0} = F$ for some k_0, then $x_k = F \quad \forall k \geq k_0$.

5) Cost: We define the terminal cost as

$$g_4(x_4) = \begin{cases} +\infty & \text{if } x_4 = F, \\ 0 & \text{if } x_4 \neq F. \end{cases} \tag{6}$$

where the failure state corresponds to the infinite penalty.

Because the goal is to deliver the lander as close to the science target as possible, the stage cost exists only for driving, as defined as

$$g_3(x_3, u_3) = \begin{cases} 0 & \text{if } x_3 = F, \\ l(x_3, u_3) & \text{otherwise.} \end{cases} \tag{7}$$

In this paper, we define the traversability cost as the length of the path going from the landing position x_3 to the science target u_3 while avoiding rover hazards X_{HazRover}.

C. Optimization Problem

The mission is considered to be a failure if

$$x_4 = F. \tag{8}$$

Since a mission designer desires a low failure probability, we consider the following risk constraint:

$$\mathbb{P}(x_4 = F) \leq \epsilon, \tag{9}$$

where $\mathbb{P}(\cdot)$ denotes the probability of the event, and ϵ is a user-specified risk threshold.

We consider the class of feedback-control policies consisting of a sequence of functions:

$$\pi = \{\mu_0, \mu_1, \mu_2, \mu_3\}, \tag{10}$$

where μ_k maps states x_k into controls $u_k = \mu_k(x_k)$ and such that $\mu_k(x_k) \in U_k(x_k)$ for all x_k.

Then, in principle, one would like to solve the following optimization problem:

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}\Big\{g_3(x_3, u_3) + g_4(x_4)\Big\} \\ \text{s.t.} \quad & \mathbb{P}(x_4 = F) \leq \epsilon \end{aligned} \tag{11}$$

where the operation $\mathbb{E}\{\cdot\}$ takes an expected value with respect to the uncertainties w_k's represented in (3). This problem can be seen as a *chance-constrained dynamic programming problem*, which could be solved (at least approximately) by using Lagrangian methods.

However, within the context of EDL mission analysis, one has to face the challenge that while the disturbance during the entry phase (i.e., w_0) is fairly well modeled, the subsequent disturbances w_1 and w_2 that are associated with the new technologies PDG and HDA are, currently, not well characterized and no probability distribution function is available for them. Therefore, we consider the following combination of average and minimax cost analysis. Let

$$J_1^*(x_1) \triangleq \min_{\mu_1,\,\mu_2,\,\mu_3} \max_{w_1\in W_1,\,w_2\in W_2} \Big[g_3(x_3,u_3)+g_4(x_4)\Big].$$

The function $J_1^*(x_1)$ is the optimal worst-case cost to reach a designated scientific target (recall that we are assuming that w_3 is identically equal to zero). Clearly, for some states x_1, $J_1^*(x_1)=+\infty$; accordingly, we define a failure set:

$$\mathcal{F}_1 \triangleq \{x_1\in\mathcal{M}\mid J_1^*(x_1)=+\infty\},$$

in other words, $\mathcal{F}_1$ contains the set of states at stage 1 that will end up (in the worst case) into a mission failure. For a *fixed* set of modeling parameters (e.g., slope tolerance for the rover or divert authority for the PDG), we then consider the following optimization problem:

Optimization problem $\mathcal{OPT}$: let U_0^s be the set of *safe* controls at stage 0, defined as

$$U_0^s \triangleq \big\{u_0\in U_0 \,\big|\, \mathbb{P}(x_1\in\mathcal{F}_1)<\epsilon\big\}.$$

Then, given all model parameters and a given risk bound ϵ, the problem is to solve

$$J_0^*(x_0) \triangleq \min_{u_0\in U_0^s} \mathbb{E}\Big\{J_1^*(x_1)\,\Big|\, x_1\notin\mathcal{F}_1\Big\}. \quad (12)$$

If set U_0^s is empty, then we say that the problem is *infeasible* for the given risk threshold ϵ.

Note that in problem $\mathcal{OPT}$ we consider a conditional expectation in order to optimize only on the non-failure states. Roughly, Problem $\mathcal{OPT}$ finds the smallest expected (over pre-ignition disturbances) worst-case cost to reach a scientific target. The following straightforward lemma characterizes the probability that $x_4=F$.

Lemma 1: Given an optimal control policy determined by solving problem $\mathcal{OPT}$, the probability that at stage 4 the state is a failure state (i.e., $x_4=F$) is: $\mathbb{P}(x_4=F)<\epsilon$.

Proof: This is a trivial consequence of the law of total probability. ■

D. Trade-off Analysis

The overall approach consists in solving a family of optimization problems $\mathcal{OPT}$, one for each combination of model's parameters (e.g., divert capabilities, slope tolerance for the rover, etc.). Imperative to this approach is a fast solution of problem $\mathcal{OPT}$, which entails possibly hundreds of millions of states: solution techniques leveraging image processing algorithms are presented in Section III. A preliminary application of this approach to the trade-off analysis for the 2018 Mars rover concept is presented in Section IV.

III. Algorithm

The optimization problem $\mathcal{OPT}$ is solved in two steps: in the first step we compute, by using backward recursion, the function $J_1^*(x_1)$; in the second step we perform the minimization of the conditional expected value. As described below, even though the state space is extremely large (i.e, hundreds of *millions* of states for trade-offs with high-resolution Martian maps), each step can be performed in a reasonable amount of time by using highly-optimized techniques from the image processing community, which collectively lead to *substantial* speedups.

A. Step 1: computation of $J_1^*(x_1)$

As stated above, the computation of $J_1^*(x_1)$ proceeds by backward recursion (this is, in fact, a standard dynamic programming problem).

1) Stage 4, terminal stage:

$$J_4^*(x_4) \triangleq g_4(x_4). \quad (13)$$

2) Stage 3, rover driving: At stage $k=3$,

$$J_3^*(x_3) \triangleq \min_{u_3\in U_3(x_3)} \Big[g_3(x_3,u_3)+J_4^*(f_3(x_3,u_3,0))\Big]. \quad (14)$$

Cost function $J_3^*(x_3)$ embeds the information of traversability cost and mission failure (that has cost $+\infty$).

3) Stage 2, HDA: In this step, we solve the equation

$$J_2^*(x_2) \triangleq \min_{u_2\in U_2(x_2)} \max_{w_2\in W_2} J_3^*(x_3). \quad (15)$$

This step is computationally intensive and we use image processing techniques for its solution. Specifically, we first carry out a *dilation* on the terminal cost map $J_3^*(\cdot)$ with a "mask" $I_{W_2}(w)$, where $I_{W_2}(w)$ is the indicator function that takes value one if $w\in W_2$ and zero otherwise. The dilated map, called $J_3^{\mathrm{D}}(\cdot)$, represents for each u_2 the worst-case cost that would ensue. This procedure is possible since the bounds on the disturbance do not depend on x_2 or u_2, and the underlying state space is a gridded square map. Then, we find the minimizing controller by performing an erosion on the dilated map $J_3^{\mathrm{D}}(\cdot)$. The dilation (the "max" operation) accounts for errors in the HDA stage. The erosion (the "min" operation) accounts for corrections made during the HDA stage. Both dilation and erosion are key functions in image processing, and efficient algorithms exist for them.

4) Stage 1, PDG: In this step, we solve the problem

$$J_1^*(x_1) \triangleq \min_{u_1\in U_1(x_1)} \max_{w_1\in W_1} J_2^*(x_2), \quad (16)$$

which has the same form of (15) and is solved by using the same sequence of dilation and erosion operations.

B. Step 2: computation of $J_0^*(x_0)$

Given the knowledge of $J_1^*(x_1)$, this step minimizes the conditional expected value in equation (12).

This step is, again, computationally intensive and we use image processing techniques for its solution. Specifically, define the "success probability map" $\mathcal{M}^{\text{succ}}$ according to the assignment: $\mathcal{M}^{\text{succ}}(u_0) = \mathbb{P}(x_1 \notin \mathcal{F}_1 | u_0)$ for all $u_0 \in \mathcal{M}$. The map $\mathcal{M}^{\text{succ}}$ can be efficiently computed by *cross-correlating* the distribution of w_0 (that has compact support and is viewed here as a "convolution mask") with a boolean map that takes value 1 at cell x_1 if $x_1 \notin \mathcal{F}_1$ (i.e., $J_1^*(x_1) < +\infty$) and value 0 otherwise (for all $x_1 \in \mathcal{M}$). The cross-correlation is implemented by using the Fourier transform, which leads to dramatic speedups. Given the map $\mathcal{M}^{\text{succ}}$, one can easily compute the conditional distribution

$$\mathbb{P}(w_0 \mid x_1 \notin \mathcal{F}_1, u_0) = \frac{\mathbb{P}(w_0)}{\mathcal{M}^{\text{succ}}(u_0)},$$

which is used to take the conditional expectation in (12). Define the "cost map" $\mathcal{M}^{\text{cost}}$ according to the assignment: $\mathcal{M}^{\text{cost}}(u_0) = \mathbb{E}\Big\{ J_1^*(x_1) \mid x_1 \notin \mathcal{F}_1 \Big\}$ if $u_0 \in U_0^s$ (where the expectation is with respect to the distribution $\mathbb{P}(w_0 \mid x_1 \notin \mathcal{F}_1, u_0)$) and $\mathcal{M}^{\text{cost}}(u_0) = +\infty$ otherwise. The map $\mathcal{M}^{\text{cost}}$ can be efficiently computed by 1) *cross-correlating* $J_1^*(x_1)$ (where the $+\infty$ cells are set to zero since they need to be excluded given the conditioning) with the distribution of w_0 (that, as before, is viewed as a "convolution mask"), and 2) normalizing the resulting map through an entry-by-entry division with $\mathcal{M}^{\text{succ}}$ with the rule that if $\mathcal{M}^{\text{succ}}(u_0) < 1 - \epsilon$, the result of the division is set equal to $+\infty$. Finally, one performs the minimization in (12) by exhaustively searching $\mathcal{M}^{\text{cost}}$ over all feasible values of u_0 (i.e., all values in U_0^s).

Computation times are discussed in detail in Section IV.

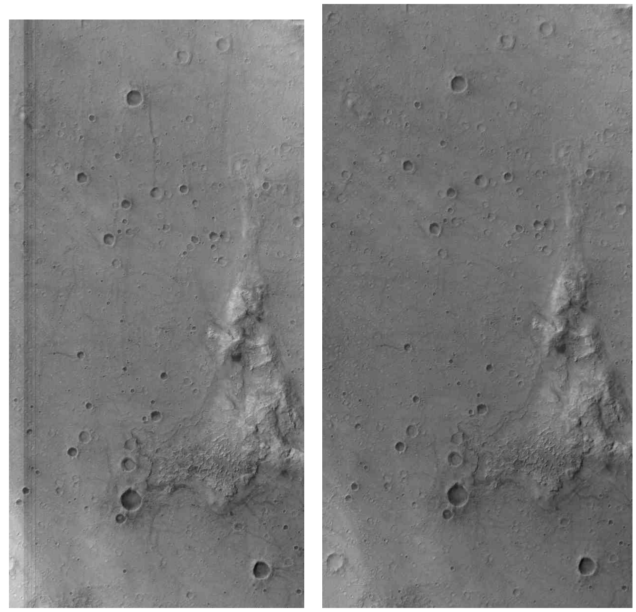
IV. Numerical Examples

A. Algorithmic Steps

This subsection illustrates the algorithmic steps in Section III using a rectangular grid-map $\mathcal{M}$ of size 3000×3000.

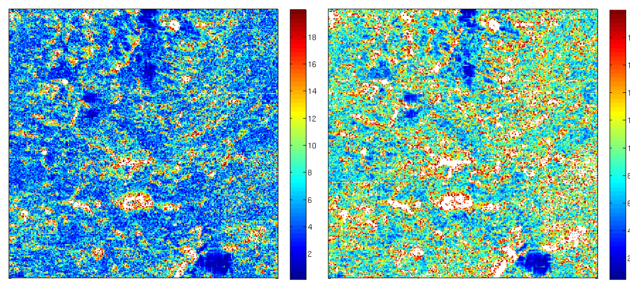
1) Input Maps and Lists: We consider two types of hazards in this paper: local slopes and rocks. The slope map is defined over $\mathcal{M}$, and each cell has a value equal to the slope of the surface at its corresponding position. All the rocks are modeled as circles and they are represented as a set of center positions and radii within $\mathcal{M}$ [11]. Other types of hazards such as ripples and scarps can be provided as additional maps. Figure 2 shows a pair of orbital images of Mars taken with the HiRISE camera onboard the Mars Reconnaissance Orbiter, from which our input terrain data such as slope maps, rock lists, and ripple maps are obtained through computer vision pipelines. The examples in Section IV-A and Section IV-B use a terrain data in Ebersewalde and East Margaritifer, respectively.

2) Derived Products: The hazard sets X_{HazRover} and $X_{\text{HazLander}}$ are a combination of rocks beyond a certain size and excessive slopes. Different thresholds are used for the rover and the lander. In order to treat the vehicle as a point, the slope map is dilated by the vehicle radius. The rock radius is also enlarged by the vehicle radius, and then the rocks are rasterized onto the grid map $\mathcal{M}$. Each cell of the hazard map has value $+\infty$ if the local slope exceeds the design tolerance of the vehicle or if a rock larger than the design tolerance touches the cell.



(a) Left image (b) Trimmed right image

Fig. 2. A pair of HiRISE stereo images. Note that the right image is taken with a slanted view.



(a) Rover hazard map (b) Lander hazard map

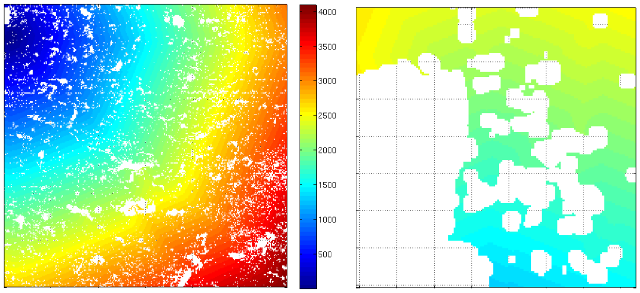
Fig. 3. Hazard maps of size 3km-by-3km for a rover and a lander. The original hazard map is dilated by the radius of the vehicle. The color represents the local slope, and the infeasible cells are left blank white.

Figure 3 shows the hazard maps for the rover and the lander. The color of each cell corresponds to the slope value, and the infeasible cells are marked with blank white in the figures. The lander has a tighter tolerance for hazards, and hence larger blank regions.

3) Stage 3: This example has a target region X_S near the upper left corner of the figure. Figure 4 shows the cost map $J_3^*(x_3)$ obtained by running Dijkstra's algorithm from each cell in X_S and taking the minimum (i.e., the closest target). The positions corresponding to $J_3^*(x_3) = +\infty$ are shown in blank white. For Figure 4–8, the figures on the left with caption (a) show the 3km-by-3km map, and the figures on the right with caption (b) show the zoom-in of a smaller region to better illustrate how the cost map evolves in the subsequent computations.

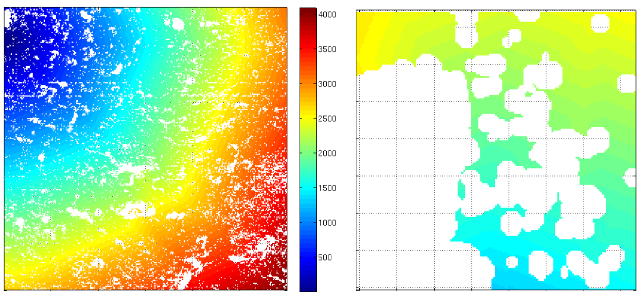
4) Stage 2: Figures 5 and 6 show the cost map for stage 2 (HDA). From $J_3^*(x_3)$, we first perform dilation to obtain $J_3^D(x_3)$, then erosion to obtain $J_2^*(x_2)$.

5) Stage 1: The stage 1 (PDG) goes through the same sequence as the stage 2, and the results are shown in



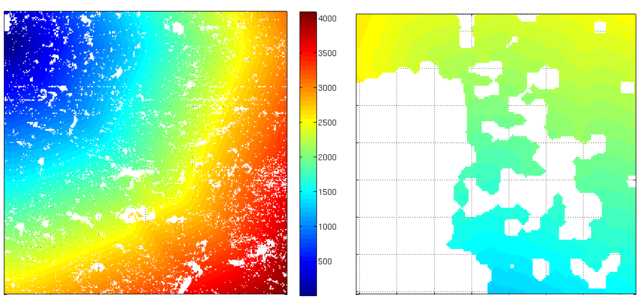
(a) 3km-by-3km map (b) Zoom-in of a 150×150 patch

Fig. 4. The cost map $J_3^*(x_3)$. The color bar indicates the cost value.



(a) 3km-by-3km map (b) Zoom-in of a 150×150 patch

Fig. 5. The cost map $J_3^D(x_3)$, after the dilation for 1m HDA error



(a) 3km-by-3km map (b) Zoom-in of a 150×150 patch

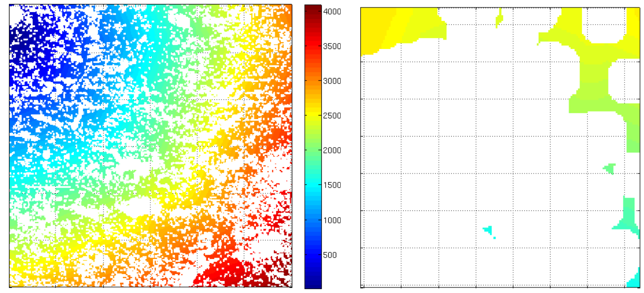
Fig. 6. $J_2^*(x_2)$ after the erosion for 2m HDA corrections

Figures 7 and 8. In this example, the control authority of PDG is an ellipse which has a semimajor axis of 75m and a semiminor axis of 30m and is tilted by 0.2 radians. The PDG execution error is a circle of radius 10m. Note that the order of the operations is important: if the erosion were performed with Figure 6(b) first, the entire map in Figure 8(b) would be feasible after the dilation. The dilation performed first could have an impact much larger than its mask size, especially when the dilation eliminates feasible "islands", leaving all cells with $+\infty$ cost for erosion.

6) Stage 0: Figure 9 shows the placement of the optimal landing ellipse. A Gaussian distribution is used for P_0, which has 4-σ semimajor axis and semiminor axis of 200m and 120m respectively, and is rotated by 0.2 radian.

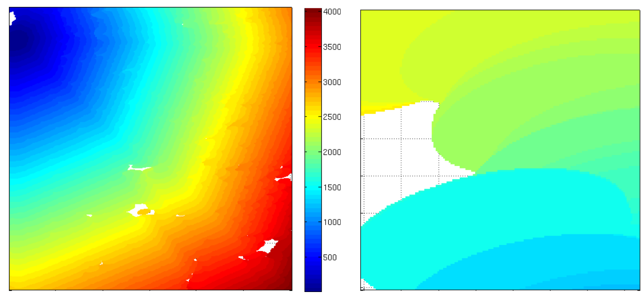
B. Trade-off Examples

The examples in this section illustrate some of the trade-offs that can be computed with the proposed approach. A slightly larger terrain of size 5,000m-by-5,000m gridded at 1m resolution is used. The HDA error is 1m, and the HDA



(a) 3km-by-3km map (b) Zoom-in of a 150×150 patch

Fig. 7. The cost map $J_2^D(x_2)$, after the dilation for 10m PDG error



(a) 3km-by-3km map (b) Zoom-in of a 150×150 patch

Fig. 8. $J_1^*(x_1)$ after the erosion for PDG corrections

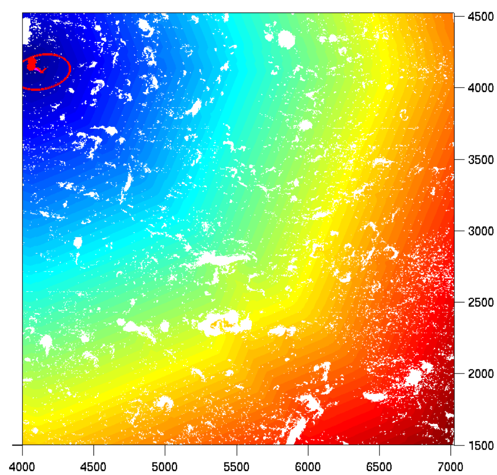


Fig. 9. The optimal landing ellipse drawn on the cost map $J_3^*(x_3)$

control authority is 2m. The PDG error is 100m, and the PDG control authority is 300m. The ignition ellipse is a circle of 1,000m radius, and the risk threshold is 1%. The rover's mission is to visit 7 out of 12 science targets.

1) HDA vs PDG: The first example shows a trade-off between HDA and PDG. By turning on and turning off each technology, we can generate a total of four cases. The left figures show the probability of mission success (i.e., no landing failure, and no rover entrapment) $\mathbb{P}(x_1 \notin \mathcal{F}_1)$, the right figures show the optimal rover route to visit multiple targets, and the optimal landing ellipse, plotted on top of the cost map $J_3^*(x_3)$ for rover driving.

Figure 10 shows a case where neither HDA nor PDG is used. This is equivalent to the landing strategy adopted by the current Mars missions such as Mars Science Laboratory

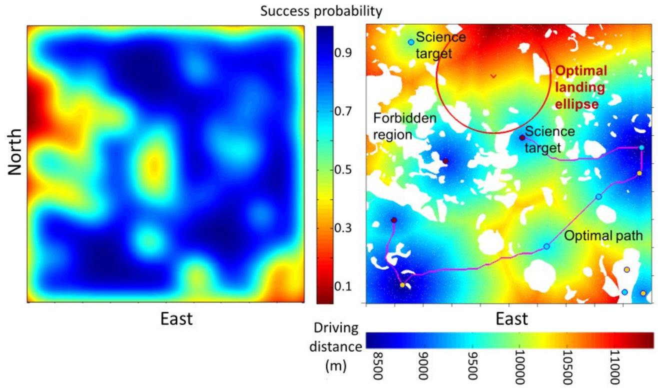


Fig. 10. Mission success map (left) and the optimal placement of a landing ellipse (right). No PDG, No HDA.

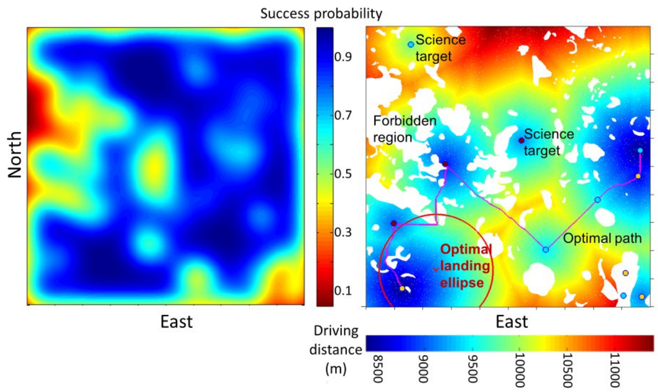


Fig. 11. Mission success map (left) and the optimal placement of a landing ellipse (right). No PDG, With HDA.

(MSL). The optimal landing ellipse is placed in the top middle portion of the map, and the expected driving distance is 10,443m.

Figure 11 shows a case where only the HDA is enabled. Although the mission success map does not look very different from the first case, the optimal landing ellipse is now placed at the lower left corner of the map. This is because the HDA erosion step takes out many small rocks scattered in the map. The expected driving distance has decreased to 9,145m.

Figure 12 shows a case where both PDG and HDA are enabled. The mission success map has changed dramatically, and the safe area to place the landing ellipse center has significantly enlarged. The expected drive distance has been further reduced to 8,657m.

Figure 13 shows a case where only the PDG is enabled. The safe region to place the landing ellipse has significantly reduced. In fact, none of the cells in the map has a probability of success greater than a risk threshold and no feasible landing ellipse was found. It may appear counterintuitive that enabling PDG makes it less safe compared to the first case shown in Figure 10. This result is due to the execution error associated with the PDG, as discussed below.

2) PDG error vs PDG control authority: This example shows the effect of the execution error and the control authority during the PDG phase, by varying these parameters and solving the $\mathcal{OPT}$ several times. The HDA was disabled

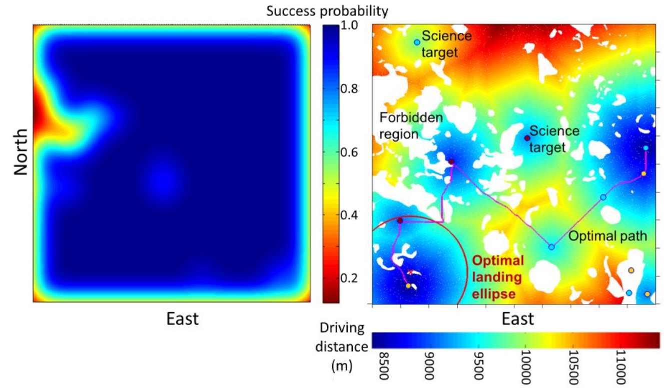


Fig. 12. Mission success map (left) and the optimal placement of a landing ellipse (right). With PDG, With HDA.

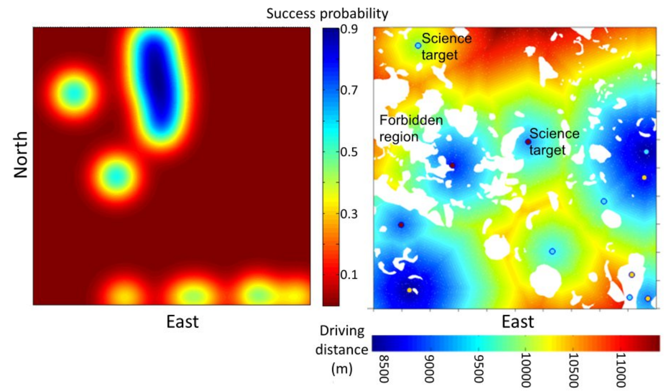


Fig. 13. Mission success map (left) and the optimal placement of a landing ellipse (right). With PDG, No HDA.

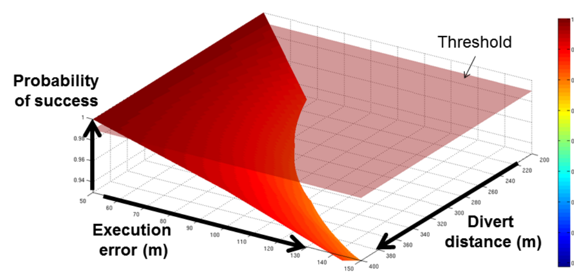


Fig. 14. Probability of mission success as a function of the PDG execution error and the divert distance.

in this example.

Figure 14 shows the probability of mission success $\mathbb{P}(x_1 \notin \mathcal{F}_1)$ given the solution of $\mathcal{OPT}$. One can notice that the relation between the execution error and the divert distance is highly nonlinear. As the execution error becomes larger, the dilation process in Stage 1 of the algorithm grows hazards by a larger amount. When there are small rocks scattered around in the terrain, each rock is *dilated* by the execution error, making wide portions of the region infeasible. When HDA is enabled, small rocks are completely *eroded* away first in Stage 2 of the algorithm, and the dilation due to the PDG execution error does not shrink the feasible region as much as the case with HDA.

To land on a rocky terrain, it is important to use HDA. In fact, although not plotted in the figure, when HDA is enabled,

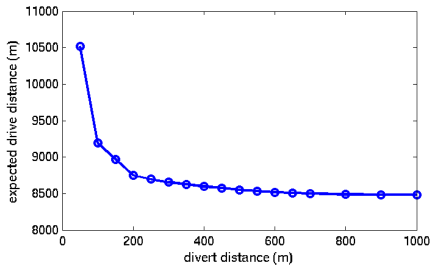
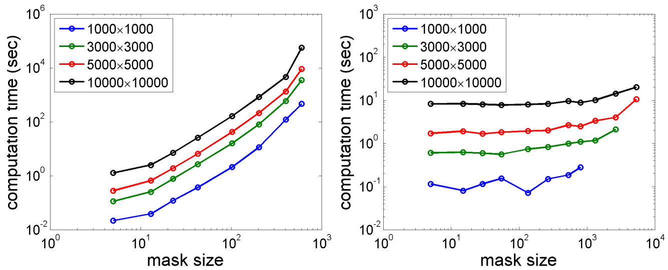


Fig. 15. Trade-off between divert distance and rover drive distance. With HDA and PDG.



(a) Dilation/Erosion (b) Cross-correlation

Fig. 16. Log-log plot of the computation time vs. the mask size. Different lines represent different map sizes.

any combination of execution error and divert distance can still ensure 100% mission success in this example.

3) PDG divert vs Rover driving: In this example, the divert distance of the PDG is varied in order to study the effect on the rover drive distance. HDA is enabled in this case. Figure 15 plots the expected rover driving distance as a function of the PDG divert distance. One can notice that a smaller divert distance can make the drive distance significantly longer. As the divert distance increases, the expected drive distance decreases, but there is a diminishing return once the divert distance becomes large enough.

C. Computation Times

Figure 16 shows the computation times of the key image processing functions, dilation/erosion and cross-correlation, as a function of the map size and the mask size. The plots are in log scales. The dilation and erosion take, respectively, the maximum and minimum value within a mask, and they can be reduced to the same operation with a flip of the sign. They were run on a machine with 192GB RAM and Intel(R) Xeon(R) X5690@3.47GHz that has 12 cores.

For the purpose of Mars missions, the largest mask for the dilation corresponds to PDG execution error w_1, which is about 100m. The PDG control authority u_1 can be as large as 1,000m, but if a faster computation is desired, the erosion operation can be performed at a reduced resolution. This subsampling would degrade the optimality of the solution, but the risk constraint is still ensured. The distribution of the ignition ellipse w_0 can be as large as several kilometers, but as Figure 16(b) shows, the cross-correlation is still orders of magnitude faster than the dilation/erosion. This is because the cross-correlation is a linear operation and can be performed efficiently using Fourier transforms.

V. CONCLUSION

This paper presented a multi-stage stochastic optimization framework for the combined EDL-Mobility analysis of planetary rover missions. By formulating the problem within a dynamic programming framework and by leveraging highly-optimized image-processing tools, we were able to perform trade-off analyses on real data with several hundred million states. Numerical examples illustrated how this tool would enable systems engineers and stake holders to make more informed decision based on quantitative analysis. Future work includes extending the decision making steps further backwards, such as to the entry insertion maneuver and to the trajectory correction maneuvers during orbital navigation. Another research direction is to extend this approach to missions with multi-landing opportunities, for example a potential Mars Sample Return mission or a lunar exploration mission with multiple vehicles.

ACKNOWLEDGMENTS

The research described in this paper was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] B. Kluge and E. Prassler, "Reflective Navigation: Individual Behaviors and Group Behaviors," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[2] M. P. Golombek, J. A. Grant, T. J. Parker, D. M. Kass, J. A. Crisp, S. W. Squyres, A. F. C. Haldemann, M. Adler, W. J. Lee, N. T. Bridges, R. E. Arvidson, M. H. Carr, R. L. Kirk, P. C. Knocke, R. B. Roncoli, C. M. Weitz, J. T. Schofield, R. W. Zurek, P. R. Christensen, R. L. Fergason, F. S. Anderson, and J. W. R. Jr, "Selection of the Mars Exploration Rover Landing Sites," *Journal of Geophysical Research*, vol. 108, no. 8072, pp. 13–1, 2003.

[3] D. Selva and E. Crawley, "Integrated Assessment of Packaging Architectures in Earth Observing Programs," in *Proceedings of the IEEE Aerospace Conference*, 2010.

[4] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, no. 7, pp. 1171–1176, 2002.

[5] F. Oldewurtel, C. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback," in *Proceedings of the IEEE Conference on Decision and Control*, 2008.

[6] Y. Kuwata and J. B. Balaram, "Combined EDL-mobility planning for planetary missions," in *Proceedings of the AIAA Infotech@Aerospace Conference*, 2011.

[7] B. Acikmese and S. Ploen, "Convex programming approach to powered descent guidance for mars landing," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[8] 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, R. L. Kirk, M. T. Mellon, S. W. Squyres, N. Thomas, and C. M. Weitz, "Mars Reconnaissance Orbiters High Resolution Imaging Science Experiment (HiRISE)," *Journal of Geophysical Research*, vol. 112, 2007.

[9] A. E. Johnson, A. R. Klumpp, J. B. Collier, and A. A. Wolf, "Lidar-Based Hazard Avoidance for Safe Landing on Mars," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, no. 6, pp. pp.1091–1099, 2002.

[10] Egon Balas, "The Prize Collecting Traveling Salesman Problem," *Networks*, vol. 19, no. 6, pp. 621–636, October 1989.

[11] A. Huertas, Y. Cheng, and R. Madison, "Passive Imaging Based Multi-cue Hazard Detection Spacecraft Safe Landing," in *Proceedings of the IEEE Aerospace Conference*, 2006.