Y2 Progress Report
Award Number: 80NSSC20K1720
Project Name: RASPBERRY SI: Resource Adaptive Software Purpose-Built for Extraordinary Robotic Research Yields — Science Instruments (USC)
PI: Pooyan Jamshidi (USC)
Program Manager: Carolyn Mercer (NASA)

Project Website: https://nasa-raspberry-si.github.io/raspberry-si/
Team Members: Pooyan Jamshidi (USC, PI), Bradley Schmerl (CMU, Co-I), David Garlan (CMU, Co-I), Javier Camara (York, Collaborator), Jianhai Su (USC, Graduate Student), Abir Hossen (USC, Graduate Student), Ellen Czaplinski (NASA, Science Consultant), Katherine Dzurilla (UArk, Science Consultant), Sonam Kharde (USC, Postdoc)

Y2 Progress:

In Y2, our primary focus has been on preparing an end-to-end solution that enabled us to evaluate our approach with both testbeds. Specifically, the **end-to-end solution** allowed us to integrate our Autonomy module with both virtual (OceanWATERS) and physical testbeds (OWLAT) and enabled us to run the integrated solution (Autonomy + Systems Under Test) in several mission scenarios, including excavation.

The RASPBERRY-SI team (Pooyan Jamshidi, Sonam Kharde, and Abir Hossen) **visited NASA JPL** (host: Hari Nayar) from August 1st – August 20th. We performed several tests and evaluated our solution on the JPL physical testbed. We have also performed several demos for the program managers at JPL and delivered presentations to discuss the technical details of the RASPBERRY-SI solution. During our visit, the virtual testbed provider team at NASA Ames (Mike Dalal) visited JPL, and we also demoed our solution with the virtual testbed successfully. All code, data, demos, and results can be found in our GitHub organization at https://github.com/nasa-raspberry-si

Our Autonomy implements a cohesive and orchestrated framework for enabling **Monitoring** (of the system states), Analysis (of structural and behavioral constraints to trigger re-planning), **Planning** (synthesizing a policy for generating actions), and **Execution** (that enacts the synthesized plan to the system under test).

We extended our **probabilistic planner** for the RASPBERRY-SI project. This extension entailed analyzing the domain in cooperation with other team members and writing prototype specifications for the PRISM probabilistic model checker, which is used in the backend of the planner. These specifications capture an abstract version of an excavation scenario as a Markov decision process for which PRISM can synthesize a policy that selects, e.g., which locations should be excavated. The planner builds a PRISM model based on the given list of excavation and dump locations, then exports a policy by querying it against a given property that encodes the multi-objective optimization problem and extracts a high-level plan from the policy.

In addition, we have addressed one of the outstanding challenges of probabilistic model checking to enable **plan synthesis at runtime**. Due to ample state space, the model verification process can take a long time to return the verification result. On the other hand, we need to produce an adaptation plan in near real-time to deal with runtime uncertainties. To address this challenge, we rely on **causal reasoning**. Specifically, we learn a causal model at design time and update the model at runtime. The causal model allows us to restrict the search space of the probabilistic planner and, therefore, the probabilistic planner can synthesize adaptation plans on the fly.

We have implemented a **unified interface between the autonomy and both testbeds (virtual and physical)**. Specifically, the interface includes (i) a message loop that is currently implemented using ROS topics, adding a planner ROS node to communicate with ow_autonomy (the executor of the PLEXIL plan), and enabling the support of the PLEXIL Update node in ow_autonomy node to retrieve the execution status of a PLEXIL plan. We have also implemented (ii) a translator between the planner and executor as part of this interface. The translator translates the policy that has been synthesized by the planner into a PLEXIL plan. The interface also includes (iii) a layer that translates testbed-specific operations into an Intermediate Representation and allows the autonomy module to work seamlessly with both testbeds. This interface could be potentially helpful for other teams involved in the ARROW program.

We implemented a time-based **fault injection** and removal for testing against the current Autonomy. A fault_tester ROS node is implemented to monitor the execution time of the current plan such that a specified arm fault will be injected when the corresponding time condition is triggered. The actual injection is made by utilizing the existing fault injection facility inside the OceanWATERS testbed. Since the OceanWATERS testbed can detect an arm fault and publish the fault status in a ROS topic, we enhanced our current Autonomy to periodically listen to the ROS topic and synthesize a new plan once the existence of a fault is notified via the ROS topic.

We made several contributions to decide what trench location to select to maximize the science value and minimize operational costs. First, we needed to determine the brine from the icy surface and categorize the brine surface based on the scientific values. We, therefore, **curated a small dataset** for initial evaluation and integration of our machine learning models with our planner in the excavation scenario (we used this scenario for the development and extension of our infrastructure). We then built a solution for category labeling, instance spotting, and instance segmentation. Specifically, we developed an **ML model that predicted Brine classes** (e.g., High Brine, Medium Brine, Low Brine). The solution estimates the science values per location and provides these predicted values to the Autonomy to select the sampling locations.

The science consultants of our project made several presentations explaining the **overall scientific value of sending a mission to an ocean world**. We summarized relevant science and engineering articles about project topics, such as the Mars Phoenix lander arm instruments and operations. Finally, we presented the main points from the Europa Lander Study 2016 Report (Hand et al., 2016) and how it relates to our project of a lander performing autonomous operations on Europa.

We designed an approach for **characterizing the "known unknowns" and the "unknown unknowns"** using our advancements in Causal Modeling and Inference. These situations demonstrate the capability of our Autonomy to deal with unexpected and unanticipated situations. In addition to characterizing these situations, we are developing a method to identify them on the fly and reason about their impacts on the mission objectives using the model we developed to characterize unexpected situations. The model-based reasoning allows Autonomy to devise a reaction to avoid a possible fault or identify the root causes and repair the system if the fault has already occurred. This method can be potentially used for offline testing of autonomous systems.

We have updated the project **website** with all presentations and research outcomes. The website serves as a medium for **disseminating** the results of our project: https://nasa-raspberry-si.github.io/raspberry-si/

We presented the results at several NASA-related conferences, including the **Planetary Science Informatics and Data Analytics Conference (PSIDA 2022)** and **53rd Lunar and Planetary Science Conference (LPSC 2022)**. We also have done an earlier presentation at Claflin University (an HBCU in SC) to promote computer science and to hire summer interns from **underrepresented and under-served** students in the STEM enterprise.

We have also done a couple of **media outreach**:

- **University of South Carolina:**
https://www.sc.edu/study/colleges_schools/engineering_and_computing/news_events/news/2020/jamshidi_ai_space.php
- **The Post and Courier newsletter:**
https://www.postandcourier.com/columbia/news/usc-researcher-wants-to-train-robots-for-nasa-deep-space-missions/article_93d9bb3c-3afa-11eb-bd4c-7700ac496485.html