A Simulation Framework for Computation Sharing in Mars Spacecraft Network

Tiago Vaquero and **Joshua Vander Hook** and **Martina Troesch** and **Steve Chien**
Jet Propulsion Laboratory
Caltech Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Introduction

As Mars becomes more crowded, with potentially seven NASA, commercial, or international missions in 2020 alone, the possibility for inter-asset collaboration becomes more tantalizing and attractive. Furthermore, Mars exploration has already established the value and feasibility of multiple simultaneously operating assets. Given that operation of ground or orbital assets increasingly relies on computationally-expensive algorithms and complex data products, such promising multi-asset, collaborative environments provide a favorable scenario for studying opportunities for autonomously sharing data and processing load.

The autonomous distribution of processing load across multiple orbital or ground assets is a game-changing paradigm for future Mars missions, providing a way to "divide" labor and request (or provide) computation or data between assets without requiring centralized servers. This computation sharing/network concept, dubbed MOSAIC (Mars On-site Shared Analytics, Information, and Computing), is part of a multi-year research initiative at JPL that seeks to develop autonomy capability to address challenges related to optimal allocation of networked computation and data storage across lossy, delay-tolerant networks for future space missions. MOSAIC networks will become increasingly important for missions that rely on networked devices to share data or computation, such as multi-rover coordination in Mars cave exploration, distributed science in wind profiling, polar ice cap transverse sampling, or global seismic profiling. More details will be available in (Vander Hook and et. al. 2018)

In this demonstration, we present a simulation framework that has been used to study the impact of the MOSAIC networks and the use of automated scheduling approaches to allocate computation load across different assets on Mars-oriented missions. The simulator provides a range of parameters for missions and integrates an automated scheduler, path planning for surface rover navigation, terrain and communication models, and a set of visualization tools.

Mars Rover Scenario

In this demonstration, we show as an example a Mars 2020 like navigation scenario in which the rover has to traverse a terrain/area as part of a walk-about. The navigation is a cyclic process that involves three major steps: sense its surroundings, plan its path, and act every 30 seconds. If a plan is not available in that cycle (because the computation is taking too long) the rover stops until a plan is available. Some of the tasks in that cycle can be distributed/shared with an orbiter or balloon to allow the rover to use the spare time to do extra science tasks or terrain analysis to identify unseen terrain types. Given the distance and communication bandwidth between the rover and the other assets, one can use a scheduler to compute the optimal regime for each 30-second cycle to determine what task can be shared and which extra task can be added to support the rover.

Simulation Framework

Herein, a mission is modeled as a set of vehicles (agents), their capabilities (tasks they can execute and share), their goals and process cycles (expressed as a task/software network), a communication network involving those vehicles (bandwidths, comm links, time windows), and the environment (surface terrain, obstacles).

We implement the above models using the Robotic Operating System (ROS), a common platform used in robotics. Figure 1 shows a simulated scenario with two Mars rovers and a balloon. Following we provide an overview of the different components and systems involved in this framework.

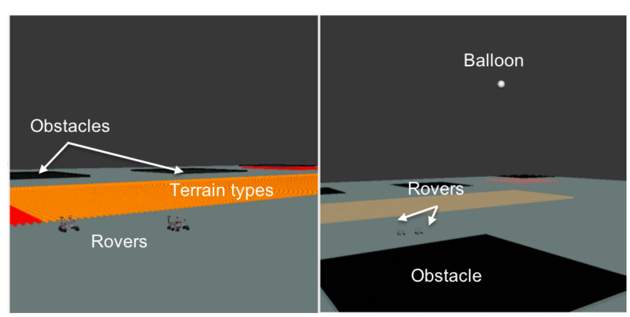

Figure 1: Simulation framework for computation sharing in a Mars exploration scenario.

Vehicles Each vehicle is modelled as an agent that has a *mobility component*, a *communication component*, and a *science component*. In the rover, the mobility component is responsible for planning the path of the vehicle through the terrain, which includes accounting for obstacles and different terrain types. Our framework allows the integration of a path planner (e.g., grid, graph or sample based planners) to determine the set of way point to target locations. The navigation component is not only defined by the path planning approach but also through the vehicle's velocities in different terrain types, power consumption during the navigation task, and the obstacle detection range.

The communication component plays a particularly important role in our simulator. It allows for vehicles to send data back and forth across the networked set of assets. Through an antenna model, our simulator models communication ranges and bandwidths for each individual vehicle, as well as data transfer processes and durations. At any given time in the simulation a time-varying contact graph can be constructed to capture the communication network topology between agents. For each agent, the graph provides a list of all the time intervals during which it can establish a directed communication link, as well as the respective bandwidth given the distances between the agents.

A science component can also be available for a vehicle. One can specify a list of science instruments a vehicle has and the sensing they can perform with them. Each instrument is specified by its type, power usage and data volume per reading, sensing duration, as well as range.

In addition to the three aforementioned components, a vehicle has processing power, memory to store data products (e.g. generated from science instruments) and a hotel load. Moreover, the framework allows the specification of task capabilities for each vehicle, including algorithmic tasks (e.g., path planning, terrain analysis) and physical tasks (e.g., drive, transmit data) as well as their computation requirements and durations (deterministic or stochastic). A task network is provided for each vehicle to model its behavior (e.g., the 30-second sense-plan-act cycles), goals (optional tasks that increase utility - tasks that can be shared are explicitly identified in the model), and the data products required for each task (e.g., a path planning task might require a map).

All the aforementioned component specifications are parameterizable and serve as the input to our simulation environment. That helps to study a large spectrum of scenarios (stochastic or deterministic) to evaluate the scheduling performance and the MOSAIC network approach.

Autonomy Each vehicle has a controller that determines the actions to be performed at a given time. The controller uses an automated scheduler (Vander Hook and et. al. 2018) to determine the computation sharing regimes at each given time (or cycle). If a task is assigned to another vehicle, the controller would dispatch, if necessary, transfer actions to send data products to the target asset so that it can perform the task (e..g, sending a picture of the terrain for the ballon to perform terrain analysis).

Environment The environment is modelled as a terrain - a map of a surface on which rovers can traverse. To model the terrain in our simulations, we use terrain data classified from HiRISE imagery from (Ono et al. 2016). Multiple terrain types are grouped into different classes or as obstacles (terrain that cannot be traversed). We do not currently take slope into account, therefore we model the velocity of a rover in a given terrain class based on the average speed over multiple slopes for that classification. Figure 2 illustrates the different terrain types and vehicle paths using different regimes.

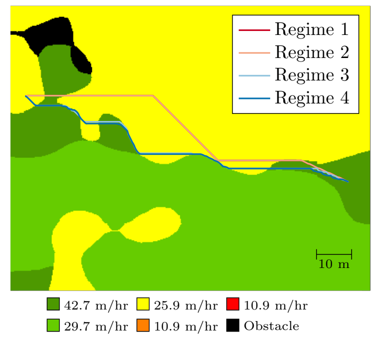

Figure 2: Example paths of rovers operating with four different regimes across different terrain types.

In order to model the different fidelity of data obtained in orbit and on the ground by the rover, we assume certain terrain types as unknown. A rover moves at the velocity of the real terrain class, but plans a path assuming a terrain with the fastest traverse velocity. However, if a rover is able to perform terrain classification, we assume it will be able to correctly classify the terrain within a given radius.

In our demonstration, we use an optional terrain analysis task to be scheduled by the controller depending on the regime to show the benefits of sharing computing resources. If a rover manages, for example, to share the planning task, the rover is able use the extra time to detect unseen terrain types and optimize its path during its walk-about.

We will show a simulation for the aforementioned scenario using different rovers with different sharing strategies, ranging from doing all processes onboard to optimizing the computing shared resources in order to illustrate the impact of the MOSAIC network.

Acknowledgments

Portions of this work were performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

Ono, M.; Rothrock, B.; Almeida, E.; Ansar, A.; Otero, R.; Huertas, A.; and Heverly, M. 2016. Data-driven surface traversability analysis for mars 2020 landing site selection. In *Aerospace Conference, 2016 IEEE*, 1–12. IEEE.

Vander Hook, J., and et. al. 2018. Dynamic shared computing resources for multi-robot mars exploration. In *paper submitted to ICAPS'18 Workshop on Planning and Robotics (PlanRob)*.