REASON-RECOURSE Software for Science Operations of Autonomous Robotic Landers

Jay McMahon
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
jay.mcmahon@colorado.edu

Nisar Ahmed
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
nisar.ahmed@colorado.edu

Morteza Lahijanian
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
morteza.lahijania@colorado.edu

Peter Amorese
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
peter.amorese@colorado.edu

Taralicin Deka
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
taralicin.deka@colorado.edu

Karan Muvvala
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
karan.muvvala@colorado.edu

Kian Shakerin
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
kian.shakerin@colorado.edu

Trevor Slack
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
trevor.slack@colorado.edu

Shohei Wakayama
University of Colorado Boulder
3775 Discovery Drive
Boulder, CO 80303
shohei.wakayama@colorado.edu

Abstract— **Future planetary exploration missions on the surface of distant bodies such as Europa or Enceladus can't rely on human-in-the-loop operations due to time delays, dynamic environments, limited mission lifetimes, as well as the many unknown unknowns inherent in the exploration of such environments. Thus our robotic explorers must be capable of autonomous operations to ensure continued operations and to try to maximize the amount and quality of the scientific data gathered from each mission. To advance our technology toward this goal, we are developing a system to maximize the science obtained by a robotic lander and delivered to scientists on Earth with minimal asynchronous human interaction. The autonomy architecture consists of two main components: REASON (Robust Exploration with Autonomous Science on-board) and RECOURSE (Ranked Evaluation of Contingent Opportunities for Uninterrupted Remote Science Exploration) for efficient and useful scientific communication between scientists and robot. The key advantage to this design is in its ability to continuously operate and adapt despite the constraints of high-latency, low-bandwith communications and an uncertain environment which today would require ground-in-the-loop operations. This paper presents the initial version of the REASON-RECOURSE system. Details of the implementation of the scientist interface and interaction with the ground software - RECOURSE - and the on-board robotic planning software - REASON - are given. Demonstration of the operation using NASA's OceanWATERS testbed will be shown.**

TABLE OF CONTENTS

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

1. INTRODUCTION

Science operations with remote autonomy currently focus on managing highly limited vehicle resources as well as operational impacts from direct and indirect task couplings. This has led to an increased dependency on ground-based human analysis and decision-making in such missions. Automated software tools and human-machine interfaces for operational planning and decision-making support have cut down on the overall effort required. Examples include autonomous science data collection software suites, e.g. AEGIS for MER [1], [2], smart geological feature detectors [3], [4], [5], AI-based activity planners and scheduling systems like ASPEN, CASPER, OASIS, [6], [7], [8], [9] and for Perseverance [11], [10], [9], [12], [13].

There have been various levels of autonomy implemented in spacecraft systems over the years, ranging from guidance navigation and control applications (e.g. [16], [14], [19], [15], [18], [17]) to autonomous operations and science (e.g. [21], [20], [22]). On deep space scientific missions, like OSIRIS-REx [28], Rosetta [30], [29], or New Horizons [31], [33], [32], [34] considerable expense and effort is put into planning observations on the ground and uploading them to the spacecraft which are then executed on-board in an open-loop framework with minimal autonomy. If something abnormal occurs, observations are missed or the spacecraft may enter safe mode which causes significant operation delays.

Overall, capable systems for particular autonomous tasks already exist and they let ground operations manage the traditional 'single-pathway' science plans in relatively well-understood environments. However, such systems are not extendable to missions in remote unexplored environments like ocean worlds and icy moons, where opportunistic multi-contingency autonomous on-board decision-making is needed in the face of evolving uncertainties and science. Furthermore, a fundamental issue missing in the state-of-the-art is a way for scientists to be able to naturally "talk" to their robotic counterparts without the barrier of complex sequencing and command interfaces.

In previous work [58], we outlined our proposed solution that seeks to address the main issues that prevent current work from being put into use on an ocean worlds exploration lander. We showed the design of an autonomy architecture that consists of two main components: **REASON** (Robust Exploration with Autonomous Science on-board) and **RECOURSE** (Ranked Evaluation of Contingent Opportunities for Uninterrupted Remote Science Exploration) which operate in tandem using for efficient and useful scientific communication between scientists and robot. In particular, the combination of REASON and RECOURSE has been explicitly designed to successfully handle high-latency, low-bandwidth communications while incorporating scientists' inputs and continuously gathering and downlinking valuable science data. Our approach focuses on creating a near-future flight system implementation that provides confidence and transparency about how the autonomous system is performing and the ability for operators to update the performance if desired. The REASON-RECOURSE system strives to prevent safe-modes from stopping science operations, since it always has further actions and plans prioritized and ready to execute.

This paper follows [58] and documents our current progress in developing and demonstrating our proposed architecture. As such, we discuss the current development state in each of the main components - REASON and RECOURSE - and show results of simulations demonstrating it's use in NASA's OceanWATERS virtual testbed.

2. REASON

REASON (Robust Exploration with Autonomous Science on-board) is the on-board component of our proposed autonomy architecture. REASON is designed to realize the system's potential while providing assurances on the system's performance. To this end, we develop a synergistic framework where we use a Formal Language to specify high-level descriptions of tasks, and the framework intelligently computes a sequence of discrete actions (policy) to be executed by the low-level motion planning framework to plan a continuous trajectory that is safe. Additionally, a middle layer informs the high-level planner about the status of low-level execution and updates the plan if needed [58].

For example, a mineralogist may come up with the task: "Drill into the soil and collect samples in location 1.The location must always be clear of rocks before drilling.". The task can be expressed as a Boolean formula:

$$\phi = F(p_{drill} \wedge F(p_{sample})) \wedge G(\neg p_{rock}) \tag{1}$$

Here F and G are temporal operators interpreted as "Eventually" and "Globally", and $\wedge$ and $\neg$ is boolean conjunction and negation, respectively. The formula ϕ is expressed over atomic propositions (Boolean predicates) that indicate the current configuration of the world. Satisfaction of task ϕ translates to accomplishing the task in the real world. Using this approach allows us to express tasks succinctly and leverage theories and algorithms developed by the Formal Methods community to provide task completion guarantees. Executing the task also yields some valuable data, which must then be downlinked within a tight bandwidth constraint. Figure 1 visualizes this process.

Task Planner—It is especially important to be able to provide guarantees on the autonomous behavior for a remote mission. Guaranteed behavior is two-fold, where firstly, the definition of a task must be completely unambiguous, and secondly, the planner must produce a behavior that is guaranteed to satisfy each task. To describe unambiguous temporal task specifications, we use Co-Safe Linear Temporal Logic (scLTL) [57] formulae. scLTL allows us to specify tasks that can be accomplished in finite time and thus is a natural choice to express high level tasks. To provide guarantees on the autonomous behavior of the system with respect to scLTL formulae, we leverage formal synthesis techniques applied to a task-planning/motion-planning architecture.

In order for the task planner to combine a task specification with the physical constraints of the autonomous system and environment, a discrete system model must be provided. This model, referred to as the *system abstraction* [46] [47], should reason over discrete states that include information about the status of the lander, as well as parts of the environment that the lander interacts with. Transitions between states in this model determine the physical capability and constraints of the system. These transitions can take the form of discrete actions, referred to as an *motion primitive*.

Using both the temporal logic task specification as well as the system abstraction, the task planner determines a sequence of high-level actions that will take the system from the current (initial) state to a final state in a manner that satisfies the given task specification without violating any task-specific and physical constraints of the system.

Motion Planner—The motion planner (Low-Level Planner) is responsible for low-level continuous path planning. The lander may have tools equipped that have moving parts and require low-level controls. The purpose of the motion planner is to determine a continuous trajectory for a tool to follow that will prevent collision with other parts of the lander itself, or obstacles in the environment. Using a motion planner can provide safety guarantees for the determined trajectory. The lander concept being used in the OceanWATERS (Ocean Worlds Autonomy Testbed for Exploration Research & Simulation, [41]) comes equipped with a robotic manipulator, and a movable camera and communication module. For safely controlling the robotic manipulator, it is assumed that the motion planner is provided with information of obstacle boundaries, including collision boundaries of the lander. The motion planner will look for a collision free solution between the start configuration and a goal configuration. A sampling-based motion planning algorithm, such as RRT, is used to determine the trajectory for the manipulator [48] [49]. For the camera and communication module, a simple linear trajectory can be implemented that can pan and tilt the apparatus to the desired pose, while staying within the safety limits. For the purposes of this architecture, it is assumed that the movable tools on the lander also have stable low-level controllers that can safely follow a desired trajectory.

Preference Planning Framework—We extend this framework to reasoning over multiple tasks at once. To maximize the productivity of the mission, the scientists may task the on-board autonomy with multiple scientifically related or unrelated objectives. However, certain scientific objectives may take priority over others. We propose a preference planning framework that studies how to satisfy all tasks in a way that adheres to the scientists' preference, as well as how to maximize overall efficiency.

For this architecture, we simplify the general notion of *preference* to simply a preferred order of satisfaction. For example, tasks related to mineralogy may take precedence over photographic tasks, and thus the planner should prioritize satisfying the mineralogy task before the photographic task.

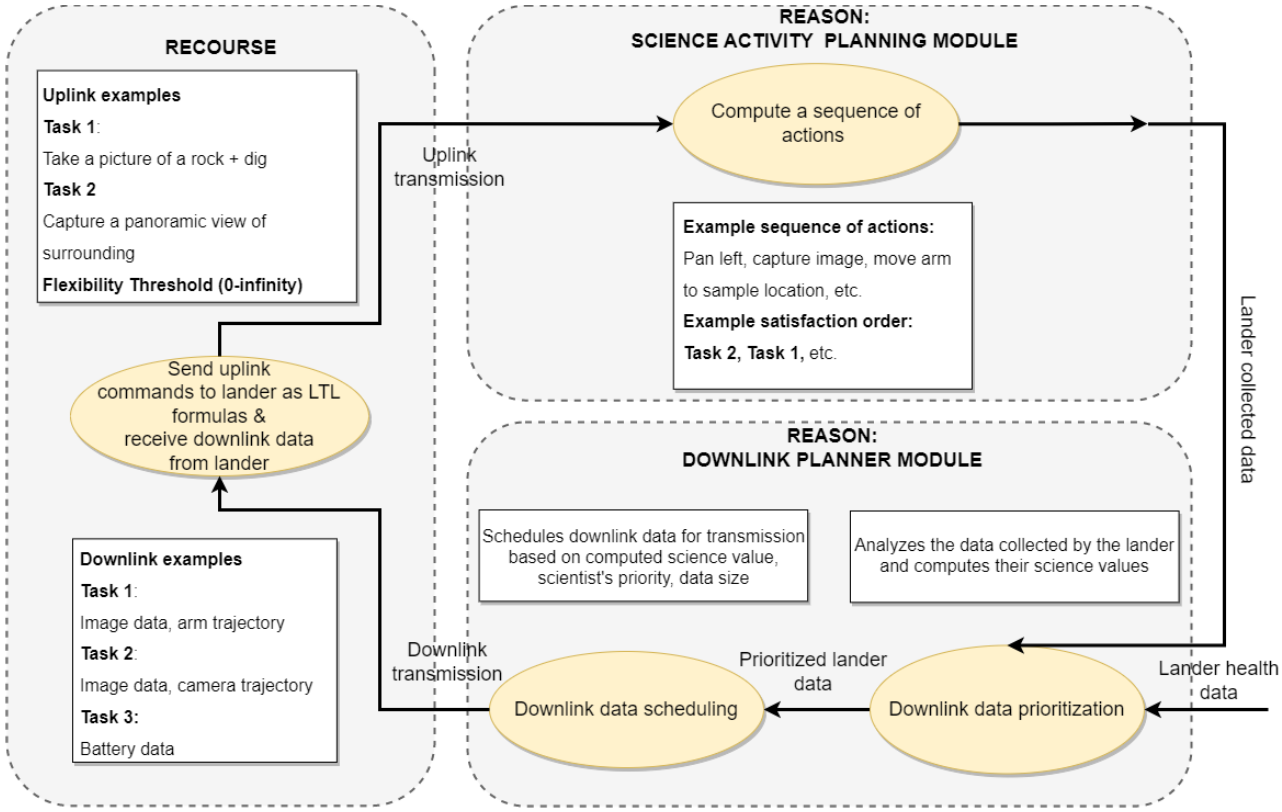


Figure 1: Data flow between different components of the proposed autonomous system

The preference planner takes in an ordered set of scLTL task formulae $\Phi = \{\phi_1, \phi_2, ..., \phi_N\}$ where each formula is ordered relative the desired order of satisfaction for a given activity segment. Likewise, the users specify a *flexibility* threshold representing the maximum allowable deviation from desired ordering. The planner will then determine the most efficient plan within the given flexibility. For example, if the scientists care heavily about satisfying the tasks in the desired order, they may set a very low flexibility threshold. Conversely, if the scientists want the planner to prioritize finishing all of the tasks in the most efficient manner, they may set a very high flexibility threshold.

The search space for this planning problem is very large due to capturing all relevant history towards completing each task. The proposed planning framework leverages graph search techniques to compute an optimal plan given the flexibility threshold constraint. However, performing a graph search on a very large search space is very computationally and memory intensive. Likewise, the planning scenarios for this framework assume that computation is done on-board the lander, which may have limited computational resources. To address this issue, we leverage a problem-agnostic heuristic to greatly speed up the computational load of computing plans, as well as the consumed memory.

Downlink planner

The goal of the downlink planner is to maximize the science information sent back to Earth, while successfully handling the high-latency and low-bandwidth communication limitations between the lander and the Earth-based ground stations. This module is responsible for two main tasks on-board the lander: Downlink data prioritization and Downlink data scheduling.

Downlink data prioritization—After the lander has collected data according to the plan generated by the preference planner (as described earlier), the downlink planning module will compute the science value of the collected data on-board. This process is called downlink data prioritization. Downlink prioritization can be achieved by methods such as *Target Signature* and *Novelty detection* methods [53], [54], [55]. Methods for downlinking selective data or processed data (such as *Image Masking*) are also being considered to reduce downlink bandwidth requirements [54], [56]. It is to be noted that, for initial demonstration purposes in OceanWATERS, we assume that scientific value of the data collected by the lander is known. Thus, in the work presented in this paper, downlink data prioritization module is considered inactive. In future work, this assumption of knowing the scientific value of data collected by the lander will be released. In such cases, there needs to be a balance between scientist's data preference and the data scientific value to compute the net downlink data value. This is briefly addressed in our previously published work [58].

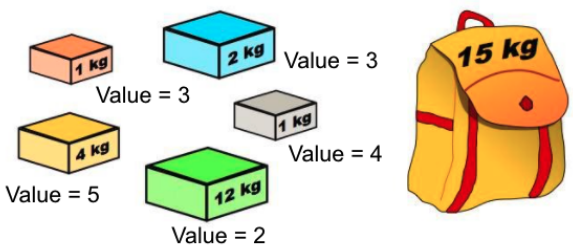
Downlink data scheduling— After science value has been assigned to the data collected by the lander (using a suitable downlink prioritization scheme), this data will be scheduled for downlinking on the basis of a weighted measure between science value, scientist's preference and data size. In our work, we framed the Downlink data scheduling problem as a Discrete Knapsack problem.

Discrete Knapsack Problem statement: Given a set of items, each with an assigned weight (w_i) and a value (v_i), determine the items to include in a knapsack so that:

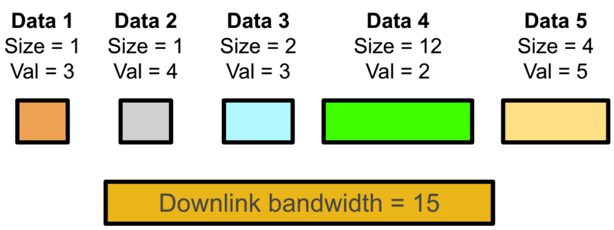
- Total weight of the items $\leq$ given weight limit of the knapsack (W).
- The total value of the included items is as large as possible.

Downlink scheduling problem statement framed as discrete Knapsack Problem: Given a set of collected data packets, with individual packet size (s_i) and a computed value (v_i), determine the data packets to include in the downlink data transmission queue so that:

- Total downlink data size $\leq$ given downlink bandwidth (BW).
- The total downlink value is as large as possible.



(a) The discrete knapsack problem



(b) The downlink data scheduling problem

Figure 2: Analogy between the discrete knapsack problem and the downlink scheduling problem

A brute-force solution to the knapsack problem can get computationally very expensive due to the high data volume that gets collected onboard. However, the discrete knapsack problem demonstrates an optimal substructure and this lets us utilize dynamic programming to find an optimal solution to the downlink data scheduling problem when framed as a discrete knapsack problem [42]. The arguments behind proving this optimal substructure have been reproduced here for reference. Also, it is to be noted that some of the variable names have been changed here as compared to the original work as presented in [42]. This is to make it easy to comprehend in terms of the downlink scheduling problem.

Claim: Let i be a data packet in an optimal solution S for total allowable downlink size = W and total data packets 1...n. Then $S' = S - i$ is an optimal solution for total allowable downlink size = $W - w_i$ and total data packets 1...$n-1$.

Here optimal solution S refers to a sequence of data packets that maximizes the downlink value.

Proof: By contradiction, suppose there was a better solution (solution with a higher downlink value) than S' for the downlink sub-problem of total downlink size = $W - w_i$ and data packets 1....$n-1$. This will mean that S' could be replaced with this better solution, yielding a valid solution for W pounds and data packets $1...n$ with larger value than the solution being considered. This contradicts our initial assumption about S being the optimal solution for this problem. Thus S' has to be an optimal solution for the downlink sub-problem with total allowable downlink size = $W - w_i$ and total data packets $1...n-1$. ■

Recursively defining the optimal solution to the downlink problem:

Let $C[i, w]$ be the maximum value that can be downlinked for a downlink scheduling problem with data packets $1...i$ and maximum allowable size = w. Then the optimal solution for this problem either will include data packet i, in which case the total value will be v_i plus a subproblem solution for data packets $1...i-1$ and the size excluding w_i, OR it will not include data packet i, in which case it is a subproblem solution for data packets $1...i-1$ and size w. The better of these two choices should be made. This all comes down to the following recursion formulation:

$$C[i,w] = \begin{cases} 0, & \text{if } i = 0 \text{ or } w = 0 \\ C[i-1,w], & \text{if } w_i \geq w \\ \max(v_i + C[i-1, w-w_i], C[i-1,w]), & \text{if } \\ \quad i > 0 \text{ and } w \geq w_i & \end{cases}$$

Computing the value of the optimal solution bottom-up:

The algorithm takes in as inputs the maximum allowable size W, the number of data packets n, and the two sequences of value and size $v = (v_1, v_2, ..., v_n)$ and $w = (w_1, w_2, ..., w_n)$ respectively. The downlink value $C[i, j]$ are stored in a table $C[0...n, 0...W]$ and are computed in row-major order, i.e. the first row is filled from left to right, then the second rown, and so on. At the end of the computation, $C[n, W]$ contains the maximum value that can be downlinked for the downlink scheduling problem with n data packets and maximum allowable downlink size of W.

Thus the pseudo code for implementing the dynamic programming to find the optimal solution to the downlink problem formulated as knapsack problem is as follows [42]:

Dynamic_downlink_scheduling(v, w, n, W)

---

for $w \leftarrow 0$ to W **do**
    $C[0, w] \leftarrow 0$;
for $i \leftarrow 1$ to n **do**
    $C[i, 0] \leftarrow 0$;
    for $w \leftarrow 1$ to W **do**
        if $w_i \leq w$ **then**
            if $v_i + C[i-1, w-w_i] > C[i-1, w]$ **then**
                $C[i, w] \leftarrow v_i + C[i-1, w-w_i]$;
            else$C[i, w] \leftarrow C[i-1, w]$;
        else$C[i, w] \leftarrow C[i-1, w]$;
return C

Constructing the optimal downlink data sequence from the computed information:

Let C be the matrix computed by the dynamic downlink scheduling routine, w is the size sequence, n is the number of data packets and W is the maximum allowed downlink size. The optimal sequence of data packets for downlink purposes is then computed as follows:

```
Downlink_data_sequence(C, w, n, W)
Downlink_seq ← φ;
while n > 0 and W > 0 do
    if C[n, W] > C[n − 1, W] then
        Downlink_seq = [Downlink_seq; n];
        W ← W − w[n];
    n ← n − 1;
return Downlink_seq
```

The final sequence of data computed by the downlink planning routine gives an optimal sequence of data packets that maximizes the downlinked science value and minimizes the wastage of the downlink bandwidth.

3. RECOURSE

In the presence of low-bandwidth and high-latency communication constraints, there are limited opportunities for end-users (scientists, i.e. exploration domain experts who are not robotics/autonomy experts) and robot autonomy (REASON module) to interact. Additionally, the user interfaces (UIs) developed in previous space exploration missions [6], [35] are too complex for mission scientists to operate and demand too much mental workload. Taking these issues into consideration, the concept of a novel ground-based system tool, RECOURSE (Ranked Evaluation of Contingent Opportunities for Uninterrupted Remote Science Exploration), has been developed [58]. The primary aims of RECOURSE are to design a schedule (i.e. activity segments processed in the REASON module) that is expected to maximize the science return with intuitive operations and to bridge a gap of situational awareness of unknown environments between human and autonomy. Below is an overview of the main two components of RECOURSE, the Uplink UI and the Downlink UI, and an explanation of how these are used in practice.

Uplink UI

The Uplink UI is a ground-side tool for transmitting signals from Earth to remote robotic explorers. As mentioned in [35], since these UIs developed in previous space exploration missions are too complicated to manipulate, there is a need to develop user-friendly ones for non-robotic experts, and in order to perform more science-driven operation, the desired UI may need to be able to allow scientists to specify search targets in a simple and intuitive way. And, as explained in Sec.2, it may be desirable for activity segments that are transmitted to robotic explorers to guarantee the completion of tasks. Furthermore, since the survey targets are the surfaces of ocean worlds and icy moons such as Europa or Enceladus, each communication takes a long time and the mission lifetime is short, thus it is not possible to modify the schedule while checking the status of the robot autonomy and surrounding environments sequentially as in the case of near-Earth exploration. Hence, it is desirable to have an (even rough) idea of how the robot's state will transition after executing an activity segment. This will help scientists avoid generating schedules that clearly fail tasks. Based on these motivations and the conops shown in Fig.3, the Uplink UI in RECOURSE has been designed and implemented. In the following, how each element of this UI is going to be used by scientists is explained.

Registration of new atomic propositions (APs)—This UI is designed to allow mission scientists to specify high-level science targets intuitively and easily, using not only point-and-click but also semantic sketch (Fig.4) for data (e.g. images) downlinked from the lander. If a sketch is drawn, a convex polytope is constructed from the points that compose it, and the pixel coordinate of the center of the polytope is transformed to the lander-based 3D coordinate based on the lander's pose and the image depth information, which are then associated with a new atomic proposition (AP). When a new AP is registered, not only the coordinate of the AP, but also the date and time of registration, the registrant's name and affiliation, etc. are stored in a database so that users can access who is paying attention to what target and when. Note: if a science target that mission scientists want the lander to investigate is already registered as an AP, this process can be skipped.

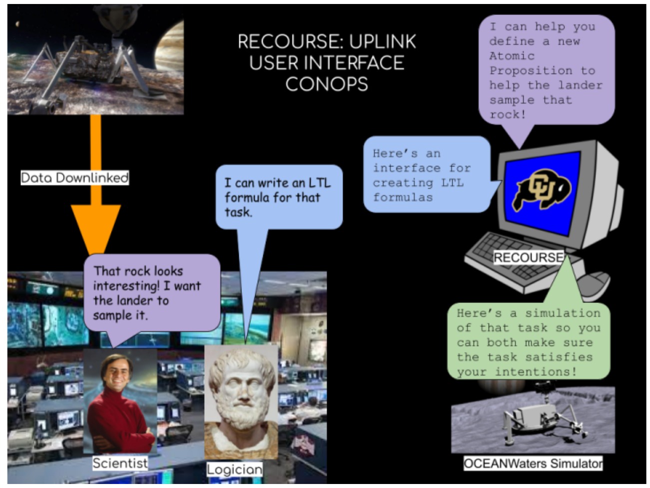


Figure 3: The conops of the Uplink UI.

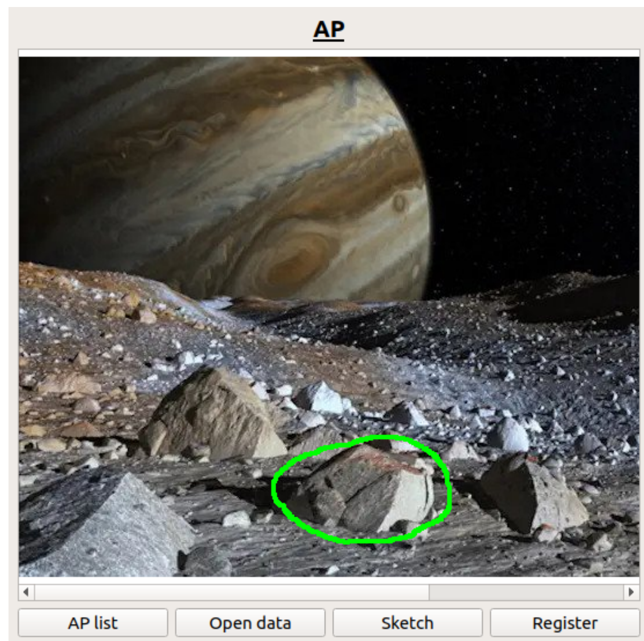


Figure 4: Example of the semantic sketch interface.

Generation of activity segments—In this procedure, a group of scientists (e.g. geologists) determine which activities (e.g. move the ice deposits in AP2 to AP3 and take a picture

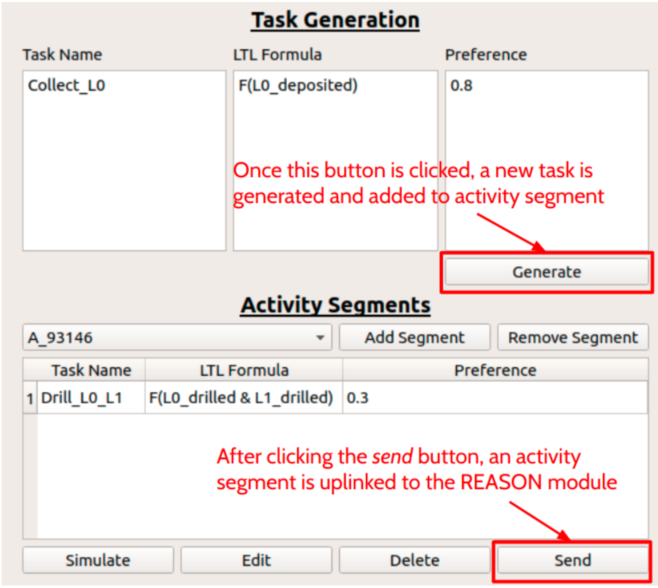


Figure 5: Example of task generation: In this example, an activity segment contains two tasks, 1. Collect_L0 and 2. Drill_L0_L1 (note: it is assumed that what each LTL formula means is encoded in the REASON module in advance).

of AP3) to uplink to the REASON module based on the registered APs, along with their preferences[2]. As described in Section 2, these activities are described by logicians using formal language (i.e. linear temporal logic, *LTL* [36]), based on the intentions of the scientists, to achieve completion while ensuring constraints on missions and instruments[3]. The preferences tied to each activity takes a value between 0 (less critical) and 1 (more critical), and is assumed to have been determined by discussion within the science group. Once all activities and preferences have been finalized, using the OceanWATERS to see if the activities that the scientists desire to have performed would not cause any serious problems, and when the *send* button in Fig.5 is clicked, the information is uplinked to the REASON module and stored in a database with the data and time of transmission, and is accessible in the Downlink UI.

Downlink UI

Ideally, tasks composing an activity segment transmitted from the Uplink UI are executed by the REASON module according to values of science preferences. However, in the REASON module, actual execution plans are scheduled considering not only task preferences but also execution costs, etc. Therefore, tasks may not always be executed as desired by the scientists, or several tasks may be skipped due to unforeseen circumstances (e.g. sudden appearance of plume). Thus, the Downlink UI, another ground-side tool that is a counterpart to the Uplink UI and its conops is shown in Fig.6, allows explicit comparison of the orders of planned and executed tasks in order to improve the explainability of the

[2]In such a space science mission, it is conceivable that several scientist groups such as biologists, chemists, and geologists may propose activities that they want a remote explorer to perform, however in order to put aside issues related to the coordination of preference values between groups for brevity, we assume that activity segments in each uplink cycle are determined by a single group.

[3]Ideally, activities described by scientists in natural language would be automatically converted into formal language, however this is not the focus of our research.

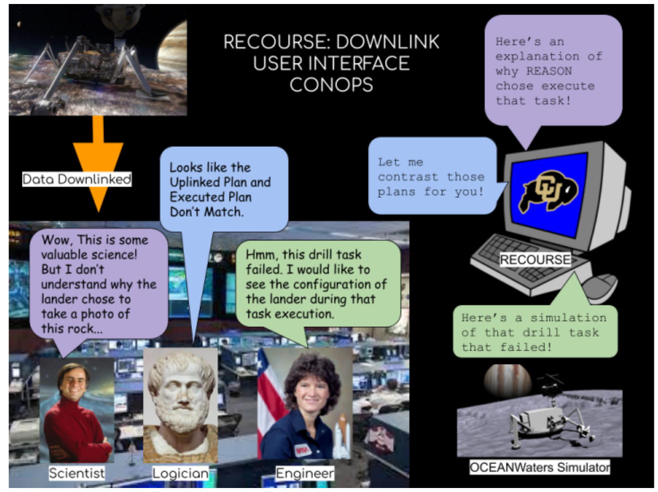


Figure 6: The conops of the Downlink UI.

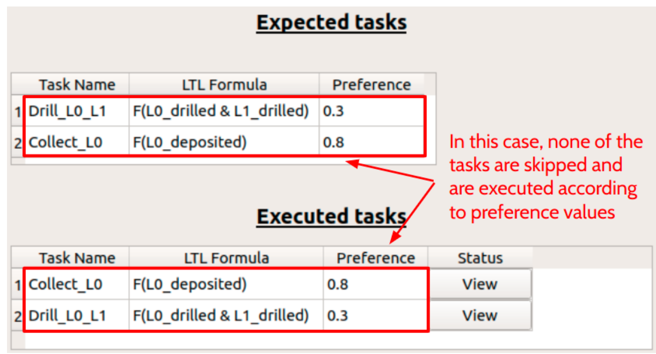


Figure 7: Order comparison between expected and executed tasks. Logicians can immediately understand which tasks are executed in which order and whether they are skipped.

on-board module behavior as shown in Fig.7.[4] The UI also allows the reproduction of executed tasks on OceanWATERS, although not necessarily a perfect replica of the environments, in order to visually capture the behaviors of the robotic lander in uncertain remote environments.

Another major objective of the Downlink UI is to display the data collected by the lander. These data are largely classified into internal data (e.g. instrument temperature, current, and voltage) and external data (e.g. camera and spectral data). It is important to recall that the amount of data that is collected on-board by the lander is significantly higher than what can be transmitted back to Earth – due to high latency and low bandwidth constraints. At the beginning of a downlink transmission window, the downlink planner (described in Sec.2) determines which data can be downlinked according to the data scientific value, data volume and the downlink bandwidth. At present, the downlink data obtained in an entire activity segment is visualized according to the data categories such as battery, camera, and telemetry data. In the particular example as shown in Fig.8, the lander collects all three types of data for the given activity segment. However, the downlinked data comprises only the battery data. This is because the size of the telemetry data collected during

[4]This visual is useful to a trained logician but may not provide detailed insight to the scientists of the motivation behind why the REASON module chose to perform one task over another. So, the exact manner of achieving interpretability is left for future research building upon previous works exploring formal plan explanation [37] [38].

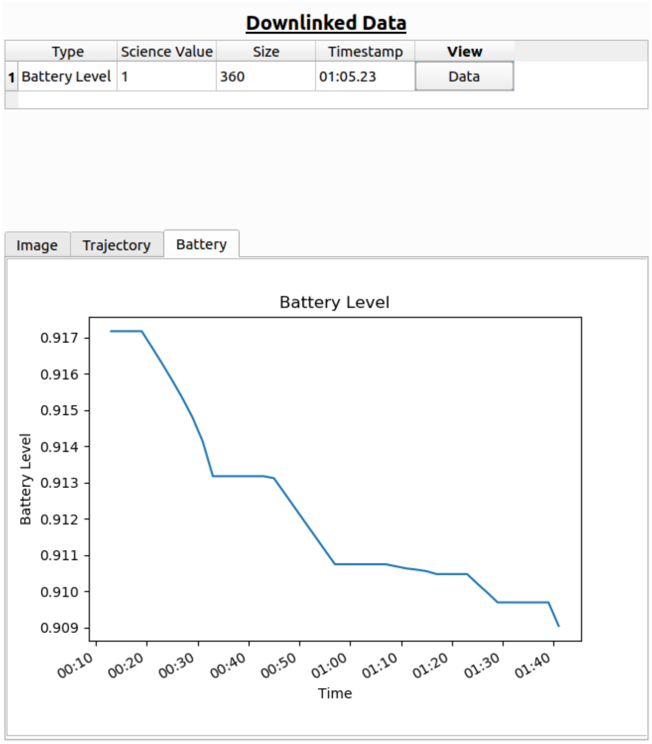


Figure 8: Example of the downlinked battery level data.

the given activity segment was very large to be transmitted at once given the limited transmission bandwidth in this example. Future versions of the downlink UI will enable an user to view data specific to tasks performed by the lander instead of viewing data from an entire activity segment.

4. OceanWATERS Example

OceanWATERS (Ocean Worlds Autonomy Testbed for Exploration Research & Simulation) is a simulation test-bed for a lander concept on an icy moon [41]. The simulation environment uses ROS Noetic with Gazebo, and RViz [39] [40] simulation environments. A visualization of the Gazebo environment can be seen in Fig.9.

Instrument Tools

The simulation test-bed comes with many built in tool functionality for simulating different robotic actions. The robotic components of the lander include a robotic manipulator and a revolving camera/communication module. Each of these tools can be controlled by simulating trajectory information. A drill and a digger tool are attached to the end-effector of the robotic manipulator. The test-bed has built in robotic actions (motion primitives) that can be used to move the drill and the digger.

Case-study

The preference planning framework was implemented in the OceanWATERS simulation test-bed, using the robotic arm. The purpose of this case study is to demonstrate the utility of various realistic planning scenarios for different flexibility thresholds. The experiment utilizes various robotic motion primitives provided by the simulation test-bed such as: "drill", "dig", and "deposit" as seen in Figure 10a, 10b,

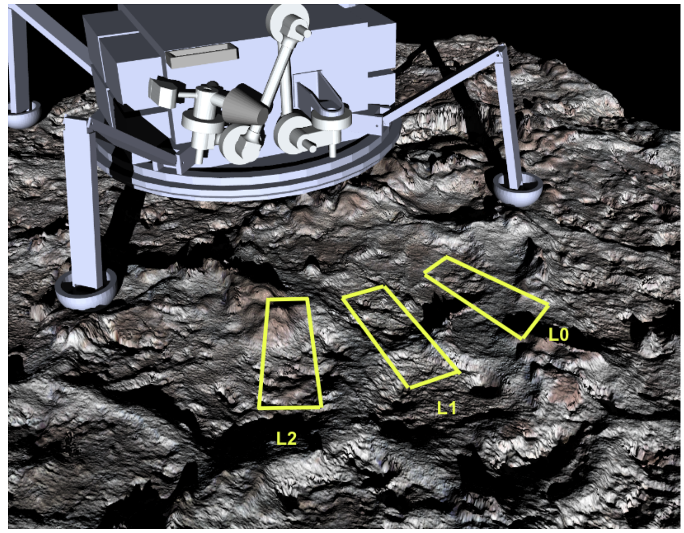


Figure 9: Simulation test-bed for OceanWATERS (Gazebo) including locations of interest $L0$, $L1$, and $L2$

and 10c, respectively.

There are three ground locations labeled $L0$, $L1$, and $L2$, as seen in Figure 9 representing areas containing possible soil of interest. The planner was given three LTL task specification formulae utilizing the *eventually* (F) temporal operator:

$$\phi_1 = F(L0_{deposited}) \quad (2a)$$
$$\phi_2 = F(L1_{deposited}) \quad (2b)$$
$$\phi_3 = F(L2_{dig} \wedge F(L1_{dig} \wedge F(L0_{dig}))) \quad (2c)$$

The formulas are interpreted as, "Eventually deposit soil from location $L0$" for ϕ_1, "Eventually deposit soil from location $L1$" for ϕ_2, and "Eventually dig location $L2$, then location $L1$, then location $L0$ in that order" for ϕ_3.

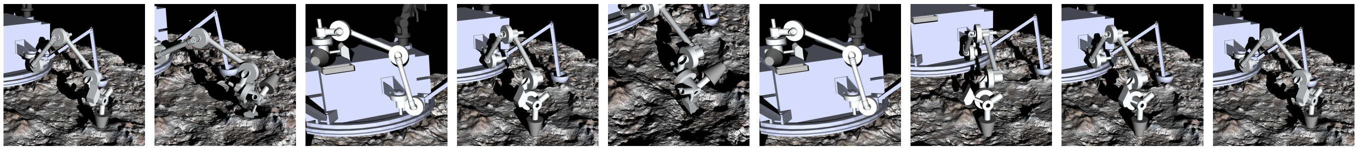
The preferred order of completion is $\phi_1 \rightarrow \phi_2 \rightarrow \phi_3$. When the preference planner is queried with *no flexibility*, the resulting action sequence is shown in Figure 10. While this sequence of actions strictly satisfies each task in the desired order, it requires more number of actions, making the execution of this plan more resource-consuming.

When the preference planner is queried with *full flexibility*, the resulting action sequence is shown in Figure 11. This plan satisfies the tasks out-of-order, i.e., $\phi_3 \rightarrow \phi_2 \rightarrow \phi_1$. By working towards satisfying ϕ_3 first, the planner makes simultaneous progress towards ϕ_1 and ϕ_2. This behavior results in fewer actions and, thus, lower resource-consumption.

5. Conclusion

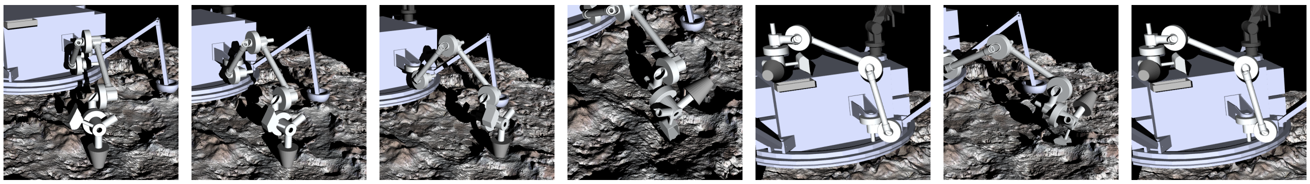
In this paper, we presented the current progress and implementation of our proposed autonomy architecture, which is designed to support deep space exploration missions in the near future.

This architecture consists of two unique components, REASON – the on-board robot autonomy executing LTL-based activity segments – and RECOURSE – the user-friendly interfaces for accelerating science-driven operation – which make



(a) Drill at $L0$ (b) Dig at $L0$ (c) Deposit (d) Drill at $L1$ (e) Dig at $L1$ (f) Deposit (g) Drill at $L2$ (h) Drill at $L1$ (i) Drill at $L0$

Figure 10: Plan for tasks specified in 2(a) - (c). With *no flexibility* the robot accomplishes the task in the order it was specified.



(a) Drill at $L2$ (b) Drill at $L1$ (c) Drill at $L0$ (d) Dig at $L1$ (e) Deposit (f) Dig at $L0$ (g) Deposit

Figure 11: Plan for tasks specified in 2(a) - (c). With *full flexibility*, the robot accomplishes the task out-of-order with fewer steps. The lander first digs at Location $L2$, $L1$, and $L0$ (a) - (c) from ϕ_3 and then digs and deposits the samples (d)-(g).

it possible to carry out tasks in a way that maximizes science return even in uncertain and dynamic environments where human interaction is very limited due to low-bandwidth, high-latency, and limited mission lifetimes. As part of the ongoing development, we presented the current prototypes of REASON and RECOURSE. A new preference planner has been developed and implemented within REASON, along with an initial implementation of a downlink planner. In RECOURSE, the uplink and downlink user interfaces have been significantly developed. The combined system has been demonstrated with NASA's OceanWATERS testbed to showcase its robustness with respect to planning with physical constraints of the multiple tools and accommodating for uncertainties in low level motion plans for tools individually or synergistically.

ACKNOWLEDGMENTS

The authors thank NASA for support of this project through the COLDTech Program, grant #80NSSC21K1031.

REFERENCES

[1] Tara A Estlin, Benjamin J Bornstein, Daniel M Gaines, Robert C Anderson, David R Thompson, Michael Burl, Rebecca Castaño, and Michele Judd. Aegis automated science targeting for the MER opportunity rover. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3):50, 2012

[2] R. Francis, T. Estlin, G. Doran, S. Johnstone, D. Gaines, V. Verma, M. Burl, J. Frydenvang,S. Montaño, R. C. Wiens, S. Schaffer, O. Gasnault, L. DeFlores, D. Blaney, and B. Bornstein. Aegis autonomous targeting for chemcam on mars science laboratory: Deployment and results of initial science team use. Science Robotics, 2(7), 2017.

[3] Michael C. Burl, David R. Thompson, Charles deGranville, and Benjamin J. Bornstein., Rockster: Onboard rock segmentation through edge regrouping. Journal of Aerospace Information Systems, 13(8):329–342, 2016.

[4] Li. Jialun, Wu. Zhongchen, Ling. Zongcheng, Cao. Xueqiang, Guo. Kaichen, and Yan Fabao. Autonomous martian rock image classification based on transfer deep learning methods. Earth Science Informatics, 13(3):951–963, Sep 2020.

[5] Hannah R. Kerner, Kiri L. Wagstaff, Brian D. Bue, Danika F. Wellington, Samantha Jacob,Paul Horton, James F. Bell, Chiman Kwan, and Heni Ben Amor. Comparison of novelty detection methods for multi spectral images in rover-based planetary exploration missions. Data Mining and Knowledge Discovery, 34(6):1642–1675, Nov 2020.

[6] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran. Aspen - automating space mission operations using automated planning and scheduling. In International Conference on Space Operations(SpaceOps 2000), Toulouse, France, June 2000

[7] S. Knight, G. Rabideau, S. Chien, B. Engelhardt, and R. Sherwood. Casper: space exploration through continuous planning. IEEE Intelligent Systems, 16(5):70–75, 2001.

[8] T. Estlin, D. Gaines, C. Chouinard, R. Castano, B. Bornstein, M. Judd, I. Nesnas, and R. Anderson. Increased mars rover autonomy using ai planning, scheduling and execution. In Proceedings 2007 IEEE International Conference on Robotics and Automation, pages4911–4918, 2007.

[9] Daniel Gaines, Joseph Russino, Ryan Doran, Garyand Mackey, Michael Paton, BrandonRothrock, Steve Schaffer, Ali-Akbar Agha-Mohammadi, Chet Joswig, Heather Justice, Ksenia Kolcio, Jacek Sawoniewicz, Vincent Wong, Kathryn Yu, Gregg Rabideau, Robert Anderson, and Ashwin Vasavada. Self-reliant rover design for increasing mission productivity. In International Conference on Automated Planning and Scheduling (ICAPS), Pasadena, CA, USA, June 2018. Jet Propulsion Laboratory, National Aeronautics and Space Administration.

[10] Akash Arora, P. Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. Multi-modal active perception for information gathering in science missions.Autonomous Robots, 43(7):1827–1853, Oct 2019.

[11] Gregg Rabideau and Ed Benowitz. Prototyping an onboard scheduler for the mars 2020 rover. In 10th International Workshop on Planning and Scheduling for Space (IWPSS 2017), Pasadena,CA, USA, June 2017. Jet

Propulsion Laboratory, National Aeronautics and Space Administration.

[12] Jared Strader, Kyohei Otsu, and Ali-akbar Agha-mohammadi. Perception-aware autonomous mast motion planning for planetary exploration rovers. Journal of Field Robotics,37(5):812–829, 2020.

[13] Roger C. Wiens, Sylvestre Maurice, Scott H. Robinson et al., The supercam instrument suite on the nasa mars 2020 rover: Body unit and combined system tests.Space Science Reviews, 217(1):4, Dec 2020.

[14] S. Bhaskaran. Autonomous navigation for deep space missions. InSpaceOps 2012, 2012.

[15] Christopher Grasso. VML 3.0 Reactive Sequencing Objects and Matrix Math Operations for Attitude Profiling. InSpaceOps 2012, Reston, Virigina, March 2013. American Institute of Aeronautics and Astronautics.

[16] Eric N Johnson, Anthony J Calise, Michael D Curry, Kenneth D Mease, and J Eric Corban. Adaptive Guidance and Control for Autonomous Hypersonic Vehicles. Journal of Guidance,29(3):725–737, May 2006.

[17] Nikos Mastrodemos, Daniel G Kubitschek, and Stephen P Synnott. Autonomous navigation for the deep impact mission encounter with comet tempel 1. Space Science Reviews,117(1-2):95–121, 2005

[18] J. Riedel, Andrew Vaughan, Robert Werner, Tseng-Chan Wang, Simon Nolet, David Myers,Nickolaos Mastrodemos, Allan Lee, Christopher Grasso, Todd Ely, and David Bayard. Optical Navigation Plan and Strategy for the Lunar Lander Altair; OpNav for Lunar and other Crewedand Robotic Exploration Applications. In AIAA Guidance, Navigation, and Control Conference, Reston, Virigina, 2010. American Institute of Aeronautics and Astronautics

[19] E Riedel, C A Grasso, and W M Owen Jr. A Survey of Technologies Necessary for the NextDecade of Small Body and Planetary Exploration. Technical report, 2009.

[20] Scott D Barthelmy, Louis M Barbier, Jay R Cummings, Ed E Fenimore, Neil Gehrels, DerekHullinger, Hans A Krimm, Craig B Markwardt, David M Palmer, Ann Parsons, Goro Sato, Masaya Suzuki, Tadayuki Takahashi, Makota Tashiro, and Jack Tueller. The Burst Alert Telescope (BAT) on the SWIFT Midex Mission. Space Sci Rev, 120(3-4):143–164, October2005.

[21] A K Jónsson, P H Morris, N Muscettola, and K Rajan. Planning in Interplanetary Space:Theory and Practice. AIPS, 2000

[22] F Teston, R Creasey, J Bermyn, D Bemaerts, and K Mellab. PROBA: ESA's autonomy andtechnology demonstration mission. 1999.

[23] S Chien, R Sherwood, D Tran, R Castano, and B Cichy. Autonomous science on the EO-1mission. 2003

[24] S Chien, R Sherwood, D Tran, and B Cichy. Lessons learned from autonomous sciencecraft experiment.. . . on Autonomous agents . . ., page 11, 2005.

[25] S Chien, D Tran, A Davies, and M Johnston. Lights out autonomous operation of an earth observing sensor web.. . . and Operations (. . ., 2007.

[26] Steve Chien, Gregg Rabideau, Russell Knight, Robert Sherwood, Barbara Engelhardt, DarrenMutz, Tara Estlin, Benjamin Smith, Forest Fisher, T Barrett, et al. Aspen-automating spacemission operations using automated planning and scheduling. In Space Ops 2000, 2000.

[27] Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy, Gregg Rabideau, Rebecca Castano, Ashley Davis, Dan Mandl, Bruce Trout, Seth Shulman, and Darrell Boyer. Using Autonomy Flight Software to Improve Science Return on Earth Observing One. Journal of Aerospace Computing, Information, and Communication, 2(4):196–216, April 2005

[28] HL Enos and DS Lauretta. A rendezvous with asteroid bennu. Nature Astronomy,3(4):363–363, 2019.

[29] Steve Chien, Gregg Rabideau, Daniel Tran, Joshua Doubleday Thompson, Federico Nespoli, Miguel Perez Ayucar, Marc Costa Sitje, Claire Vallat, Bernhard Geiger, Nico Altobelli, Manuel Fernandez, Fran Vallejo, Rafael Andres, and Michael Kueppers. Activity-based scheduling of science campaigns for the rosetta orbiter. Invited Talk, in Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 2015.

[30] M. Costa, M. Perez-Ayucar, M. Almeida, M. Ashman, R. Hoofs, S. Chien, J. Beteta, and M. Kueppers. Rosetta: Rapid science operations for a dynamic comet. In International Conference On Space Operations (Space Ops 2016), Daejeon, Korea, May 2016.

[31] B. A. Bauer and W. M. Reid. Automating the pluto experience: An examination of the new horizons autonomous operations subsystem. In2007 IEEE Aerospace Conference, pages 1–10, 2007

[32] Brian Bauer, Alice F. Bowman, Omar Custodio, Glen Fountain, Sarah A. Hamilton, Helen Hart,Chris Hersman, Adrian Hill, Valerie Mallder, Nickalaus Pinkine, Gabe Rogers, Rebecca Sepan,Karl Whittenburg, and Stephen Williams. Lessons Learned from the New Horizons July 4th Anomaly.

[33] Yanping Guo and Robert W. Farquhar. Baseline design of new horizons mission to pluto and the kuiper belt. Acta Astronautica, 58(10):550–559, 2006

[34] Robert C. Moore. Autonomous safeing and fault protection for the new horizons mission to pluto. Acta Astronautica, 61(1):398–405, 2007. Bringing Space Closer to People, Selected Proceedings of the 57th IAF Congress, Valencia, Spain, 2-6 October, 2006.

[35] Vandi Verma, Dan Gaines, Gregg Rabideau, Steve Schaffer, and Rajeev Joshi. Autonomous Science Restart for the Planned Europa Mission with Lightweight Planning and Execution, The 27th International Conference on Automated Planning and Scheduling (ICAPS), Pittsburgh, Pennsylvania, June 18-23, 2017.

[36] Christel Baier and Joost-Pieter Katoen. Principles of Model Checking, The MIT Press, Cambridge, MA, 2008.

[37] Bastian Seegebarth, Felix Müller, Bernd Schattenberg, and Susanne Biundo. Making Hybrid Plans More Clear to Human Users - A Formal Approach for Generating Sound Explanations, The 22nd International Conference on Automated Planning and Scheduling (ICAPS), Sao Paulo, Brazil, June 25-29, 2012.

[38] Joseph Kim. Bayesian Inference of Linear Temporal Logic Specifications for Contrastive Explanations, Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, August 10-16, 2019.

[39] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator, IEEE/RSJ International Conference on Intelligent Robots

and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, pp. 2149-2154 vol.3, doi: 10.1109/IROS.2004.1389727.

[40] David Coleman. "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," Journal of Software Engineering for Robotics, vol. ArXiv, 2014.

[41] National Aeronautics and Space Administration. Ocean Worlds Autonomy Testbed for Exploration Research & Simulation (OceanWATERS), GitHub Repository, https://github.com/nasa/ow_simulator.

[42] Cheng Li, et al. "Dynamic Programming Solution to the Discrete Knapsack Problem".

[43] A. Bhatia, M.R. Maly, L. E. Kavraki, and M. Y. Vardi. Motion planning with complex goals. Robotics Automation Magazine, IEEE, 18(3):55–64, Sep. 2011.

[44] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in IEEE international conference on robotics and automation. IEEE, 2015,pp. 346–352.

[45] Giuseppe De Giacomo and Moshe Y Vardi. Linear temporal logic and linear dynamic logic on finite traces. In IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pages 854–860. Association for Computing Machinery, 2013.

[46] Keling He, Morteza Lahijanian, Lydia E. Kavraki, and Moshe Y. Vardi. Reactive synthesis for finite tasks under resource constraints. In Int. Conf. on Intelligent Robots and Systems (IROS), pages 5326–5332, Vancouver, BC, Canada, Sep. 2017. IEEE.

[47] Keliang He, Morteza Lahijanian, Lydia E. Kavraki, and Moshe Vardi. Automated abstraction of manipulation domains for cost-based reactive synthesis. IEEE Robotics and Automation Letters, 4(2):285–292, Apr. 2019.

[48] I. A.S ucan and L. E. Kavraki, "A sampling-based tree planner for systems with complex dynamics," IEEE Transactions on Robotics, vol. 28, no. 1, pp. 116–131, 2012

[49] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in Int. Conf. on Robotics and Automation, vol. 2. IEEE, 2000, pp. 995–1001

[50] H. Kress-Gazit, G. Fainekos, and G. J. Pappas. Where's waldo? sensor-based temporal logic motion planning. In Int. Conf. on Robotics and Automation, pages 3116–3121, Rome, Italy, 2007. IEEE.

[51] M. Lahijanian, J. Wasniewski, S.B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees,. In Int. Conf. on Robotics and Automation, Anchorage, Alaska, 2010. IEEE.

[52] Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. IEEE Transactions on Robotic, 28(2):396–409, Apr. 2012.

[53] Rebecca Castafio et al. "Rover Traverse Science for Increased Mission Science Return". In:(2003)

[54] Tara Estlin et al. "Enabling Autonomous Science for a Mars Rover". In: American Institute of Aeronautics and Astronautics, May 2008

[55] Srija Chakraborty et al. "Expert Guided Rule Based Prioritization of Scientifically Relevant Images for Downlinking over Limited Bandwidth from Planetary Orbiters". In: Proceedings of the AAAI Conference on Artificial Intelligence33.01 (July 2019), pp. 9440–9445.

[56] D. R. Thompson, T. Smith, and D. Wettergreen. "Information-optimal selective data return for autonomous rover traverse science and survey". In:2008 IEEE International Conference on Robotics and Automation. 2008, pp. 968–973

[57] Kupferman, O. and Vardi, M.Y., 2001. Model checking of safety properties. Formal methods in system design, 19(3), pp.291-314.

[58] J. McMahon, N. Ahmed, M. Lahijanian, P. Amorese, T. Deka, K. Muvvala, T. Slack, and S. Wakayama. "Expert-Informed Autonomous Science Planning for In-situ Observations and Discoveries," 2022 IEEE Aerospace Conference (AERO), 2022, pp. 1-11, doi: 10.1109/AERO53065.2022.9843445.

BIOGRAPHY

Jay McMahon *received his B.S. from the University of Michigan, M.S. from USC and a Ph.D. from the University of Colorado Boulder in 2011. All three degrees in Aerospace Engineering. He is currently an Associate Professor in the Smead Aerospace Engineering Sciences department at the University of Colorado Boulder where his research focuses on spacecraft guidance, navigation and control and asteroid missions and associated science.*

Nisar Ahmed *Nisar Ahmed is an Associate Professor and H.J. Smead Faculty Fellow in the Smead Aerospace Engineering Sciences Department at the University of Colorado Boulder, and holds a courtesy appointment in the Computer Science Department. He is a member of the Research and Engineering Center for Unmanned Vehicles (RECUV) and directs the Cooperative Human-Robot Intelligence (COHRINT) Lab. He conducts research in probabilistic modeling, estimation and control of intelligent autonomous systems for problems involving human-robot/machine interaction, distributed sensor and information fusion, and decision-making under uncertainty.*

Morteza Lahijanian *is an Assistant Professor in the Dept. of Aerospace Engineering Sciences and Dept. of Computer Science (by courtesy) at the University of Colorado Boulder, USA and the director of Assured, Robust, and Interactive Autonomous (ARIA) Systems group. He received his B.S. in Bioengineering from University of California, Berkeley, M.S. in Mechanical Engineering from Boston University, and Ph.D. in Mechanical Engineering from Boston University. He conducted his postdoctoral research in the Dept. of Computer Science at Rice University. Prior to joining CU Boulder, he was a research scientist in the Dept. of Computer Science at University of Oxford, UK.*

Peter Amorese *received his B.S. from the University of Colorado Boulder in 2021 and is currently pursuing a M.S. at the University of Colorado Boulder in Aerospace Engineering. His research interests include formal synthesis, high-level autonomous planning with multiple task preferences, and other formal methods topics.*

Taralicin Deka *is a Ph.D. candidate at the Orbital Research Cluster for Celestial Applications Lab (ORCCA lab) at the University of Colorado Boulder. She is originally from Assam, India. She holds a B.Tech degree in Electrical Engineering from the National Institute of Technology Silchar, India, and an MSc. degree in Aerospace Engineering Sciences from the University of Colorado Boulder, USA. Her current research focuses on multi-agent relative-motion planning in perturbed environments.*

Karan Muvvala *is a Ph.D. student in the Aerospace Engineering Sciences department at the University of Colorado Boulder. He received his M.S. in Mechanical Engineering from the University of Colorado Boulder in 2021. His research interest includes Formal Methods, Verification and Planning for Autonomous systems, Game Theory and Control Theory.*

Trevor Slack *received his B.S. from the University of Colorado Boulder in 2021 and is currently pursuing a M.S. at the University of Colorado Boulder. Both degrees in Aerospace Engineering. His research interests includes autonomous risk identification and classification, simulation design, and human-robot interaction.*

Shohei Wakayama *is a Ph.D. student in the Aerospace Engineering Sciences department at the University of Colorado Boulder. He received his B.S. in Mechanical Engineering from Kyushu University in 2018. His research interest includes autonomous search and exploration, decision making under uncertainty, and human-robot interaction.*

Kian Shakerin *is a Ph.D. student in the Orbital Research Cluster for Celestial Applications Lab (ORCCA Lab) within the Department of Aerospace Engineering Sciences at the University of Colorado Boulder. He received his Masters in Astrophysics from the University of St Andrews in 2019. His research interests include spacecraft autonomy, guidance and control, and astrodynamics.*