Probabilistic Kinematic State Estimation for Motion Planning of Planetary Rovers

Sourish Ghosh[1], Kyohei Otsu[2], and Masahiro Ono[2]

***Abstract*— Kinematics-based collision detection is important for robot motion planning in unstructured terrain. Especially, planetary rovers require such capability as a single collision may lead to the termination of a mission. For onboard computation, typical numeric approaches are unsuitable as they are computationally expensive and unstable on rocky terrain; instead, a light-weight analytic solution (ACE: Approximate Clearance Evaluation) is planning to be used for the Mars 2020 rover mission. ACE computes the state bounds of articulated suspension systems from terrain height bounds, and assess the safety by checking the constraint violation of states with the worst-case values. ACE's conservative safety check approach can sometimes lead to over-pessimism: feasible states are often reported as infeasible, thus resulting in frequent false positive detection. In this paper, we introduce a computationally efficient probabilistic variant of ACE (called p-ACE) which estimates the probability distributions of states in real time. The advantage of having probability distributions over states, instead of deterministic bounds, is to provide more flexible and less pessimistic worst-case evaluation with probabilistic safety guarantees. Empirically derived distribution models are used to compute the total probability of constraint satisfaction, which is then used for path assessment. Through experiments with a high-fidelity simulator, we empirically show that p-ACE relaxes the deterministic state bounds without losing safety guarantees.**

I. Introduction

To competently perform motion planning on uneven terrain, a mobile robot must predict the effect of its interaction with the environment. The accurate prediction of this interaction helps planners to make more efficient plans, reducing the risk of detours and immobilization. It is particularly important for motion planners of planetary rovers, which have to drive long distances without being trapped by terrain hazards. Due to the severe restriction in radiation-tolerant computational resources, the current navigation software only relies on the geometrical statistics of terrain within rover-sized discs [1], [2]. Such simplified interaction model tends to fail in a cluttered environment, limiting the use of autonomous navigation mode only in benign terrain.

In response to the increasing demand for longer autonomous traversal in complex environments, NASA/JPL Mars 2020 rover (Fig. 1) is planning to use a new motion-planning algorithm. Approximate Clearance Evaluation (ACE) [3], [4] is developed as its core algorithm for estimating the rover-terrain interaction. It is a light-weight inverse-kinematics solver that computes the bounds of a kinematic state based on the uncertainty in wheel-terrain contact. Since ACE conservatively evaluates the worst-case state in an analytic form, it is guaranteed to be safe, as well as computationally efficient. However, the conservatism sometimes makes ACE over-pessimistic: traversable paths are reported as collisions, especially in terrains with higher rock abundance. It is mainly because large uncertainty in wheel placement is directly propagated to the vehicle's other states. Such pessimism may result in the excessive reduction of candidate paths during motion planning.

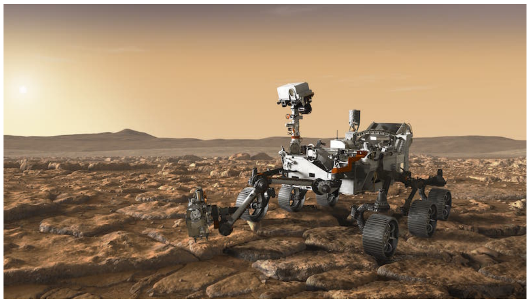


Fig. 1: An artist's concept of the Mars 2020 rover exploring a rocky environment. (Credit: NASA/JPL-Caltech)

There have been other efforts in the area of robot navigation in natural terrain. In the context of kinematics estimation, the most common approach is to use a numeric method to iteratively solve the non-linear optimization problem under various kinematic constraints. Many physics-based simulators have this capability, such as ODE (Open Dynamics Engine) [5] and ROAMS (Rover Analysis, Modeling, and Simulation) [6], [7]. Based on the numerical kinematics evaluation, motion planning has been performed for wheeled robots [8], [9] and tracked robots [10]. In general, the numerical approach is computationally expensive for motion planning, which requires thousands of pose evaluations in each planning cycle, and almost intractable for planetary rovers. Moreover, the run-time performance changes significantly according to the complexity of the vehicle model and the terrain shape.

For the realization of a robust yet efficient collision prediction, this paper presents a probabilistic variant of ACE (named p-ACE). p-ACE estimates the probability distributions over vehicle states based on the vehicle kinematics model and terrain height uncertainty. It has a

*This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. U.S. Government sponsorship acknowledged.

[1]Sourish Ghosh is with the Department of Mathematics, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. Email: sourishg@iitkgp.ac.in

[2]Kyohei Otsu and Masahiro Ono are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA. Email: {Kyohei.Otsu, Masahiro.Ono}@jpl.nasa.gov

prominent run-time performance by fully exploiting the analytic formulation in ACE. Based on the state bounds computed by ACE, we empirically train the distribution models with Monte-Carlo methods. By having probability distributions over states instead of deterministic bounds, we can set more flexible and less pessimistic worst-case evaluation with probabilistic safety guarantees. The total probability of constraint satisfaction is used in risk-aware motion planning. The simulation results with the ROAMS simulator shows that p-ACE successfully relaxes ACE's deterministic bounds without losing safety guarantees.

The rest of paper is structured as follows: In Section II, we introduce the formulation of ACE and discuss its conservatism. In Section III, a probabilistic extension of ACE is presented. Section IV provides the simulation results, and Section V concludes the paper.

II. Approximate Clearance Estimator (ACE)

A configuration of the rover with the rocker-bogie differential articulation is represented by a 10-element vector

$$[x, y, z, \phi, \theta, \psi, \delta_l, \delta_r, \beta_l, \beta_r] \in \mathbb{R}^{10} \tag{1}$$

where (x, y, z) is the position of the rover center in the world frame; (ϕ, θ, ψ) is the orientation representing roll, pitch and yaw; (δ, β) is the rocker and bogie angles in the left and right suspensions. The objective of ACE is to determine the bounds of these configurations in response to a pose query (x, y, ψ) in 2D plane under uncertainty in wheel-terrain contact. This section describes the analytic solutions presented in [3].

A. Reference Frames

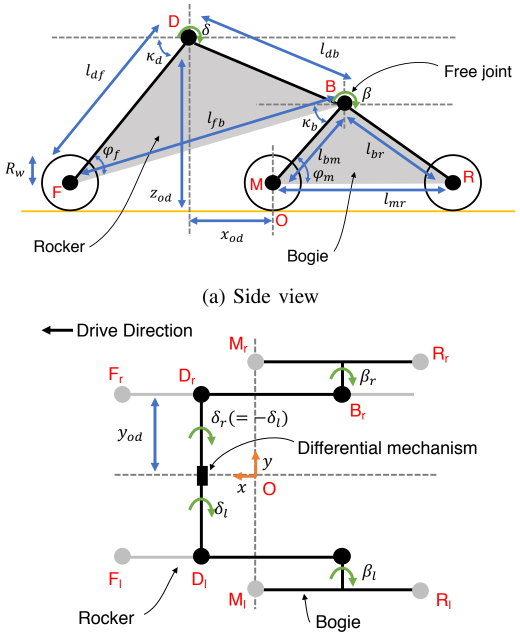
Following the aerospace convention, the body frame $\mathscr{F}_r$ of the rover employs the forward-right-down coordinate system. The origin is set to the center of middle wheels at the height of ground contact point when the rover is stationary on the flat ground. $\mathscr{F}_r^{level}$ is defined at the same origin with $\mathscr{F}_r$, but z-axis aligned with the gravity direction by compensating the pitch angle. A global reference frame is defined as a north-east-down coordinate system, denoted by $\mathscr{F}_w$. The terrain geometry such as a Digital Elevation Map (DEM) is expressed in this frame. The wheel heights are also described in this frame, which act as an input for the inverse-kinematics problem. Note that since z is pointing down, greater wheel height indicates that the wheel is shifted downward.

B. Rocker-Bogie Articulation Model

The schematic of the rocker-bogie suspension is shown in Fig. 2. It is a two-link suspension system comprised of a trailing rocker-arm with one wheel and a bogie with two wheels pivoted at one end of the rocker arm. According to the previous Mars rover conventions, we define the rocker side as the direction of motion.

We start with the bogie link state. The bogie joint height can be estimated from middle and rear wheel heights (z_m, z_r)

$$z_b = z_m - l_{bm} \sin \kappa_b(z_m, z_r) \tag{2}$$



(a) Side view

(b) Top view

Fig. 2: Schematic of a rocker-bogie differential suspension.

where $\kappa_b(\cdot)$ denotes the triangular geometry for the bogie triangle defined as

$$\kappa(z_m, z_r) = \varphi_m + \sin^{-1}\left(\frac{z_m - z_r}{l_{mr}}\right) \tag{3}$$

and $l_{(\cdot)}$ and $\varphi_{(\cdot)}$ represent the lengths and angles defined in Fig. 2. Using the height of bogie joint z_b, the rocker joint height can be computed as

$$z_d = z_f - l_{df} \sin \kappa_d(z_f, z_b) \tag{4}$$

where κ_d represents the similar triangular geometry with (3) but for the rocker link. Given the heights of the wheels and joints, rocker and bogie angle changes are computed as

$$\delta_l = -\delta_r = \frac{\kappa_d(z_{f_r}, z_{b_r}) - \kappa_d(z_{f_l}, z_{b_l})}{2} \tag{5}$$

$$\beta_l = \kappa_d(z_{f_l}, z_{b_l}) - \kappa_b(z_{m_l}, z_{r_l}) - \kappa_{d0} + \kappa_{b0} \tag{6}$$

$$\beta_r = \kappa_d(z_{f_r}, z_{b_r}) - \kappa_b(z_{m_r}, z_{r_r}) - \kappa_{d0} + \kappa_{b0} \tag{7}$$

where κ_{d0} and κ_{b0} denote the angles of the rocker and bogie joints when the rover is on a flat plane. The body roll and pitch angles are obtained as

$$\phi = \sin^{-1}\left(\frac{z_{d_r} - z_{d_l}}{2y_{od}}\right) \tag{8}$$

$$\theta = \kappa_{d0} - \frac{\kappa_d(z_{f_l}, z_{r_l}) + \kappa_d(z_{f_r}, z_{r_r})}{2} \tag{9}$$

and the height of $\mathscr{F}_r$ with respect to $\mathscr{F}_w$ by

$$z_o = \frac{z_{d_l} + z_{d_r}}{2} + x_{od} \sin\theta \cos\phi - z_{od} \cos\theta \cos\phi \tag{10}$$

where (x_{od}, y_{od}, z_{od}) denotes the nominal position of the right rocker joint.

C. State Uncertainty Propagation

The ACE algorithm is designed to quickly estimate the bounds of the vehicle's attitude and suspension states. For an input terrain geometry m and target rover pose $\mathbf{x} = (x, y, \psi)$, ACE determines the bounds of the wheel heights using terrain data and propagates the bounds to the vehicle state with kinematic equations described above. An important observation here is that the extremes of state bounds correspond to one of $2^6 = 64$ possible combinations of wheel height extremes. Based on this observation, we can compute the worst-case states in a closed form. Let us represent an interval as follows

$$[x] \equiv [x^-, x^+] = \{x \in \mathbb{R}^* \mid x^- \leq x \leq x^+\} \tag{11}$$

where symbol $\mathbb{R}^*$ denotes an extended real defined as $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, \infty\}$. Using this notation, we express the set of wheel height intervals

$$\{[z_{f_l}], [z_{m_l}], [z_{r_l}], [z_{f_r}], [z_{m_r}], [z_{r_r}]\} \tag{12}$$

for all six wheels. From these wheel height intervals, we can derive closed-form solutions for other state intervals $([z_o], [\phi], [\theta], [\delta_l], [\delta_r], [\beta_l], [\beta_r])$ using the kinematic relations. State bounds can be formulated using the wheel height intervals as follows:

$$[z_b] = [z_m^- - l_{bm} \sin \kappa_b(z_m^-, z_r^-),\ z_m^+ - l_{bm} \sin \kappa_b(z_m^+, z_r^+)] \tag{13}$$

$$[z_d] = [z_f^- - l_{df} \sin \kappa_d(z_f^-, z_b^-),\ z_f^+ - l_{df} \sin \kappa_d(z_f^+, z_b^+)] \tag{14}$$

$$[\delta_l] = -[\delta_r] \tag{15}$$

$$= \left[\frac{\kappa(z_{f_r}^-, z_{b_r}^+) - \kappa_d(z_{f_l}^+, z_{b_l}^-)}{2}, \frac{\kappa(z_{f_r}^+, z_{b_r}^-) - \kappa_d(z_{f_l}^-, z_{b_l}^+)}{2} \right] \tag{16}$$

$$\begin{aligned} [\beta_l] \subseteq \Big[&\kappa_d(z_{f_l}^-, z_{b_l}^+) - \kappa_b(z_{m_l}^+, z_{r_l}^-) - \kappa_{d0} + \kappa_{b0}, \\ &\kappa_d(z_{f_l}^+, z_{b_l}^-) - \kappa_b(z_{m_l}^-, z_{r_l}^+) - \kappa_{d0} + \kappa_{b0} \Big] \end{aligned} \tag{17}$$

$$\begin{aligned} [\beta_r] \subseteq \Big[&\kappa_d(z_{f_r}^-, z_{b_r}^+) - \kappa_b(z_{m_r}^+, z_{r_r}^-) - \kappa_{d0} + \kappa_{b0}, \\ &\kappa_d(z_{f_r}^+, z_{b_r}^-) - \kappa_b(z_{m_r}^-, z_{r_r}^+) - \kappa_{d0} + \kappa_{b0} \Big] \end{aligned} \tag{18}$$

$$[\phi] = \left[\sin^{-1}\left(\frac{z_{d_r}^- - z_{d_l}^+}{2y_{od}} \right),\ \sin^{-1}\left(\frac{z_{d_r}^+ - z_{d_l}^-}{2y_{od}} \right) \right] \tag{19}$$

$$\begin{aligned} [\theta] = \Bigg[&\kappa_{d0} - \frac{\kappa_d(z_{f_l}^+, z_{b_l}^-) + \kappa_d(z_{f_r}^+, z_{b_r}^-)}{2}, \\ &\kappa_{d0} - \frac{\kappa_d(z_{f_l}^-, z_{b_l}^+) + \kappa_d(z_{f_r}^-, z_{b_r}^+)}{2} \Bigg] \end{aligned} \tag{20}$$

$$\begin{aligned} [z_o] \subseteq \Bigg[&\frac{z_{d_l}^- + z_{d_r}^-}{2} - z_{od} \cos|\theta|^+ \cos|\phi|^+ \\ &+ x_{od} \min(\sin\theta^- \cos|\phi|^-,\ \sin\theta^- \cos|\phi|^+), \\ &\frac{z_{d_l}^+ + z_{d_r}^+}{2} - z_{od} \cos|\theta|^- \cos|\phi|^- \\ &+ x_{od} \max(\sin\theta^+ \cos|\phi|^-,\ \sin\theta^+ \cos|\phi|^+) \Bigg] \end{aligned} \tag{21}$$

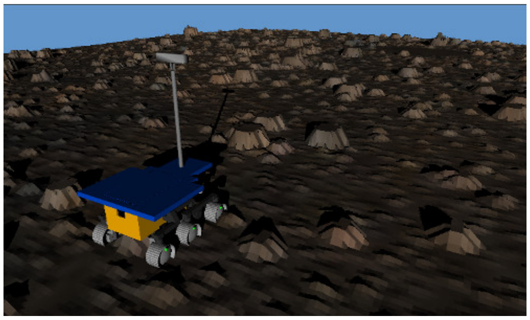


Fig. 3: A screenshot of the ROAMS simulator showing a model of the Rocky 8 rover.

D. Belly Pan Clearance

Using the vehicle attitude, the unit normal vector of the rover belly pan in $\mathscr{F}_r^{level}$ is given as:

$$\hat{\mathbf{n}} = \frac{1}{\sqrt{1 + \tan^2\theta + \tan^2\phi}} \begin{bmatrix} \tan\theta \\ -\tan\phi \\ 1 \end{bmatrix} \tag{22}$$

assuming roll and pitch to be the conservative fixed-axis rotation. The planar function representing the belly pan plane is given as:

$$f(x, y) = -x\tan\theta + y\tan\phi - c_0\sqrt{1 + \tan^2\theta + \tan^2\phi} \tag{23}$$

where c_0 denotes the nominal clearance when the rover is place on a flat plane. The plane function $f(x, y)$ is actually the height of the belly pan point (x, y) in $\mathscr{F}_r^{level}$. The minimum clearance can be computed by finding the point that has the smallest vertical gap:

$$c = \min_{(x,y) \in \mathscr{B}} (H(x, y) - f(x, y) - z_o^+) \tag{24}$$

where $H(x, y)$ is the height of the terrain at point (x, y) and $\mathscr{B}$ is the set of (x, y) pairs that define the belly pan.

E. Conservatism of ACE

The input to ACE is a set of wheel height bounds from terrain measurements. In the real world scenario, it is difficult to obtain the accurate measurement of terrain height, because of sensor noise or occlusions. Therefore, we typically set margins to ensure that true wheel heights are actually bounded by measurement intervals. The uncertainty in wheel heights propagates to the other states of the vehicle via kinematics equations. Large uncertainties in the estimated state bounds make ACE over-conservative.

To show the over-conservatism in ACE estimation, we performed simulations with the ROAMS simulator shown in Fig. 3. A DEM of the Jezero crater on Mars (a candidate site for the Mars 2020 mission) is used as a base terrain. We populate rocks according to the empirical model [11], which is parameterized by the Cumulative Fractional Area (CFA) covered by rocks. In the simulation environment, we drove the Rocky 8 rover model covering the entire terrain and recorded all vehicle states with ground clearance for every pose at every simulation step. ROAMS uses the numeric Newton-Raphson method to solve the kinematic state of the

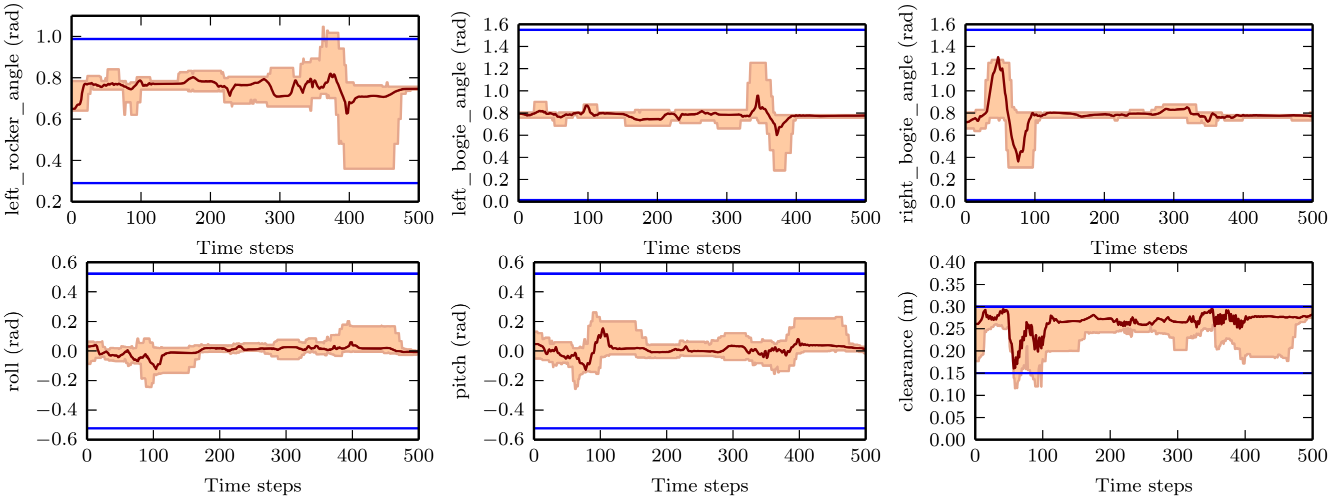


Fig. 4: ACE estimation results. A model of the Rocky 8 Rover was driven on a synthetic terrain of the Jezero crater with 5% CFA using the ROAMS simulator, and the states were recorded at every simulation step. The red line corresponds to the ground-truth state value and the shaded region represents the intervals between the ACE bounds. The blue lines indicate the allowable state range.

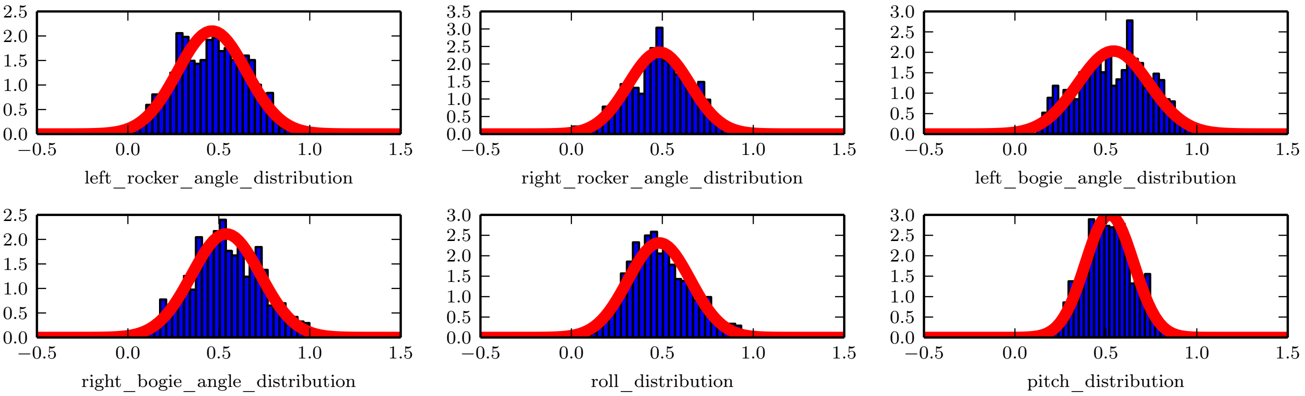


Fig. 5: Normalized histograms of the all the state variables (left and right rocker angles, left and right bogie angles, roll angle, pitch angle) computed using (27). The red curve represents the corresponding normal distribution.

rover, as well as performing multi-body dynamics simulation to accurately model rover behavior on rough terrain. Hence, we treat the output from ROAMS as ground-truth.

We ran the ACE algorithm to predict state bounds while driving. Fig. 4 shows the state bounds on a 5% CFA terrain. The shaded region depicts the uncertainty of the state values. It can be seen from the plot that the uncertainty span (i.e., the difference between lower and upper bounds) is quite high at certain time steps. State validation is done conservatively by checking if the complete state range is within its constraint bounds (blue lines in Fig. 4). This worst-case range validation technique leads to false positives during collision checking. In other words, it only takes one state with a large uncertainty to mistakenly report a collision despite all of the other states being well within the allowable range. Although this conservative behavior is preferable for safe operations, it can capture a configuration that is physically impossible, thereby excessively reducing path options. We thus introduce the probabilistic variant of ACE (p-ACE) in the next section, which essentially aims to reduce the pessimism of ACE while giving a probabilistic safety guarantee.

III. Probabilistic ACE (p-ACE)

p-ACE aims to relax the hard constraints imposed by ACE by removing probabilistically unlikely configurations. In deterministic ACE, if either the lower or the upper bound of a state violates the constraint bounds then a collision would be reported regardless of whether its true state is safe. To mitigate this pessimistic worst-case range validation of a state, we introduce a probabilistic notion of constraint satisfaction, where we estimate a probability distribution over a state, instead of deterministic upper and lower bounds. Computing a distribution of a kinematic state is an expensive operation, typically requiring numeric methods for possible configurations. In order to avoid computationally expensive operations, we introduce the notion of a *normalized* state variable. By empirically verifying that the normalized state follows a normal distribution, we develop an efficient method to perform probabilistic state estimation.

A. *Problem Formulation*

Given a target rover pose $\mathbf{x} = (x, y, \psi)$ and a terrain model m, we want to estimate the probability of constraint satisfaction of the remaining states $\omega \in \mathbb{R}^7$. This probability can be expressed as

$$\mathbb{P}(\omega \in \Omega|\mathbf{x},m) = \int_{\omega\in\Omega} p(\omega|\mathbf{x},m)d\omega \tag{25}$$

where $p(\omega|\mathbf{x},m)$ is the joint probability distribution of all the states and Ω is the allowable range of state values from safety constraints. The objective of p-ACE is to estimate the constraint satisfaction probability (25) in response to queries from motion planners. For readability, we will omit the symbol m for the rest of paper, assuming that the terrain model does not change in a single motion planning cycle.

B. Normalized State

To efficiently estimate the probability, we introduce a concept of normalized state. Given an ACE-bounded state $[\omega_i] \equiv [\omega_i^-, \omega_i^+]$, we define the normalized state as follows:

$$\hat{\omega}_i = \frac{\omega_i - \omega_i^-}{\omega_i^+ - \omega_i^-}\,. \tag{26}$$

The normalized state $\hat{\omega}$ is a ratio that represents the location of a state variable ω inside the kinematics-based upper/lower bounds.

C. Identifying Probability Distributions

To characterize the probability distribution over a normalized state, we employ a numeric approach based on offline Monte-Carlo simulations. Here, we use the ROAMS simulator as in the previous section, but it is straightforward to extend the approach to any simulations/hardware setups. We ran simulations on different synthetic Mars terrains with varying rock distributions (CFA: 1%–10%). Fig. 5 shows the histograms of all the normalized state variables combining different CFA settings. From these histograms, we can empirically derive that the normalized state follows a normal distribution (shown in red)

$$\hat{\omega}_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{27}$$

where the distribution is parameterized by a state-specific mean and variance (μ_i, σ_i^2).

Assuming the normal distributions over normalized states, we can derive the probability distribution for i-th state from definition (27)

$$p(\omega_i|\mathbf{x}) = \mathcal{N}\left(\omega_i^- + \mu_i(\omega_i^+ - \omega_i^-),\ \sigma_i^2(\omega_i^+ - \omega_i^-)^2\right) \tag{28}$$

using the computed ACE bound $[\omega_i]$ for input pose $\mathbf{x}$.

Note that, although in many cases a random variable will follow the normal distribution, our method is not limited to a specific type of distribution. We also stress that the distribution identification process is performed offline, hence it does not affect the performance of online state estimation.

D. Estimating Constraint Satisfaction Probability

Since we separately identified the underlying probability distributions for each state, it is reasonable to assume variable independence between different states. Under the independence assumption, (25) is decomposed to

$$\begin{aligned}\mathbb{P}(\omega \in \Omega|\mathbf{x}) &= \prod_i \mathbb{P}(\omega_i \in \Omega_i|\mathbf{x}) && (29)\\ &= \prod_i \int_{\omega_i\in\Omega_i} p(\omega_i|\mathbf{x})d\omega_i\,. && (30)\end{aligned}$$

In (28), we derived that each state follows a normal distribution. Therefore, it is trivial to compute the probability of satisfaction from the above equation.

E. Extension to Clearance Satisfaction Probability

Once we have estimated the distributions of the suspension and attitude states, we can extend the probabilistic notion to clearance. From the definition in (24), the clearance depends on the attitude of the rover. We propagate the distributions from attitude to clearance using the linearization of (23):

$$f_l(x,y) = -x\theta + y\phi - c_0\,. \tag{31}$$

The corresponding clearance $c_l(x,y)$ is obtained as

$$c_l(x,y) = H(x,y) + x\theta - y\phi + c_0 - z_o^+\,. \tag{32}$$

Here, we observe a major difference in the definition of clearance compared to the definition in (24). In p-ACE, every point (x,y) of the belly pan has a probability distribution of its clearance. In the deterministic definition, we defined clearance as the minimum vertical gap between the terrain and the belly pan. In p-ACE, the probability of constraint satisfaction of clearance (denoted as ω_c) is found by taking the minimum constraint satisfaction probability of all $c_l(x,y)$

$$\mathbb{P}(\omega_c \in \Omega_c|\mathbf{x}) \equiv \min_{(x',y')\in\mathscr{B}} \mathbb{P}(c_l(x',y') \in \Omega_c|\mathbf{x}) \tag{33}$$

where Ω_c denotes the allowable state range for clearance and $\mathscr{B}$ is defined similarly as in (24).

F. Collision checking along a path

Let $X = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k, \cdots\}$ denote a possible candidate path generated from a motion planning algorithm. The collision risk of a path can be defined as a simple probabilistic property

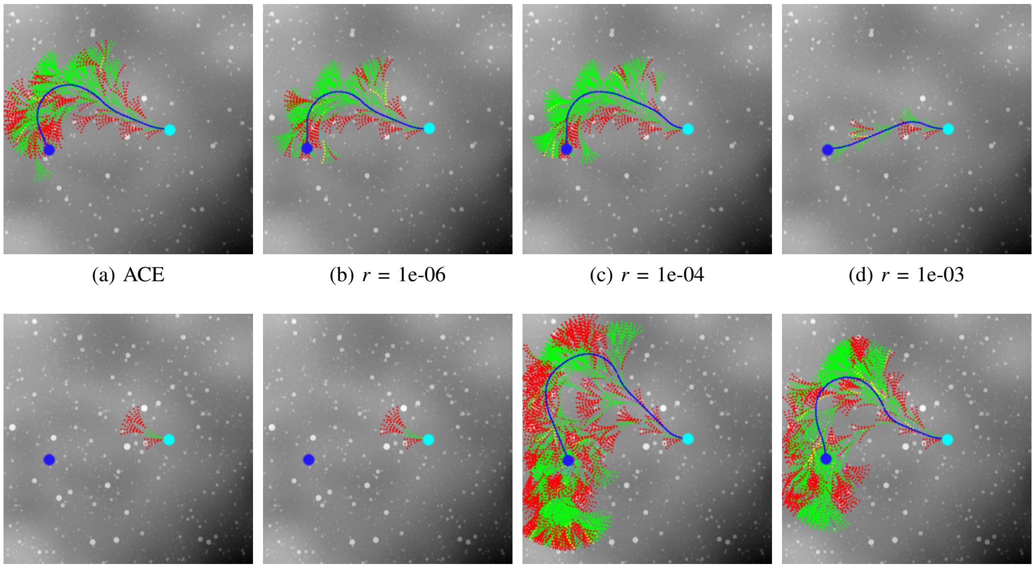
$$R(X) \equiv \max_k \mathbb{P}(\omega \notin \Omega|\mathbf{x}_k) \tag{34}$$

Note that ω is an extended state including ω_c. We define a threshold r called *risk-factor* on $R(X)$. The path X is collision free if $R(X) < r$.

IV. Experimental Results

A. Probabilistic Collision Estimation

We performed experiments to compare the performances of p-ACE with ACE. We synthesized multiple Martian terrains with different rock distributions. For a set of fixed-orientation poses $\{\mathbf{x} = (x,y,\psi)|\psi = 0\}$, we ran both methods, and generated binary/probabilistic collision maps based on its output. In the probabilistic maps, each pose is evaluated with a collision risk defined in (34) instead of a binary feasibility value from ACE. Examples of these maps


(a) ACE (b) r = 1e-06 (c) r = 1e-04 (d) r = 1e-03

(e) ACE (f) r = 1e-06 (g) r = 1e-04 (h) r = 1e-03

Fig. 6: Risk-aware A* motion planning using ACE and p-ACE. The terrain is a simulation of the Jezero crater with a total map size of 15m × 15m. CFA level for (a)-(d) is 5% and for (e)-(h) is 7%. p-ACE is executed at three different risk-factors (r) defined in (34). The *sky-blue* circle denotes the start state and the *deep-blue* circle denotes the end state. The *blue* curve is the optimal path found. The *green* arcs denote traversable edges whereas the *red* arcs denote untraversable edges. We observe that as we increase the risk-factor, the number of untraversable edges decrease. In (e) due to hard ACE-bounds no path is found whereas in (g) and (h) optimal yet safe paths are found even when the risk-factor is low (1e−4 and 1e−3, respectively). This shows that p-ACE reduces the pessimism of ACE and often leading to better and more optimal paths, as seen in (d).

are presented in Fig. 7. Using the probabilistic map, we can evaluate how likely the configuration will violate constraints, whereas the ACE collision map only reports if the pose is safe or not. Needless to say, the ability to quantify the expectations will help various motion planners that take risk into account.

We computed statistics from this setup. Fig. 8 shows the cumulative distribution functions of constraint violation probabilities for all poses detected as infeasible by deterministic ACE (*i.e.*, all configurations with $\mathbb{P}(\omega \notin \Omega|\mathbf{x}) > 0$). The constraint violation probability is calculated for each individual state as well as the joint state. These results indicate the following: 1) Constraint violation of *clearance* contributes most to the overall collision risk than other states like suspension and attitude angles; 2) Almost 10% of the unsafe poses detected by ACE has actually very low collision risk (less than 1%) which can be easily relaxed during risk-aware motion planning discussed in the next section.

B. Risk-aware Motion Planning

We performed experiments on risk-aware motion planning with ACE or p-ACE as a collision checker. We used the A* path planning algorithm with arc-based motion primitives. Following Mars rover conventions, each motion is parameterized as an arc with a constant curvature in a discrete set. The heuristic function for the A* search is defined as the Euclidean distance between the current pose and the goal. We restricted our motion planning region to a 15m × 15m map. As terrain models, we used synthetic Martian terrain data of the Jezero crater. Various rock abundance levels were used ranging from 1% to 15% CFA.

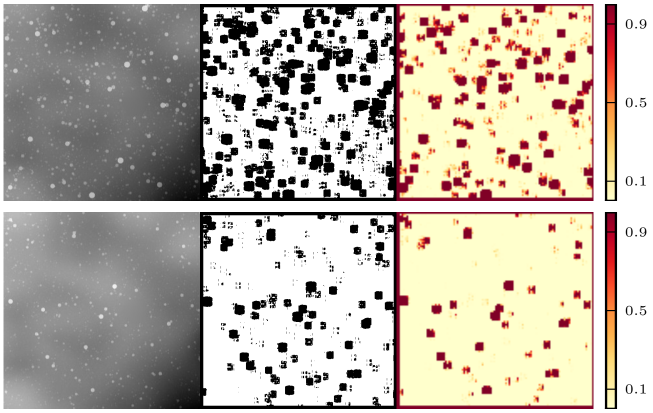


Fig. 7: Extension of a binary collision map (ACE) to a probabilistic collision map (p-ACE). **Left**: DEM. **Middle**: ACE collision map in C-space (*black* = inaccessible, *white* = accessible). **Right**: p-ACE collision map drew as a heatmap in C-space. **Top**: CFA 10%. **Bottom**: CFA 5%. For all rover poses, the yaw angle is fixed to $\psi = 0$.

We compared ACE and p-ACE at various risk-factors: 1e−6, 1e−4 and 1e−3. For deterministic ACE, collision checking is done on a pose $\mathbf{x}$ by checking if the ranges of all the states ω are well within bounds of their corresponding constraints Ω (*i.e.*, $\omega \in \Omega$). For p-ACE, based on the risk-factor r, (34) is used to check if a candidate path is probabilistically safe. For each risk-factor and each CFA level we ran 100 simulations and measured the probability of success in path planning. *Success* here means the ability to compute a safe path to the goal. Fig. 6 shows representative simulation results on 5% and 7% CFA. Fig. 9 shows the probability of success using ACE and p-ACE. It is interesting

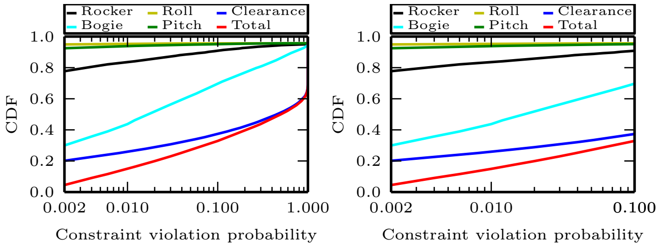


Fig. 8: A cumulative distribution function (CDF) of the constraint violation probability of states on potentially inaccessible poses determined by ACE ($\forall \mathbf{x}\ \mathbb{P}(\omega \notin \Omega|\mathbf{x}) > 0$). The x-axis is in log-scale. The figure on the right shows the trend in low probability regions: $[0.002, 0.1]$.

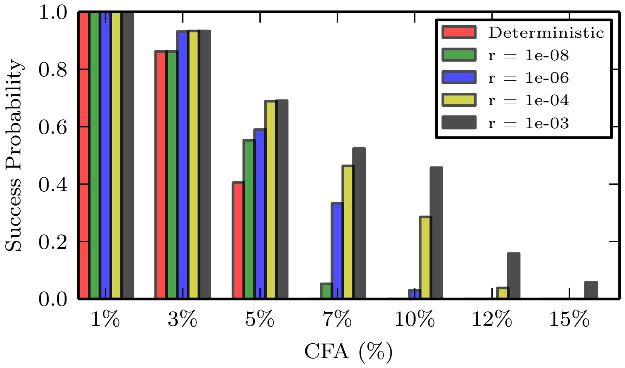


Fig. 9: A comparison of the probability of success of risk-aware path planning using ACE and p-ACE at various risk-factors (r) and CFAs.

to observe that for CFA $\geq$ 7% deterministic ACE has 0% success rate, whereas p-ACE with a risk-factor of 1e$-$3 had around 50% success rate. Even with the small relaxation of hard bounding in ACE, the success rate of path planning is drastically increased. This is a clear indication of the reduced pessimism involved with p-ACE. Note that all the tests are run with the Rocky 8 rover model which is a much smaller rover compared to the Curiosity rover. Hence it has low success probabilities in higher CFA terrains. The Curiosity rover however, can operate at $>$ 15% CFA but our main objective here is to show the improved success rate of planning using p-ACE.

V. Conclusions

In this paper, we introduced a probabilistic variant of ACE for kinematic state estimation of articulated suspension rovers. We discussed the over-pessimism involved in the worst-case range validation of states in ACE. We formulated p-ACE to compute probability distributions of rover states and use them as a measure for safety assessment. From the empirical evaluation, we showed that this method relaxes ACE's hard bounds at higher rock abundance levels, and provides a path's safety probability which can be directly fed into risk-aware motion planners.

References

[1] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *IEEE Aerospace Conference*, vol. 5, pp. 2025–2036, 2002.

[2] J. Biesiadecki and M. Maimone, "The Mars Exploration Rover surface mobility flight software: driving ambition," in *IEEE Aerospace Conference*, 2006.

[3] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for kinematically constrained articulated suspension systems," *arXiv e-prints*, 2018.

[4] K. Otsu, *Study on Robotic Intelligence for Vision-based Planetary Surface Navigation*. PhD thesis, The University of Tokyo, 2016.

[5] *Open Dynamics Engine*. http://www.ode.org.

[6] A. Jain, J. Guineau, C. Lim, W. Lincoln, M. Pomerantz, G. Sohl, and R. Steele, "ROAMS: Planetary surface rover simulation environment," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.

[7] A. Jain, J. Balaram, J. Cameron, J. Guineau, C. Lim, M. Pomerantz, and G. Sohl, "Recent developments in the ROAMS planetary rover simulation environment," in *IEEE Aerospace Conference*, vol. 2, pp. 861–876, 2004.

[8] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, pp. 141–166, 2007.

[9] D. Helmick, A. Angelova, and L. Matthies, "Terrain adaptive navigation for planetary rovers," *Journal of Field Robotics*, vol. 26, no. 4, pp. 391–410, 2009.

[10] P. Papadakis and F. Pirri, "3D mobility learning and regression of articulated, tracked robotic vehicles by physics-based optimization," in *Workshop on Virtual Reality Interaction and Physical Simulation, Eurographics*, pp. 147–156, 2012.

[11] M. Golombek, A. Huertas, J. Marlow, B. McGrane, C. Klein, M. Martinez, R. Arvidson, T. Heet, L. Barry, K. Seelos, *et al.*, "Size-frequency distributions of rocks on the northern plains of Mars with special reference to Phoenix landing surfaces," *Journal of Geophysical Research: Planets*, vol. 113, no. E3, 2008.