

# Autonomous Navigation over Europa Analogue Terrain for an Actively Articulated Wheel-on-Limb Rover

William Reid<sup>1</sup>, Michael Paton<sup>1</sup>, Sisir Karumanchi<sup>1</sup>, Brendan Chamberlain-Simon<sup>1</sup>,  
Blair Emanuel<sup>1</sup>, Gareth Meirion-Griffith<sup>1</sup>

**Abstract**—The ocean world Europa is a prime target for exploration given its potential habitability [1]. We propose a mobile robotic system that is capable of autonomously traversing tens of meters to visit multiple sites of interest on a European analogue surface. Due to the topology of European terrain being largely unknown, it is desired that this mobility system traverse a large variety of terrain types. The mobility system should also be capable of crossing unstructured terrain in an autonomous manner given the communications limitations between Earth and Europa.

A wheel-on-limb robotic rover is presented that may actively conform to terrain features up to 1.5 wheel diameters tall while driving. The robot uses a sampling-based motion planner to generate paths that leverage its unique locomotive capabilities. The planner assesses terrain hazards and wheel workspace limits as obstacles. It may also select a mobility mode based on predicted energy usage and the need for limb articulation on the terrain being traversed. This autonomous mobility was evaluated on chaotic salt-evaporite terrain found in Death Valley, CA, an analogue to the European surface. Over the course of 38 trials, the rover autonomously traversed 435 m of extreme terrain while maintaining a rate of 0.64 traverse ending failures for every 10 m driven.

## I. INTRODUCTION

Within the next decade the National Aeronautics and Space Administration (NASA) will launch the Europa Clipper mission to orbit Jupiter. The mission's primary goal is to observe the ocean world Europa and determine its potential habitability [2]. To complement Europa Clipper, a static lander mission has been proposed as part of the Europa Lander Study [3]. The lander would be a static in-situ sampling mission that would characterize the biological potential of Europa's ocean by investigating the Europa surface. A natural extension to both Europa Clipper and Europa Lander would be to send a mobile agent to the European surface that would visit multiple sites. To address such a mission concept, we investigate robotic mobility on a Europa analogue terrain. Using the Europa Lander mission for reference, a Europa surface platform would require a significant level of autonomy given the average 43 minute light distance between Earth and Europa, the limited communication throughput of 80 kbps, and the likelihood of a short mission lifetime, most likely between 20 and 40 Earth days [3].

Given limited Europa surface imagery (no less than 6 m per pixel resolution), the surface topology is largely unknown

<sup>1</sup>All authors are with the Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena CA 91109 [william.reid@jpl.nasa.gov](mailto:william.reid@jpl.nasa.gov) ©2020, California Institute of Technology



Fig. 1: The *RoboSimian* rover traversing a salt-evaporite field at the Devil's Golf Course in Death Valley, CA.

on the mobile robot scale. However, terrestrial Europa analogue surfaces have been identified. These include icy glacial terrain [4], salt-evaporites [5], penitentes formed from ice sublimation [6], regolith [7] and icy chaos terrain [8]. In this study, salt-evaporite terrain at the Devil's Golf Course in Death Valley, CA is selected as an analogue terrain to be traversed. This site, as shown in Figures 1, 3, 4, and 6 is populated with unstructured nodules and cracked salty crusts.

Traversing over such terrain presents unique challenges given the high probability of getting mobility elements of the robot caught on irregular terrain geometries. The mobility challenge is compounded by the need to be autonomous. The system should be able to localize itself, plan safe and efficient paths to pre-defined waypoints, and execute traverses along these paths. In this work we propose a terrain adaptive wheel-on-limb mobility system that may traverse over terrain features up to 1.5 wheel diameters tall. Coupled with this mobility system is a localization, mapping, terrain evaluation, and motion planning autonomy stack that provides for navigation between user-defined waypoints.

Wheel-on-limb mobility systems have been demonstrated in both lunar and Martian analogue exploration applications in [9]–[11]. This class of mobility system provides energy efficient wheeled locomotion in benign environments and high-traversability limbed-locomotion in more challenging terrain. Planning for these high degree-of-freedom platforms is susceptible to the *curse of dimensionality*, which sampling-based motion planners seek to address [12]. Sampling-based planning algorithms such as RRT\* [13], FMT\* [14] and BIT\* [15] are almost-surely asymptotically optimal given a metric cost function. Application of such planners to autonomous traversal of planetary analogues with wheel-on-

limb platforms is described in [11] and [16]. The autonomy system presented in this paper differs in that it provides an autonomy system that does not rely on operator intervention to perform a long duration ( $>10$  m) traverse of unstructured terrain, and uses proprioceptive sensing to react to local terrain features that may not have been considered in the global plan.

The robotic system used in this work is a variant of the *RoboSimian* platform originally developed for the DARPA Robotics Challenge [17]. The robot’s mobility system has four limbs, each comprised of eight driven joints (six for limb articulation, one for wheel steering, and one for wheel driving). The limbs’ unique kinematic structure allows a versatile set of mobility modes such as driving with terrain adaptation, inchworming, and wheel walking. Demonstration of these modes along with an energy efficiency evaluation on salt-epaporite Europa analogue terrain is provided in [18]. *RoboSimian* adapts to the terrain using force and inertial measurement unit (IMU) proprioceptive sensing. Light Distance Active Ranging (LiDAR) is used to generate digital elevation maps (DEMs) of the surrounding terrain. These DEMs are fused with pre-generated global DEMs to form a global map in which the robot localizes itself.

The almost-surely asymptotically optimal ABIT\* sampling-based motion planner [19] is used to generate a path that has both its distance/energy minimized while remaining feasible. Path feasibility is determined as a function of terrain geometry and the limitations of the mobility system. The ABIT\* planner is suited to this problem given that it attempts to reduce the number of path feasibility checks. Feasibility checks for the *RoboSimian* planning problem are computationally expensive given the robot’s complex kinematic structure and terrain topology. The planner is adapted so that it may choose between two wheel rolling mobility modes. The first mode only actuates the wheel actuators and is therefore energy efficient, while the second actively articulates the limb joints while the wheels are being driven so as to adapt the platform configuration to the terrain.

The contributions of this work are (a) the development of a model-based wheel-on-limb motion planner that selects mobility modes based on an distance/energy cost function, (b) demonstration of an autonomous mobile robot navigating across highly unstructured Europa analogue terrain, and (c) evaluation of these navigation strategies over a variety of salt-epaporite terrain topologies. By coupling a highly capable proprioceptive-based terrain adaptive mobility system with exteroceptive navigation, we demonstrate for the first time a system capable of autonomously navigating Europa analogue terrain.

The paper presents global terrain evaluation and traversal strategies in Section II, while the local traversal methods are detailed in Section III. The entire navigation stack is evaluated through analogue terrain field trial experiments. A description of the experiment setup is provided in Section IV, while the experimental results are presented in Section V. Section VI provides discussion and conclusions.

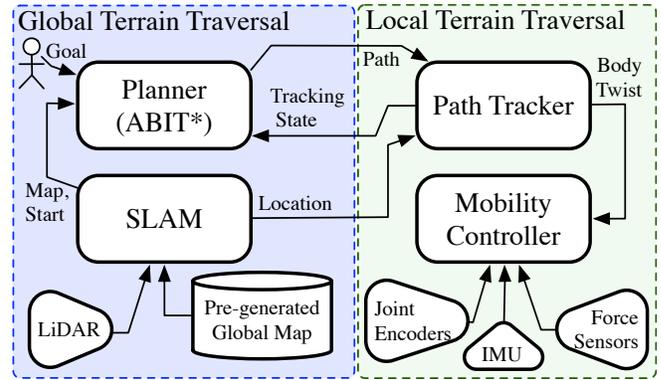


Fig. 2: System diagram of the *RoboSimian* autonomy system.

## II. GLOBAL TERRAIN TRAVERSAL

This section describes the on-board pipeline that the *RoboSimian* rover uses to navigate to operator-defined goal poses on a global map. An overview of the system can be seen in the left half of Figure 2. Before any navigation is performed a global map of the terrain is generated from a batch of LiDAR scans. The navigation system then initializes the onboard simultaneous localization and mapping (SLAM) module and registers a local onboard map to the global map. To navigate to a provided goal, the global motion planner generates a distance-efficient path of  $SE(3)$  poses from the robot’s current pose to its goal pose using the global map. During path traversals, the SLAM module continuously updates the position of the rover with respect to the local map.

### A. Pre-Generated Global Map

The use of a global map in this work is justified given the planned use of a high resolution LiDAR scanner in the proposed Europa Lander mission. As discussed in [3] and [20], Europa Lander would generate a  $100 \times 100$  m DEM at 5 cm per pixel during descent. All global maps generated in this work are produced by stitching together scans from a Leica BLK360 LiDAR scanner. This results in an approximate  $100 \times 100$  m voxel map at 5 cm voxel resolution for each site scanned. The map is transformed into a Euclidean Signed Distance Field (ESDF) map using the Voxblox library [21]. ESDF maps are 3D voxel maps that encode the signed distance to the surface for each voxel. The sign indicates if the voxel is inside or outside of a surface.

### B. SLAM

State estimation is provided by the Large-Scale Mapping and Positioning (LAMP) module [22]. This LiDAR-based SLAM implementation localizes point clouds generated from a Velodyne HDL-32E LiDAR (shown in Figure 3), computes scan-to-scan odometry, constructs a pose graph, and computes backend loop closures. As with the a-priori global map data product, the module produces a local ESDF map using the Voxblox library [21]. Global localization between this local map and the global a priori map is conducted using the libpointmatcher ICP library [23].

### C. Motion Planner

The RoboSimian autonomy system’s motion planner uses the ABIT\* sampling-based planning algorithm, which is tailored to the terrain-adaptive suspension capabilities of the rover. ABIT\* has been chosen given its ability to efficiently explore the rover’s state space and find safe paths to the goal. The planner is an almost-surely asymptotically optimal sampling-based path planning algorithm that makes use of anytime graph search techniques to quickly find initial solutions [19]. It also minimizes edge checks between sampled states compared with other state-of-the-art asymptotically optimal sampling-based planners. This makes ABIT\* suitable for the RoboSimian planning problem given the computational bottleneck is performing traversability checks between samples.

a) *ABIT\* and BIT\**: ABIT\* is based on the BIT\* algorithm. BIT\* begins by sampling a batch of rover poses from the global map. A search tree is initialized as a set of vertices composed of the sampled states, the goal states, and the current state as the tree’s root; and an empty set of edges. A priority queue is also initialized by expanding the start vertex and adding valid edges to its neighboring nodes. Construction of the tree then proceeds by iteratively expanding child vertices of the next vertex in the priority queue and removing this vertex from the queue. The priority queue is ordered given the summation of the cost-to-come value of an edge’s parent vertex, the cost between the parent and the child, and a heuristic cost to the goal. A vertex expansion proceeds by finding all of the valid edges to neighboring vertices and adding these edges to the priority queue. If a candidate edge has a child vertex that is already connected to the search tree and the cost of the candidate edge is less than the existing edge, then the tree is re-wired. If the goal vertex is included as a child vertex in the edge set, a path to the goal exists and this path is the resolution-optimal solution. Further expansion within this batch of samples then ceases. Anytime planning continues by sampling a new batch states and repeating the vertex expansion procedure. The new batch of samples is only taken from within a hyper-ellipsoid that encompasses the volume of states that may improve on the existing solution.

ABIT\*’s improvements on BIT\* are two-fold: (1) initial solutions within a sample batch are found faster given an inflated cost-to-go heuristic, and (2) the algorithm avoids the computationally expensive procedure of finding the resolution-optimal path in each batch of samples by stopping its search as soon as its found a solution that is within an inflation of the resolution-optimal cost. Any suboptimal paths found by ABIT\* are repaired in future planner iterations by keeping track of inconsistent states. An inconsistent state occurs when the cost-to-come value of a vertex lowers after it has already had edges to child vertices placed in the priority queue. Detailed descriptions of BIT\* and ABIT\* are given in [19].

b) *Planning Considerations for RoboSimian*: Within the RoboSimian autonomy system, planning with ABIT\*

involves sampling directly on the surface mesh pointcloud and ESDF map. Sampling involves randomly indexing the mesh pointcloud, which produces an  $\mathbb{R}^3$  point on the surface. This point is perturbed in a small random direction to avoid sampling only at the discretization of the mesh. The surface normal associated with the points neighboring the sampled point at the scale of the robot’s footprint is then calculated. A random rotation about this surface normal is generated and then checked for validity by checking to see if the rover may settle on the terrain at this pose.

Motion validation between two states  $\mathbf{x}_0$  and  $\mathbf{x}_1$  consists of first yawing  $\mathbf{x}_0$  to align with  $\mathbf{x}_1$  and checking states at a specified angular resolution. The straight line between the aligned state and  $\mathbf{x}_1$  is then projected onto the surface to generate a list of surface points at a specified linear resolution. These surface points are then settled and validated.

The rover settling procedure, used in both state sampling and motion validation, generates a feasible configuration consisting of an  $SE(3)$  rover pose,  $\mathbf{g}_R$ , and an  $\mathbb{R}^3$  position,  $\mathbf{p}_{w_i}$ , for each wheel  $i$ . The settling algorithm inputs are an  $\mathbb{R}^3$  point on the surface mesh of the ESDF map,  $\mathbf{p}_m$ , an  $SO(2)$  rotation,  $\theta$ , about the gravity vector, and the rover’s 2D footprint containing an  $\mathbb{R}^2$  position for each wheel.

Settling begins by placing the wheel footprint rotated by  $\theta$  to a specified distance above  $\mathbf{p}_m$ . The height of each wheel is then lowered until a surface contact is found. This is done by iteratively lowering wheel heights and querying the ESDF map for the distance from the center of each wheel to the closest terrain surface. The wheel model is approximated as a sphere with a contact threshold equal to the wheel’s radius. If the wheel is in unobserved space after attempting surface contact it is considered a failed settle.

Once  $\mathbf{p}_{w_i}$  is found for each wheel, the algorithm searches for a window of safe chassis heights. To begin, the limb with the highest contact point is found and the chassis height is set so that it complies with the minimum height saturation limit of this limb. Starting at this height, the chassis is checked for collisions with terrain. The height of the body is then iteratively raised until it is free of collisions. This becomes the minimum safe body height for this pose. Each wheel contact is then checked to make sure the respective limbs do not exceed the maximum saturation limits, or rather the furthest a limb can extend. If a maximum saturation limit is reached at the minimum body height, then the settle fails. The body height continues to iteratively rise until either the maximum saturation limit is reached or there are collisions from obstacles above the rover. This results in a minimum and maximum safe body height.

Another constraint that needs to be checked for is terrain overhangs that wheels may get stuck underneath. To account for these directional hazards in the planner, we make use of the gradient information in the ESDF map. This happens during motion validation of the trajectory of discrete poses between two states. At each step in the trajectory the planner examines the terrain below each settled wheel. An overhang is detected if there are voxels with gradients pointing to a surface above themselves. The planner checks these upward

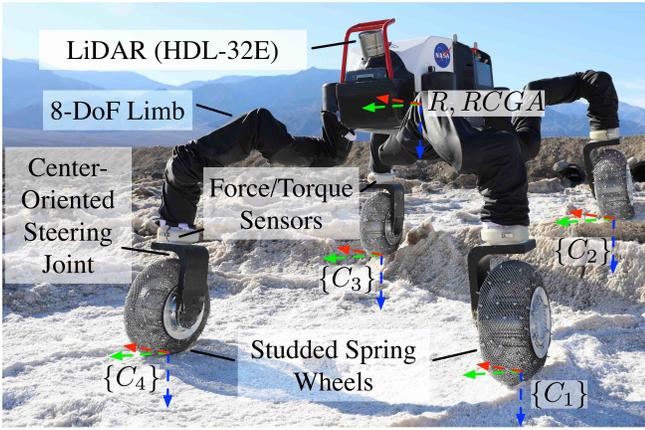


Fig. 3: Labeled view of the RoboSimian rover. The labelled coordinate frames include the rover frame  $\{R\}$ , the rover-centered-gravity-aligned frame  $\{RCGA\}$ , and the contact frames for each wheel  $\{C_i\}$  where  $i$  is the limb number.

facing gradients with the direction vector of the rover to determine if the wheel is in a hazardous state.

If all of these safety checks pass, then the state is assigned a mobility mode (actively articulated suspension on or off) based on the wheel contact points with the terrain. A minimum height offset between the minimum and maximum wheel contact heights is used as a threshold for enabling active suspension.

Overall, the planner seeks to minimize a weighted path distance. Each path segment distance is weighted by its associated mobility mode. The weighting is as an approximation of the expected energy usage of the mode. Upon completion, the planner returns a path of  $SE(3)$  poses that each have an associated safe body height window and mobility mode.

### III. LOCAL TERRAIN TRAVERSAL

Local terrain traversals for the RoboSimian platform on unstructured terrain rely on two main control modules: a mobility controller for local terrain adaptation while wheel driving, and a path tracker for driving between waypoints. These modules are summarized within the full architecture of the autonomy system in Figure 2.

#### A. Mobility Controller

The mobility controller for RoboSimian takes in body twists as inputs and drives the relevant limb and wheel joints to meet these twists. The mobility controller operates in two modes: with or without actively articulated suspension. Actively articulated suspension drives the limb joints as well as the wheels so that the limbs adapt to the terrain profile that is being traversed. This control strategy relies only on proprioceptive feedback. It uses force sensors positioned at the steering joint of each limb (shown in Figure 3) along with an inertial measurement unit (IMU) and limb joint encoders. When not using actively articulated suspension the limb joints are held stationary while the wheels are driven.

This mobility mode is efficient on benign terrain, however it becomes unsafe as the local terrain heights between wheel/ground contact points increase in variance.

The objectives of the actively articulated suspension motion controller are to keep all four wheels in contact with the terrain and to keep the roll and pitch of the rover's chassis at zero, relative to the gravity vector. The wheel-contact objective ensures that each of the four wheels maintain traction. The orientation objective makes sure that the robot's LiDAR is close to level throughout a traverse.

As shown in Figure 3, each limb touches the terrain at a wheel/ground contact point defined by  $C_i$ , where  $i$  is the associated limb number. Additionally, a rover chassis frame  $R$  is located at the center of the robot's chassis, while a rover-centered-gravity-aligned coordinate frame  $RCGA$  is coincident with  $R$  and has a  $z$ -axis that is aligned with the gravity vector. Motion of the wheel contact frame  $C_i$  relative to the rover chassis frame  $R$  is constrained along a plane that is parallel to the  $xz$ -plane of the  $R$  frame.

The multi-objective motion controller is formulated with the following control policy:

$$v_{EE,z}^{RCGA} = k_f(f_{z,ref}^{RCGA} - f_{z,meas}^{RCGA}) + k_o(o^{RP}) + k_\ell(\ell_{err}), \quad (1)$$

where  $f_{z,ref}^{RCGA}$  is a  $4 \times 1$  vector of the expected vertical limb end-point forces for a quasi-static system expressed in the  $RCGA$  frame,  $f_{z,meas}^{RCGA}$  is a  $4 \times 1$  vector of the vertical load cell force measurements projected into  $RCGA$ ,  $o^{RP}$  is a  $4 \times 1$  vector of the limb end-point vertical deflections required to keep the roll and pitch at their desired angles, and  $\ell_{err}$  is a  $4 \times 1$  vector of distances the limb end-points have to move to maximize distance away from suspension saturation across all limbs. Each  $k$  is a  $1 \times 4$  vector of control gains. The output  $4 \times 1$  velocity vector  $v_{EE,z}^{RCGA}$  contains the desired velocities of each of the wheel-contact frames relative to the  $z$ -axis of  $RCGA$ . These limb-workspace velocities are mapped to their corresponding limb joint velocities by interpolating over pre-generated lookup tables. The lookup tables constrain motion of the wheel-contact frames to a plane that is parallel to the  $xz$ -plane in  $R$  and  $RCGA$ . A detailed explanation of this controller is provided in [18] and is based on a similar wheel-on-limb controller described in [10].

#### B. Path Tracker

A path generated by the path planner (described in Section II-C) is composed of a set of waypoint  $SE(3)$  poses relative to a global reference frame. The role of the path tracker is to direct the robot towards the next waypoint pose and decide if a waypoint pose has been reached. The path tracker is implemented as a state machine with three possible states: (1) steering, (2) driving, and (3) waypoint reached. In the steering state the robot performs a turn on the spot until the robot is pointing its  $x_R$ -axis at the goal position. During the driving state the robot drives forward (along its  $x_R$ -axis) until its  $R$  frame is coincident with the waypoint position or the yaw error has exceeded a pre-defined threshold. While driving the robot may drive its

wheels with actively articulated suspension on or off. Once the waypoint has been reached the path tracker returns to the steer state if there is a new waypoint to follow, otherwise path tracking is complete. This path tracker has been selected for its simplicity of implementation. A drawback of the method is excessive steering when the rover is in close proximity to a waypoint pose. In future work, a model-predictive controller, similar to [24], will be implemented to reduce unnecessary deviations from the path.

#### IV. EXPERIMENT SETUP

This section introduces the experiments used to evaluate the performance of the RoboSimian autonomy system on salt-evaporite terrain. A total of 38 planning and path execution trials were attempted. Each trial was started with an operator giving the robot one or multiple goal positions. The robot planned a path to the goal and then executed the path. For each trial the planner was given between 45 s and 120 s to generate a path, while the average time required to find an initial solution was 1.8 s. When the robot reached its final goal position the trial ended. If for any reason the robot was unable to complete the path, the trial ended with a path following failure.

All trials were conducted on one of three terrain sites, each selected for their varying terrain topologies. Site 1 was the most benign area, site 2 the roughest area, and site 3 an area full of ridges and overhanging obstacles. Site 1 was chosen as a baseline case, while Site 2 and 3 were chosen as challenging sites that would expose shortcomings in the proposed planning and control strategies. Prior to any of the traverses, a global map of each site was generated from LiDAR scans. Figure 4 shows images and pre-generated maps associated with each of the sites.

In all trials the robot's traverse range was limited to a radius of 20 m as the robot was powered over a 20 m long tether. Power was provided over the tether by two gasoline generators. An operator persistently followed the robot so as to prevent the tether from snagging on terrain or from being driven over by the robot.

To evaluate the performance of the autonomy system, the following metrics are used: *translational and rotational distance*, *disengagement rate*, and *energy efficiency*.

##### A. Translational and Rotational Displacement

Aside from measuring how far the robot travelled, translational and rotational distance travelled are used to evaluate the effectiveness of using a pre-generated global map versus only relying on a map generated from local LiDAR data. Translational distance,  $d$ , is the distance the rover frame moves, while rotational distance,  $d_{rot}$ , is calculated using the distance travelled by the wheel attached to limb 1:

$$d = \sum_{m=1}^M \|\Delta \mathbf{p}_R^G\|, \quad d_{rot} = \sum_{m=1}^M (\Delta \psi_R^G \|\mathbf{p}_{C_1,xy}^R\|),$$

where  $m$  is the timestep number,  $\Delta$  is the difference between a value at the current and previous timestep,  $\mathbf{p}_{C_1}^G$  is the position vector of the wheel 1 contact frame relative to the

global frame,  $\psi_R^G$  is the yaw of the robot relative to the global frame, and  $\mathbf{p}_{C_1,xy}^R$  is the  $x$  and  $y$  components of the position vector of the wheel 1 contact frame relative to the robot frame. The distance of wheel 1, as opposed to the rover frame, is used given that the rotational motion contributes to an arc distance travelled by the wheel.

An additional distance measure is the root mean squared error (RMSE) between the planned and actual wheel contact positions. RMSE values over entire trajectories are calculated to evaluate how well planned paths are followed at the various test sites.

##### B. Disengagement Rate

Disengagement rate measures the number of failures per distance travelled. This metric is commonly used in autonomous car performance evaluation [25]. Our disengagement rate has the units number of failures per 10 meters. A failure is defined as any event that causes the robot to not safely proceed along its traverse. Typical failures observed during the experiment traverses included wheel hubs snagging on terrain, a wheel being unable to overcome overhanging or vertical terrain features, or hardware faults such as overheating. This provides a measure for how well the autonomy system performs on a given terrain.

##### C. Energy Efficiency

Energy efficiency measures the energy the robot expended per distance travelled. This is used to evaluate whether mobility mode switching affects efficiency. This metric is also used as an estimate of how challenging the terrain is given that more limb articulation and therefore more energy is required to traverse increasingly challenging terrain. Energy efficiency,  $\eta$ , is calculated as the summation of each limb joint actuator's power output for each traverse time-step divided by the total distance traversed. The calculation is

$$\eta = \sum_{m=1}^M \sum_{j=1}^J (I_{j,m}^2 w_j + I_{j,m} k_{emf,j} N_j \omega_j) (t_m - t_{m-1}) / d, \quad (2)$$

where  $M$  is the number of time-steps,  $J$  is the number of active joints in the robot,  $I_{j,m}$  is the motor current at joint  $j$  and time-step  $m$ ,  $w_j$  is the winding resistance of motor  $j$ ,  $k_{emf,j}$  is the back-emf constant of motor  $j$ ,  $N_j$  is the gear ratio of actuator  $j$ ,  $\omega_j$  is the angular rate of motor  $j$ , and  $t_m$  is the time at time-step  $m$ . The power cost of computation and sensing is not considered in this metric. These "hotel costs" average 880 W for the RoboSimian rover during typical operations.

#### V. EXPERIMENT RESULTS

Results from each of the experiment trials are first presented through three illustrative trials. These trials have been selected given that they took place on distinct terrain sites and that there were no traverse failures. Figure 5 shows the planned path ground track versus the executed path for a trial at each of the three terrain sites. Additionally, a planned wheel height is plotted against the actual wheel height for

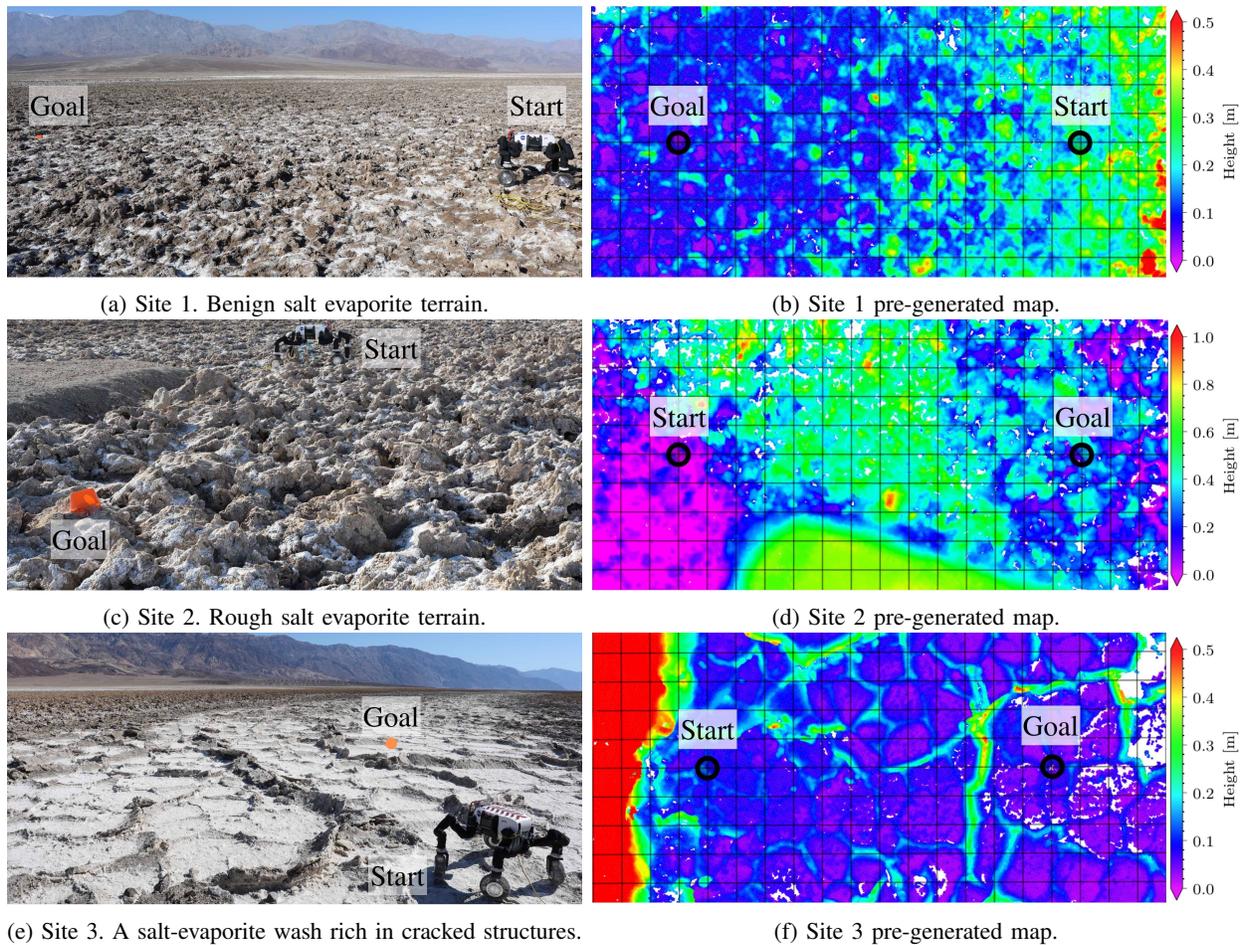


Fig. 4: The terrain, start/goal positions, and elevation maps for the three test sites. (Best viewed in color.)

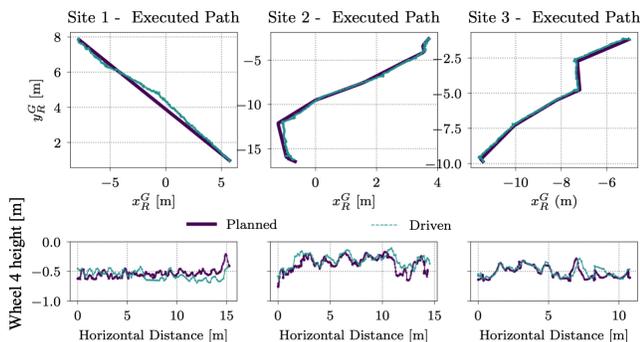


Fig. 5: Illustrative trials from the three terrain sites are highlighted in each column of plots. The top row shows the planned and executed paths of the rover frame  $R$  relative to a global frame  $G$ . The bottom row shows the planned versus executed contact point heights for wheel 4.

each of the trials. During the trial at site 1 it is observed that the robot veers away from the path almost halfway through the traverse. It is also noted that the planned and executed wheel heights differ during this period of path deviation. The discrepancy between the path tracker and the planned path

is attributed to the robot slipping on a large terrain feature at coordinate  $(-5\text{ m}, 6\text{ m})$ , causing the robot to yaw. For the remaining two trials at sites 2 and 3, the roughly consistent agreement between both the path track and wheel height plots demonstrate that the path tracker and SLAM modules performed nominally. Trials at sites 2 and 3 may be viewed in the attached video.

To further evaluate the performance of the autonomy system we look at the performance of the rover with respect to each of the metrics from Section IV. The data used in this evaluation is presented in Table I.

In total, 435.3 m of paths were traversed. The maximum distance driven in a single trial was at site 3 during a multiple goal drive of 31.0 m. RMSE values for total wheel-contact point deviation between planned and actual paths across all trials were 0.14 m, 0.11 m, and 0.11 m for sites 1, 2 and 3 respectively. This demonstrates a consistent path following ability no matter the terrain roughness. Distance driven data also shows the discrepancy between the robot having or not having global terrain knowledge. At site 1 two trials were run with only local terrain knowledge. The resulting executed paths featured a larger number of turning maneuvers than the trials with global terrain knowledge. This resulted in

significantly larger rotational distance of 57.4 m compared with 2.6 m when global terrain knowledge was used.

The average energy efficiency of the robot were 30.4 kJ/m, 58.6 kJ/m and 54.2 kJ/m for sites 1, 2, and 3 respectively. The discrepancy between the energy data at site 1 compared to sites 2 and 3 is expected given that there are many more challenging salt-evaporite structures at sites 2 and 3. Site 2 had denser spacing of salt-evaporite structures than site 3, which may explain the slightly larger value for energy efficiency at site 2. Contrary to predictions, the trials that used the two mobility modes were less energy efficient than those that only used actively articulated suspension. At site 1 the average energy efficiency when using both modes was 28.3 kJ/m versus 23.3 kJ/m when only using one. At site 2 the relative discrepancy was similar with an average efficiency of 61.5 kJ/m for both modes against 57.6 kJ/m for one. This counter-intuitive behavior resulted from there being limited areas where driving with actively articulated suspension was feasible according to the traversability analysis performed. To force the planner to switch between modes

TABLE I: The number of trials, total distance travelled, average energy efficiencies, disengagement rates and planner parameter definitions for each traverse site.

	Planning Parameters						Total	Avg.	
	Single/Multiple Goals	Global			Local				
		Single	Multi.	Single					
Site	Num. Modes.	1	2	1	1				
1	Num. Trials	2	2	0	2	6	-		
	Dist. [m]	33.5	31.7	-	34.5	99.7	-		
	Rot. Dist. [m]	<b>1.28</b>	<b>1.36</b>	0	<b>57.35</b>	60.0	-		
	Num. Failures	0	0	-	0	0	-		
	Disengagement Rate [fail/10m]	0.0	0.0	-	0.0	-	<b>0.0</b>		
	Avg. Eng. Eff. [kJ/m]	<b>23.3</b>	<b>28.3</b>	-	39.6	-	<b>30.4</b>		
2	Num. Trials	9	3	0	0	12	-		
	Dist. Driven [m]	64.5	32.7	-	-	97.2	-		
	Rot. Dist. [m]	13.4	9.7	-	-	23.2	-		
	Num. Failures	8	2	-	-	10	-		
	Disengagement Rate [fail/10m]	1.24	0.61	-	-	-	<b>0.93</b>		
	Avg. Eng. Eff.	<b>57.6</b>	<b>61.5</b>	-	-	-	<b>58.6</b>		
3	Num. Trials	13	0	7	0	20	-		
	Dist. Driven [m]	131.4	-	107.0	-	238.4	-		
	Rot. Dist. [m]	141.2	-	57.0	-	207.6	-		
	Num. Failures	7	-	7	-	14	-		
	Disengagement Rate [fail/10m]	0.53	-	0.65	-	-	<b>0.59</b>		
	Avg. Eng. Eff.	52.0	-	58.3	-	-	<b>54.2</b>		
Total Number of Failures						<b>24</b>	-		
Total Number of Trials						<b>38</b>	-		
Total Translational Distance [m]						<b>435.3</b>	-		
Disengagement Rate [fail/10m]						-	<b>0.64</b>		



(a) A hub snag.

(b) Terrain collapse.

Fig. 6: Two examples of terrain features that caused failures.

at all, a five times cost penalty was assigned to driving with actively articulated suspension relative to driving without it. The actual energy cost penalty between the two modes is closer to two times according to collected energy data.

The disengagement rate data demonstrates that the autonomy system performed well in the benign terrain of site 1 with almost 100 m travelled with no failures. Sites 2 and 3 were more challenging to traverse with average disengagement rates of 0.93 failures/10 m for site 2 and 0.59 failures/10 m for site 3.

Out of the 38 trials, there were 14 completed trials and 24 trials that ended in failure. Two of these failures were due to hardware overheating, 12 were due to wheel hubs snagging on terrain, nine trials failed due to a wheel getting stuck on an overhanging or vertical terrain feature, and one failure was due to a wheel collapsing the terrain and then falling into the resulting hole. Figure 6 shows images of an example wheel hub snag as well as the collapsed terrain failure.

The hub snag failures are challenging given that small terrain protrusions (Figure 6a) can result in traverse ending events. This failure case was observed in our previous work summarizing traverses on salt-evaporite terrain [18], which resulted in the re-design of the hub joint. The re-design reduced the surface area of the hub orthogonal to the wheel axis. This re-design reduced the frequency of hub-snag events, however they are still an impediment. Future work will explore improving the wheel/hub collision model that is used in the motion planner. Additionally, we plan to integrate a local traversal controller that stops and/or backs the rover away when a high current draw of a hub-snag is recorded. It is envisioned that after recovering from such an event the planning process would start over again. This reactive control and re-planning strategy may also be effective in addressing failures caused by overhanging or vertical terrain features. To address an event similar to the terrain collapse failure, transitioning to a walking gait where the affected wheel is lifted may be a promising reactive control strategy.

## VI. CONCLUSIONS

A full autonomy stack for robotic traversal of highly unstructured Europa analogue terrain is demonstrated and evaluated. This work outlines a system that is moving toward

an autonomous planetary exploration system that can access terrain that is not traversable by the current state-of-the-art in planetary exploration rovers. Illustrative examples of the robot localizing, planning, and executing paths are given for sites with varying levels of traverse difficulty. These examples show a system that is able to leverage the active articulation of the RoboSimian platform to find safe and distance-efficient paths.

The planning strategy also provides autonomous selection of mobility modes so as to increase energy efficiency in benign terrain. The experiments performed demonstrated this ability, however energy efficiency data did not show decreased energy usage when an executed path switched to driving without actively articulated suspension. Further investigations of mobility mode switching given either a more liberal traversability analysis and/or flatter terrain are required. Additionally, future work will consider switching to limbed mobility modes like inchworming or wheel walking, which have been shown to reduce slip when traversing highly sloped terrain [26].

It is shown that global pre-generated maps assist the autonomous motion planner in being able to plan ahead. For the case of a mobile agent on the surface of Europa, it is safe to assume that data from the rover's landing system would be used to formulate this global map. According to [20], a descent imaging LiDAR would provide coverage of a 100 m×100 m area around the lander with less than 5 cm resolution per pixel.

The disengagement rates observed show that more robust planning and control strategies are required when traversing more challenging terrain sites. As we observed in [18], susceptibility to hub-snag events is a major cause of failure in rough terrain. More sophisticated reactive control strategies and refinement of the motion planner's collision models are needed to increase robustness to these failures.

Autonomous robotic navigation of dense and unstructured salt-evaporite Europa analogue structures is a challenging task. Our experimental evaluations help highlight some of the unique mobility challenges presented by this task. The tightly coupled set of control, perception, and planning strategies developed in this work are also an important contribution toward the realization of a safe and *self-reliant* method for exploring the European surface, one of the most important sites for exploration in our solar system.

## VII. ACKNOWLEDGEMENTS

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). The information in this document is predecisional and is provided for planning and discussion only.

## REFERENCES

- [1] Space Studies Board, National Research Council, *Vision and voyages for planetary science in the decade 2013-2022*. National Academic Press, 2012.
- [2] C. B. Phillips and R. T. Pappalardo, "Europa Clipper mission concept: Exploring Jupiter's ocean moon," *Eos Trans. AGU*, vol. 95, no. 20, pp. 165–167, 2014.
- [3] K. P. Hand *et al.*, "Europa Lander study," NASA JPL, Tech. Rep. JPL D-97667, 2016.
- [4] R. Greeley, R. Sullivan, *et al.*, "Europa: Initial Galileo geological observations," *Icarus*, vol. 135, no. 1, 1998.
- [5] T. McCord, G. Hansen, *et al.*, "Salts on Europa's surface detected by Galileo's near infrared mapping spectrometer," *Science*, vol. 280, no. 5367, pp. 1242–1245, 1998.
- [6] D. E. J. Hobbey, J. M. Moore, *et al.*, "Formation of metre-scale bladed roughness on Europa's surface by ablation of ice," *Nature Geoscience*, vol. 11, no. 12, 2018.
- [7] B. Buratti and M. Golombek, "Geologic implications of spectrophotometric measurements of Europa," *Icarus*, vol. 75, no. 3, 1988.
- [8] G. Collins and F. Nimmo, *Europa*. The University of Arizona Press, 2009, ch. Chaotic terrain on Europa, pp. 259–281.
- [9] B. H. Wilcox, T. Litwin, *et al.*, "ATHLETE: A cargo handling and manipulation robot for the Moon," *J. Field Rob.*, vol. 24, no. 5, 2007.
- [10] F. Cordes, F. Kirchner, and A. Babu, "Design and field testing of a rover with an actively articulated suspension system in a mars analog terrain," *J. Field Rob.*, vol. 35, no. 7, pp. 1149–1181, 2018.
- [11] W. Reid, R. Fitch, *et al.*, "Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover," *J. of Field Rob.*, vol. 37, no. 5, pp. 786–811, 2020.
- [12] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [14] L. Janson, E. Schmerling, *et al.*, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Rob. Res.*, vol. 34, no. 7, pp. 883–921, 2015.
- [15] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Batch Informed Trees (BIT\*): Informed asymptotically optimal anytime search," *Int. J. Rob. Res.*, 2020.
- [16] V. SunSpiral, D. Wheeler, *et al.*, "Development and field testing of the foothill planning system for the ATHLETE robots," *J. Field Rob.*, vol. 29, no. 3, pp. 483–505, 2012.
- [17] S. Karumanchi, K. Edelberg, *et al.*, "Team RoboSimian: Semi-autonomous mobile manipulation at the 2015 DARPA Robotics Challenge finals," *J. Field Rob.*, vol. 34, no. 2, 2016.
- [18] W. Reid, G. Meirion-Griffith, *et al.*, "Actively articulated wheel-on-limb mobility for traversing Europa analogue terrain," in *FSR*, 2019.
- [19] M. P. Strub and J. D. Gammell, "Advanced BIT\* (ABIT)\*: Sampling-based planning with advanced graph-search techniques," in *IEEE ICRA*, 2020.
- [20] E. Schindhelm, R. Rohrschneider, *et al.*, "A scanning LiDAR system for active hazard detection and avoidance during landing on Europa," in *IEEE AeroConf*, 2018.
- [21] H. Oleynikova, Z. Taylor, *et al.*, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *IEEE/RSJ IROS*, 2017.
- [22] K. Ebadi, Y. Change, *et al.*, "LAMP: Large-Scale Autono Mapping and Positioning for exploration of perceptually-degraded subterranean environments," in *IEEE ICRA*, 2020.
- [23] F. Pomerleau, F. Colas, *et al.*, "Comparing ICP Variants on Real-World Data Sets," *Auto. Robot.*, vol. 34, no. 3, pp. 133–148, 2013.
- [24] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Rob. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016.
- [25] Google. (2015) Google self-driving car testing report on disengagements of autonomous mode. [Online]. Available: <https://tinyurl.com/reys8x>
- [26] W. Reid, S. Karumanchi, *et al.*, "Mobility mode evaluation of a wheel-on-limb rover on glacial ice analogous to Europa terrain," in *IEEE AeroConf*, 2020.