Onboard re-planning for robust mapping using pre-compiled backup observations

Martina Troesch and **Faiz Mirza** and **Gregg Rabideau** and **Steve Chien**
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

Mapping target bodies by imaging as much of the surface as possible is a common scientific goal for space missions where a spacecraft is orbitting a body, such as a comet, asteroid, or planet. An observation schedule to achieve the mapping goal is generally generated on the ground and then uploaded to the spacecraft. However, without some re-planning capability onboard, opportunities may be lost due to observation failures or unexpected changes in resource availability. The computational and memory restrictions for spacecraft make it difficult to perform the geometric reasoning and calculations required to select observations to achieve the mapping goal onboard, meaning that any re-planning capabilities are also limited. In this paper we present a method for robust mapping by re-planning observations onboard using pre-compiled backup observations. The nominal schedule and backup observations are generated using the Compressed Large-scale Activity Scheduler and Planner, which are then translated into a Task Network and goal definitions. These can be used by MEXEC, an onboard planning and execution software. We demonstrate our method using a hypothetical scenario of a spacecraft orbiting a comet.

Introduction

Designing operations to achieve scientific objectives for space missions requires a careful balance between meeting observation requirements and satisfying onboard resource constraints. Automated scheduling tools that model spacecraft state against scientific observation opportunities can be used to generate schedules that optimize scientific objectives while respecting spacecraft trajectory and resource constraints. For orbiting missions, one common objective is to map a target body by covering as much of the surface as possible with imaging observations. The geometric reasoning needed to calculate the coverage achieved by the mapping observations and to model resource usage against the spacecraft state can require a significant amount of computational power and memory. Often, this type of scheduling cannot be done on embedded hardware with limited computational resources.

A limiting factor for how much coverage can be achieved is the data volume available onboard for scientific data. If the state of the data volume is not accurate when the schedule is produced, science opportunities could be missed or dropped. Furthermore, if an observation fails, re-planning an attempt to map the missed observation using geometric reasoning is unlikely due to the aforementioned computational and memory limitations available for planning onboard.

To overcome these challenges, we propose an approach for robust mapping by using pre-compiled backup observations to re-schedule onboard. A baseline schedule, as well as backup observations to be used onboard are generated on the ground. To generate the baseline schedule and backup observations, we use an adaptation of the Compressed Large-scale Activity Scheduler and Planner (CLASP) (Knight and Chien 2006; Rabideau et al. 2010). CLASP uses a variety of models to define a scenario, including: geometric models of the spacecraft and its instruments, such as the field of view and position of the instrument on the spacecraft; resource constraints, such as data volume and power available onboard and usage by instruments; and trajectory and orientation of the spacecraft and target body. It also takes definitions of science goals, which includes a priority as well as instrument and geometric constraints.

To represent the onboard planner and controller, we use MEXEC (Verma et al. 2017). MEXEC uses spacecraft state updates and projections to schedule and execute tasks. Tasks are uploaded to MEXEC and represent an executable unit in a plan that also contain information about constraints and model behavior. By maintaining the contraints and effects onboard in the tasks, both the controller and planner can react to unexpected onboard events.

To demonstrate this approach, we use a model based on the DAWN (Russell and Raymond 2011) spacecraft with one of its framing cameras orbiting a miniature version of the comet Finlay as a fictional mapping scenario.

This paper is organized as follows. We begin by setting up a motivating scenario, followed by a description of the CLASP algorithm and a summary of MEXEC. We then explain our approach for robust mapping and show experimental results, followed by related work, future work, and a conclusion.

Motivating Scenario

We have constructed an example scenerio where the scientific objective is for an orbiting spacecraft to map as much of a comet target body as possible. The images must be taken with a solar incidence angle between 0 and 90 degrees (i.e., during the day). For the spacecraft, we use a model based on DAWN, taking its framing camera, with a cross track look angle of 2.735 degrees, as the sole instrument used for mapping and assume it is always nadir pointed. We then use a modified version of the comet 15P/Finlay as the target body. The model for Finlay was adjusted to set the radius to 25 meters in order to create very fast orbits for quicker simulations. This results in a polar orbit with a period of approximately 2.5 minutes. The spacecraft is limited to 70 Gibits of memory and the data rate for the framing camera is set to 0.17 Gibits per second. We focus on a mapping scenario between downlinks, so we do not model the downlink rate. Without data volume restrictions, it would take approximately 306 orbits to map the entire surface.

This mimics realistic scenarios, such as the recent ROSETTA mission. The ROSETTA mission, an international collaboration among the European Space Agency (ESA), the National Aeronautics and Space Administration (NASA), and some European national space agencies, highlighted the importance of studying comets as a means to better understand the origins of our solar system (Glassmeier et al. 2007). Part of the investigation included characterization of the surface using the Optical, Spectroscopic, and Infrared Remote Imaging System (OSIRIS), which mapped the surface with different filters (El-Maarry et al. 2015).

CLASP

The presented scenario was converted into spacecraft, instrument, and target body models as well as science goals that could be ingested into CLASP.

Spacecraft, Instrument, and Body Models

The spacecraft, instruments, and bodies are modeled through the use of spice kernels and Keyhole Markup Language (KML) files. Spice kernels define the spacecraft frames and trajectory; any target body's size, frames, trajectory, and orientation; as well as instrument field of view and position relative to the spacecraft. A KML file defines which instruments are to be used in the scenario, their modes, their cross track look angles, and data rates.

Science Campaigns

A request for scientific observations is specified in a *campaign* written in KML. A campaign is defined by regions on a body to be observed, which instruments and their modes are to be used, geometric constraints (e.g., illumination or distance), and a priority. Our scenario is interested in global coverage, therefore we defined regions that cover the entire body, with 6 regions evenly distributed around the equator going from ± 30 degrees latitude, and 4 evenly spaced at each pole extending from the equatorial panels to ± 88 degrees latitude (as appropriate for the pole).

Algorithm

The CLASP algorithm uses all of the aforementioned inputs as well as a gridded representation of the surface to calculate and select observations. The gridded representation of the surface is generated by selecting a number of grid points that should go around the equator. This defines the separation between grid points, D. Lines are created at constant latitudes such that the lines are D apart and the spacing between gridpoints in the line are also spaced at D intervals. For our scenarios, we use 800 gridpoints around the equator, which results in approximately 200,000 gridpoints.

After the set of gridpoints have been established, CLASP computes visibility swaths per instrument based on the trajectory of the spacecraft, the field of view of the instruments, and the position and orientation of the target body. CLASP uses the CSPICE Toolkit provided by the Navigation and Ancillary Information Facility (NAIF) (Acton Jr 1996; Acton et al. 2018) to calculate the visibility swaths. With these visibility swaths, CLASP calculates the intersection with campaigns based on gridpoints that are within the polygons for both the instrument swaths and the campaign regions of interest, resulting in *potential observation records*. A potential observation record is created for each unique instrument mode that satisfies an observation. In other words, an observation record is created at a gridpoint in a visibility swath if the visibility of the instrument meets all of the constraints for a campaign at that point. Multiple observation records can be generated at the same point if they are for different instruments. If the same instrument is used to satisfy multiple campaigns at the same point, only one potential observation record is created. The observation record inherits the priority for the highest priority campaign it satisfies.

CLASP then uses Squeaky Wheel Optimization (SWO) with a priority-based, greedy scheduler to schedule potential observation records. SWO operates in two loops: the outer loop, which adjusts the priorities of the potential observation records that are inputs to the scheduler, which makes up the inner loop. The scheduler loops through each observation record in priority order, attempting to add each observation to the schedule. When an observation record is added, all other observation records that are in the same swath polygon for the same instrument must also be added. This set of obserservation records are all added to the schedule if no spacecraft constraints are violated. When the inner scheduling loop has completed, the outer loop increases the priority for any observation records that were not scheduled (i.e., the squeaky wheels). After the last iteration, the best scoring schedule, calculated based on the original campaign priorities, is returned. For our scenario, we only use one iteration of SWO since there is only one campaign with a single priority for global coverage.

MEXEC

MEXEC is an onboard scheduling and execution software that operates on *Task Networks* with a multi-mission design. It monitors system state to respond to current conditions and projects effects forward in time in order to predict future state. It uses this information to schedule and execute tasks, detect conflicts or constraint violations, and revise its current schedule accordingly. MEXEC consists of three major components, the *Timeline* Library, the *Planner*, and the *Controller*, whose interactions are shown in Figure 1.

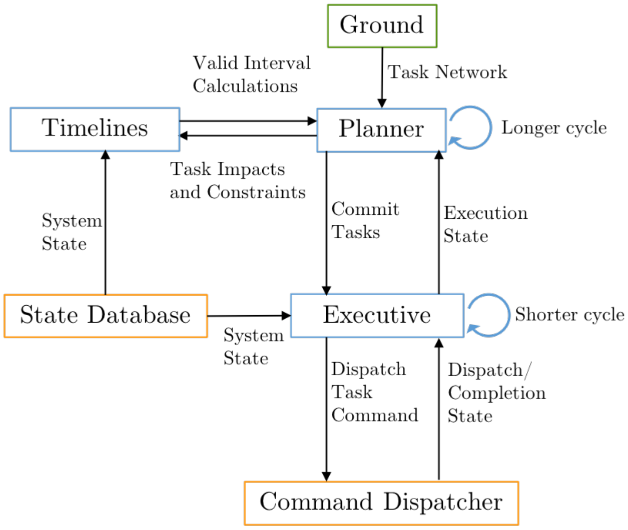


Figure 1: Diagram of the interactions between components in MEXEC.

Task Networks

Task Networks allow for encoding of intents and constraints that are lost when converting ground generated plans into sequences. Task Networks for MEXEC consist of tasks, templates, and timelines. A task represents the smallest operable unit in MEXEC and contains resource and timing constraints, expected effects (impacts), a command to execute (if applicable), contingencies in case of failure (if desired), and some metadata for goal tasks. Tasks can be used to represent Goals, which are higher level tasks that decompose into lower level tasks to achieve a desired objective. For the case of robust mapping, the Goal task stores which parts of the body are covered or uncoverd by the lower level observation tasks, with the goal of maximizing that coverage. Lower level tasks associated with a goal can come from existing tasks or from templates. Templates are tasks that are not explicitly requested to be scheduled, but are instead used by the Planner to either satisfy constraints or goals in the schedule. Timelines represent the states and resources used by tasks.

Timeline

Timelines provide a representation of past and future (predicted) system state. The current state is incorporated by subscribing to periodic state updates from the flight system for the most recent state. They also store the impacts of scheduled tasks and extrapolate them in order to predict future state. This can then be used to calculate valid intervals in which new tasks can be scheduled conflict free. Valid interval calculation looks at all of the current contraints and impacts on the timelines and provides intervals where adding the new task's constraints and impacts will not create any new constraint violations. The Timeline library is also used independently of the other components of MEXEC for other software, such as the M2020 Rover's Onboard Planner (Rabideau and Benowitz 2017).

Planner

The Planner runs on a cycle with several actions taken during each iteration. It first commits upcoming scheduled tasks within a *commit window* to the Controller to be executed. Next, the Planner uses valid interval calculation provided by Timeline to schedule any unscheduled tasks. The Planner then looks for any constraint violations, tries to repair them by moving and adding tasks. Lastly, it attempts to improve the schedule by shifting tasks closer to their preferred start times. The Planner will continuously receive updates from the Controller in order to maintain an accurate estimate of the current state. If a task comes back from the Controller with failure, the Planner will process the contingency actions defined in the Task.

Controller

When a Task is committed, ownership of that task is transferred from the Planner to the Controller. The Controller runs on a shorter cycle than the Planner and converts the committed task's constraints into control conditions expressed as a boolean expression tree. Like the other components, the Controller subscribes to system state in order to monitor when a task's constraints are met. When all constraints specified for the beginning of the task, including the start time, are met, the command defined in the task is dispatched. Each task then follows an internal state machine, constantly checking that all constraints on the task continue to be met. The Controller also sends updates to the Planner as tasks progress through their states, including the completion status when the task completes.

Approach

To generate the Task Network used by MEXEC for robust mapping, we use CLASP to generate three different sets of observations with varying time and data volume parameters. A contact diagram of the flow of information from CLASP to MEXEC can be seen in Figure 2.

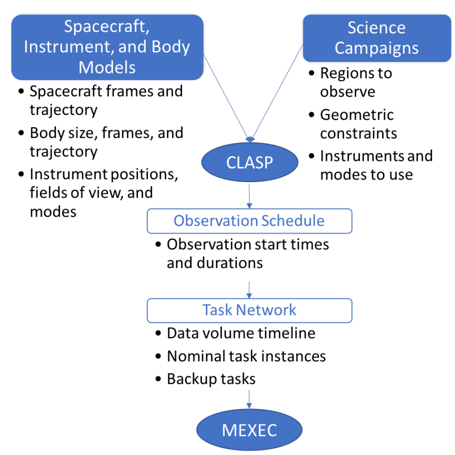


Figure 2: Contact diagram for the flow of information from the inputs to CLASP to MEXEC.

Generation of the Nominal Schedule

We first generate a baseline, or nominal, schedule of observations for one mapping cycle between downlinks using the described motivating scenario, but with the assumption that there are only 64 Gibits of data volume available onboard. The resulting schedule generates 367 distinct observations covering 14,183 gridpoints, roughly 7 percent of the total gridpoints on the body. The observations generated by CLASP in this run are shown in Figure 3 and are used as the *nominal* observation schedule, constituting an accurate schedule if the ground estimate of available data volume is accurate and if all observation tasks are executed without fault. In either of the two cases where the model deviates from actuality, science oppurtunities could be lost. MEXEC will be able to use the backup observations generated by the final CLASP schedule to fill in the gaps and maximize science.

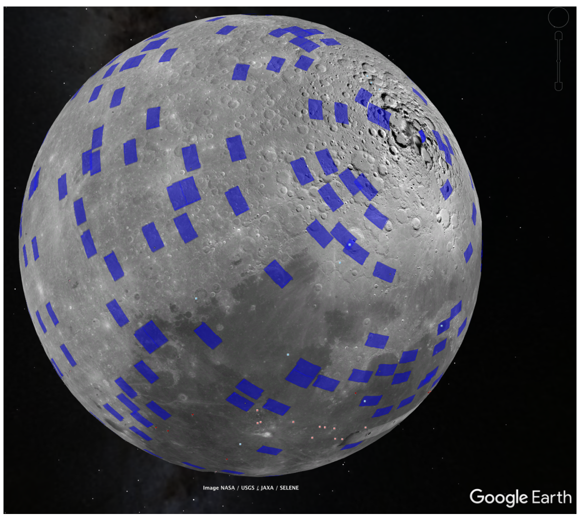


Figure 3: Surface coverage of the data limited nominal observation schedule.

Generation of the Backup Observations

Generating the backup observations requires two additional CLASP runs. In the first additional run, we re-run the scenario, but with no data volume limit, which results in an observation schedule that covers the entire body once. This observation schedule is discretized into unique 1 second observations, which we call *coverage* observations that are used to segment the body into *observation IDs*. These IDs are used as an estimate for the gridpoints used by the initial CLASP schedule generation. For this experiment, the body is segmented into 6,257 observation IDs covering 202,905 CLASP gridpoints. This segmentation helps MEXEC determine if a particular area is already covered by another scheduled observation.

The final CLASP iteration is run with unlimited data volume and 5 images requested per gridpoint, generating 26,124 one second tasks. These observations are used as the *backup* observations, i.e., the tasks that MEXEC can add in order to take advantage of extra data volume in a useful way or to replace a failed observation. These backup observations are a super set of the first two CLASP schedules, containing at least 5 observations of each gridpoint. There are not exactly five times the 6,257 observation IDs as some of the backup observations work as backups for multiple observation IDs.

Using all of the resulting observation schedules, we determine which observation IDs are covered and by which observations. An observation is considered to cover an observation ID if the overlap in coverage between the observation being considered and the coverage observation corresponding to the observation ID is greater than 75%. The overlap in coverage between two observations is the size of the intersection between their observation polygons. This is done for both the nominal and backup observations, as either can cover multiple observation IDs, as these IDs are only an estimate of gridpoints. This makes backup observations a good attempt at replacing lost science, but feasibly a backup observation could not cover up to 25% of the original observation gridpoints. Overlap between the backup observations is not considered.

Scheduling the Observations in MEXEC

To initialize MEXEC, a timeline is created to represent the data volume with the constraint that the value must remain between 0 and the data volume limit. A template task for backup observations is created with an impact to increase the data usage rate by 0.17 for the duration of the task. Every nominal and backup observation task can then be created as an instance of this template. MEXEC is first provided with the number of observation IDs in this experiment, as well as the times of both the earliest and latest observations. It uses this information in order to create a goal task, a task with no execution command that spans the time of the entire observation set and stores whether each observation ID is covered by one of the observation tasks, with the goal of maximizing the number of observation IDs covered.

MEXEC then reads the nominal observations, their observation IDs and start times, into fully specified tasks and schedules them. This is the nominal observation schedule generated on the ground by CLASP. As each task is scheduled by the MEXEC planner, its observation ID is marked as covered in the goal task. This is followed by a list of backup observations, each with a geometrically calculated list of observation IDs that they cover sufficiently. While MEXEC runs, if there is data volume available, MEXEC will schedule a backup observation task that covers at least one uncovered observation ID. Data volume could be available either because some other task failed to execute or because the initial CLASP data volume model was too conservative. For this initial implementation, memory usage was not considered when uploading observations and backups to MEXEC. Each task is relatively small, consisting of a few integers, however there can be many tasks.

Simulations in MEXEC

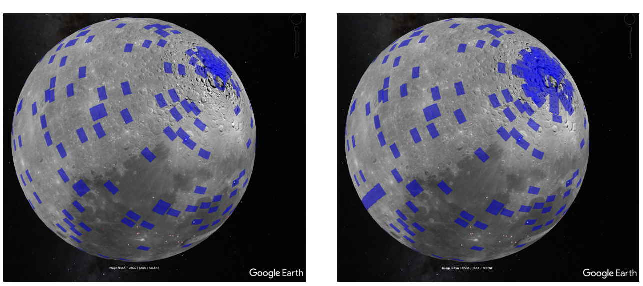
We simulated three different scenarios in MEXEC. We first look at the case where the ground estimate of data volume used by CLASP to generate the nominal schedule was too conservative and MEXEC is able to take advantage of the additional available data volume onboard. Next, we report task failure for every fixed number of observations to observe MEXEC replacing the lost science with backup observations that cover at least one additional observation ID. The assumption is made that if an observation fails, the entire observation is not achieved and there is no partial completion of the observation. We then look at a case where the ground overestimated data volume availability and MEXEC is unable to schedule the entire nominal schedule. For each run, a list of observations that were scheduled and executed successfully is produced, and from this list we calculate the number of original CLASP target gridpoints that were covered.

Results

Figure 4 shows the observations scheduled when MEXEC is able to determine that there is more data volume available onboard than was anticipated when generating the nominal schedule. As the discrepency between anticipated and actual data volume onboard increases, a higher percentage of additional gridpoints are covered, as shown in Figure 5. The relationship between increase in data volume and increase in coverage is expected, considering that every observation is the same length and gridpoints are equally distributed around the body.

When the onboard data volume is less than was anticipated when generating the nominal schedule, a similar expected relationship is seen. These observations are displayed in Figure 6 and show more sparse observations than the nominal. Figure 7 indicates that the linear relationship is maintained, regardless of whether the anticipated data volume was over or under estimated. The less data volume that is available, the fewer observations MEXEC is able to schedule.

MEXEC handles observation task faults by replacing the observation with another task that covers at least one observation ID currently uncovered in the schedule. By doing this, MEXEC is able to minimize the effect of task faults on the overall amount of coverage, as shown in Figure 9.



(a) 70 Gibit Data Volume (b) 80 Gibit Data Volume

Figure 4: Surface coverage when the nominal schedule is based on an underestimate of the available data volume onboard. The figure shows the results when the ground underestimate of data volume is 64 Gibits and the available data volume onboard is (a) 70 Gibits and (b) 80 Gibits.

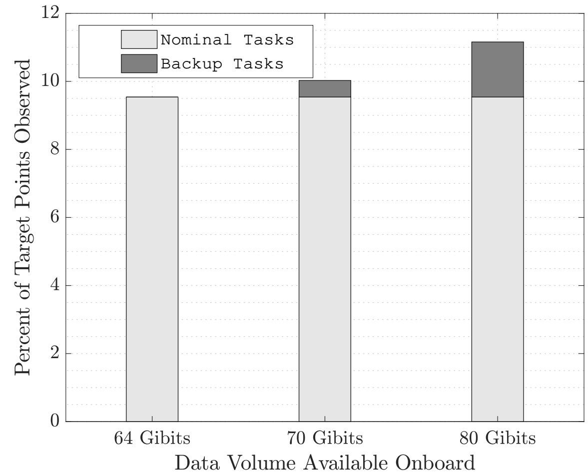
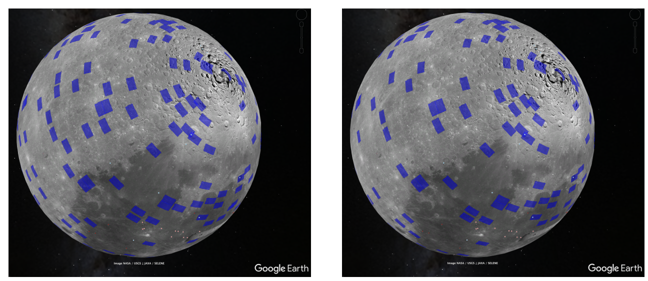


Figure 5: Percent of gridpoints observed broken down by nominal and backup tasks when the ground uses an underestimate of data volumen available.



(a) 60 Gibit Data Volume (b) 50 Gibit Data Volume

Figure 6: Surface coverage when the nominal schedule is based on an overestimate of the available data volume onboard. The figure shows the results when the ground overestimate of data volume is 64 Gibits and the available data volume onboard is (a) 60 Gibits and (b) 50 Gibits.

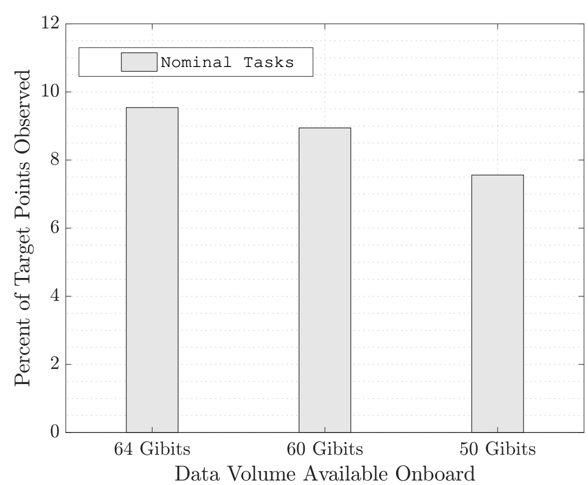
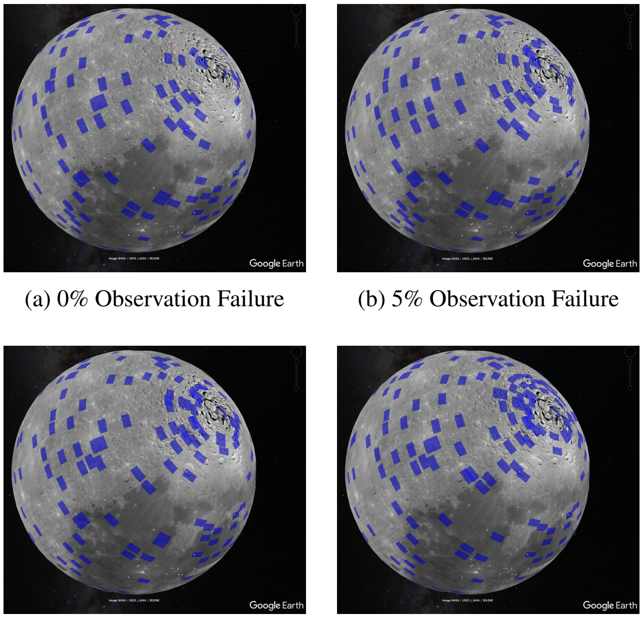


Figure 7: Percent of gridpoints observed when the ground uses an overestimate of data volumen available.



(a) 0% Observation Failure

(b) 5% Observation Failure

(c) 10% Observation Failure

(d) 20% Observation Failure

Figure 8: Surface coverage when (a) 0%, (b) 5%, (c) 10%, and (d) 20% of tasks fail during execution and backup tasks are scheduled to take advantage of the available data volume.

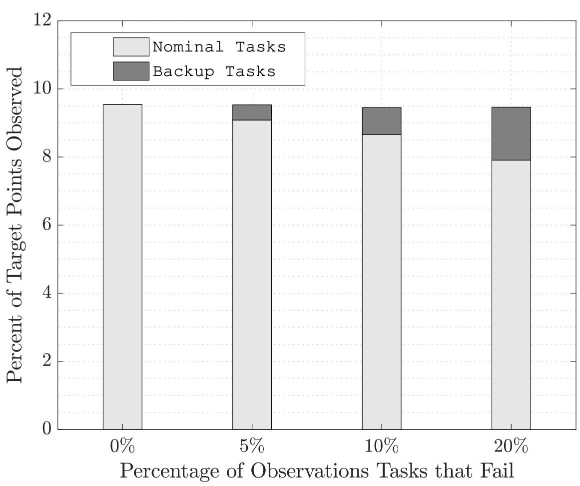


Figure 9: Percent of gridpoints observed broken down by nominal and backup tasks when tasks fail during execution.

The percentage of gridpoints covered does not meaningfully change, regardless of how often observation tasks fail. This makes the mapping very robust to faults in any part of an imaging task, as the planner can make up for the lost science. Figure 8 shows the observations scheduled with different task fault rates, and shows that the overall number of observations scheduled only changes negligibly.

Using Scenario A as an example, robust mapping in this way is not computationally prohibitive. The initial scheduling of the nominal observations is the slowest process, taking an average of 40ms upfront, with the backup observation and goal calculations taking an average of 8ms per cycle. These simulations were run with a 3.1 GHz Intel Core i7 processor, significantly more powerful than the average flight computer.

Related Work

Over the years, we have seen a tremendous amount of work on AI planning and execution systems for spacecraft and robotics (Gat et al. 1998; Simmons and Apfelbaum 1998; Muscettola et al. 1998; Frank et al. 2001; Verma et al. 2005; Rasmussen, Ingham, and Dvorak 2005). A few have been used to autonomously control spacecraft operating in space (Jónsson et al. 2000; Chien et al. 2005; 2016). At a high-level, many of these have a similar architecture with separate planner and executive components. The role of the executive, however, is quite different in each. In (Jónsson et al. 2000), a smart executive reasoned about possible execution time ranges and projections into the future. (Chien et al. 2005) used separate planning and execution models, where activities planned using a declarative model, would initiate procedural scripts specified in the Spacecraft Command Language (SCL). In (Chien et al. 2016), a simple sequencer executes the activities at absolute times assigned by the planner, leaving the planner to handle all plan repair changes, big and small.

Using lessons learned from previous systems, MEXEC was first developed as a prototype for the Europa Clipper mission to show onboard fault recovery (Verma et al. 2017). A similar design was prototyped, and later adopted for the Mars 2020 rover (Rabideau and Benowitz 2017). The MEXEC planner and controller were developed in conjunction to ensure a consistent, cooperative design. The same declarative constraint model used to plan an activity is also used to execute the activity with some flexibility in its start and end times. To ensure timely execution of activities, the MEXEC controller focuses only on the issues that occur near the current time, and is limited to making small changes to the plan. The MEXEC planner is then left to handle future, potential problems that may require more widespread plan changes.

The charter for MEXEC has not been to provide a new AI control architecture, but instead to be a new component in low-risk, heritage flight software and operations. MEXEC code was written completely from scratch. It inherits design from the literature but was developed to meet current flight software requirements and constraints, and allow human operators the ability to control the level of autonomy. A stripped down version is planned to be uplinked to ASTERIA in June 2019. ASTERIA is a Cubesat originally used to detect transitting exoplanets, now used for research and experiments in its third extension (Smith et al. 2018).

CLASP has been used to evaluate potential coverage mapping for the planned Europa Clipper and JUipter ICy moons Explorer (JUICE) missions (Troesch, Chien, and Ferguson 2017). In those scenarios, the goal was to determine what the potential maximum coverage could be, given no data volume restrictions, in contrast to the work in this paper, which considered data volume available and selected observations to execute. CLASP has also been used for mission studies for the NISAR mission (Doubleday and Knight 2014). The Orbiting Carbon Observatory-3 (OCO-3) mission, scheduled to launch in April 2019, is using CLASP to generate the nominal schedule for multiple different observation types (Moy et al. 2019). CLASP was used as the ground scheduler for the IPEX CubeSat, but used the CASPER planner onboard to monitor resources, adjust the schedule, add follow-on activities based on feature detection, and add lower priority observations that did not make it into the original schedule (Doubleday et al. 2015). The ECOSystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) instrument has used CLASP for operations (Yelamanchili et al. 2019).

Other recent orbital mapping work includes DAWN, the spacecraft for which on which our model is based, which used framing camera images to map from various altitudes, including Survey Orbit (Roatsch et al. 2016a) and Low Altitude Mapping Orbit (LAMO) (Roatsch et al. 2016b). Together, the multiple maps are combined to create an Atlas of Ceres, a dwarf planet and the largest object in the Astroid Belt. These images and others, together with Geographic Information Systems (GIS) software have also been used to develop a geologic mapping of Ceres (A.Williams et al. 2018). Other mapping missions have also shown more complex mapping cases. In our approach, the body considered was generalized to be convex. The ROSETTA mission, specifically observations using OSIRIS, has demonstrated more challenging mapping scenarios such as coverage of a non-convex body and more complex illumination requirements (Preusker, F. et al. 2015).

Future Work

In this approach, the assumption was made that all grid-points and areas of the body were of equal priority to image. This is not always the case, which means that MEXEC should make sure the chosen backup is the best choice, rather than picking the first available choice.

An additional change could be made to allow for different sized observations. Rather than only specifying one second observation tasks, treating an entire observation swath as one task would save memory and speed up valid interval calculation. The difficulty with this approach is that calculating backups becomes more complex and unclear when observations are of varying size, but it also makes the problem more interesting, with MEXEC only able to choose backups that do not exceed the data volume limit. Presently all observation tasks require the same amount of data volume.

Using larger observation sizes is just one way in which memory could be saved, and looking into other memory saving optimizations is another direction to explore in the future. Storing all of the constraints, impacts, parameters, states, etc. of a task leads to large memory needs when there are many tasks. On our work with ASTERIA this has already begun to limit the number of tasks that MEXEC can simultaneously handle. In order to make re-planning mapping tasks more robust and viable on flight missions, optimizing the memory usage of the system would be essential.

In the future, it may become more feasable to generate the backup observations onboard, given progress in flight computer capabilities. For example, CLASP uses ray tracing algorithms as part of its coverage calculations, and if future missions include GPUs onboard, CLASP would be greatly sped up and able to generate the nominal and backup schedules onboard.

Conclusion

We have presented a method for robust mapping using ground generated backup observations. Given an example scenario to map the surface of a comet with an orbiting spacecraft, we use CLASP to create a nominal observation schedule as well as backup observations. These observations are translated into a Task Network as tasks and goals to be scheduled and executed by MEXEC. We have demonstrated that when the system state onboard deviates from the estimated state when generating the nominal schedule, MEXEC can schedule tasks to maximize the use of the data volume constraint to observe the body's surface. In the case where there is less data volume onboard than expected, MEXEC schedules and executes as many of the tasks as allowed by the data volume restriction. In the case where more data volume is availabe than expected during nominal scheduling, MEXEC is able to add tasks that observe regions that have not been completely covered already.

Acknowledgments

We would like to thank some of our colleagues at JPL including Patrick Doran and Brian Kennedy for creating some of the spice kernels used in these experiments, Carolyn Ortega for describing the science campaigns, Rashied Amini and Lorraine Fesq for their support and encouragement, and Robert Bocchino for his feedback on MEXEC.

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

Acton, C.; Bachman, N.; Semenov, B.; and Wright, E. 2018. A look towards the future in the handling of space science mission geometry. *Planetary and Space Science* 150:9–12. DOI 10.1016/j.pss.2017.02.013, `https://doi.org/10.1016/j.pss.2017.02.013`.

Acton Jr, C. H. 1996. Ancillary data services of nasa's navigation and ancillary information facility. *Planetary and Space Science* 44(1):65–70.

A.Williams, D.; Buczkowski, D. L.; Mest, S. C.; Scully, J. E.; Platz, T.; and Kneissl, T. 2018. Introduction: The geologic mapping of ceres. *Icarus* 316:1–13.

Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davies, A.; Mandl, D.; Frye, S.; Trout, B.; Shulman, S.; and Boyer, D. 2005. Using autonomy flight software to improve science return on earth observing one. *Journal of Aerospace Computing, Information, and Communication (JACIC)* 196–216.

Chien, S.; Doubleday, J.; Thompson, D. R.; Wagstaff, K.; Bellardo, J.; Francis, C.; Baumgarten, E.; Williams, A.; Yee, E.; Stanton, E.; and Piug-Suari, J. 2016. Onboard autonomy on the intelligent payload experiment (ipex) cubesat mission. *Journal of Aerospace Information Systems (JAIS)*.

Doubleday, J., and Knight, R. 2014. Science mission planning for nisar (formerly desdyni) with clasp. In *International Conference On Space Operations (SpaceOps 2014)*.

Doubleday, J.; Chien, S.; Norton, C.; Wagstaff, K.; Thompson, D. R.; Bellardo, J.; Francis, C.; and Baumgarten, E. 2015. Autonomy for remote sensing - experiences from the ipex cubesat. In *International Geoscience and Remote Sensing Symposium (IGARSS 2015)*.

El-Maarry, M. R.; Thomas, N.; Giacomini, L.; Massironi, M.; Pajola, M.; Marschall, R.; Gracia-Berná, A.; Sierks, H.; Barbieri, C.; Lamy, P. L.; et al. 2015. Regional surface morphology of comet 67p/churyumov-gerasimenko from rosetta/osiris images. *Astronomy & Astrophysics* 583:A26.

Frank, J.; Jónsson, A.; Morris, R.; Smith, D. E.; and Norvig, P. 2001. Planning and scheduling for fleets of earth observing satellites.

Gat, E.; Bonnasso, R. P.; Murphy, R.; et al. 1998. On three-layer architectures. *Artificial intelligence and mobile robots* 195:210.

Glassmeier, K.-H.; Boehnhardt, H.; Koschny, D.; Kührt, E.; and Richter, I. 2007. The rosetta mission: flying towards the origin of the solar system. *Space Science Reviews* 128(1-4):1–21.

Jónsson, A. K.; Morris, P. H.; Muscettola, N.; Rajan, K.; and Smith, B. D. 2000. Planning in interplanetary space: Theory and practice. In *AIPS*, 177–186.

Knight, R., and Chien, S. 2006. Producing large observation campaigns using compressed problem representations. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS2006)*. Baltimore, Maryland.

Moy, A.; Yelamanchili, A.; Chien, S.; Eldering, A.; and Pavlik, R. 2019. Automated scheduling for the oco-3 mission. Submitted for publication to IWPSS2019.

Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no ai system has gone before. *Artificial intelligence* 103(1-2):5–47.

Preusker, F.; Scholten, F.; Matz, K.-D.; Roatsch, T.; Willner, K.; Hviid, S. F.; Knollenberg, J.; Jorda, L.; Gutiérrez, P. J.; Kührt, E.; Mottola, S.; A´Hearn, M. F.; Thomas, N.; Sierks, H.; Barbieri, C.; Lamy, P.; Rodrigo, R.; Koschny, D.; Rickman, H.; Keller, H. U.; Agarwal, J.; Barucci, M. A.; Bertaux, J.-L.; Bertini, I.; Cremonese, G.; Da Deppo, V.; Davidsson, B.; Debei, S.; De Cecco, M.; Fornasier, S.; Fulle, M.; Groussin, O.; Güttler, C.; Ip, W.-H.; Kramm, J. R.; Küppers, M.; Lara, L. M.; Lazzarin, M.; Lopez Moreno, J. J.; Marzari, F.; Michalik, H.; Naletto, G.; Oklay, N.; Tubiana, C.; and Vincent, J.-B. 2015. Shape model, reference system definition, and cartographic mapping standards for comet 67p/churyumov-gerasimenko - stereo-photogrammetric analysis of rosetta/osiris image data. *A&A* 583:A33.

Rabideau, G., and Benowitz, E. 2017. Prototyping an onboard scheduler for the mars 2020 rover. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

Rabideau, G.; Chien, S.; Mclaren, D.; Knight, R.; Anwar, S.; Mehall, G.; and Christensen, P. 2010. A tool for scheduling themis observations. In *International Symposium on Space Artificial Intelligence, Robotics, and Automation for Space*. European Space Agency Sapporo, Japan.

Rasmussen, R.; Ingham, M.; and Dvorak, D. 2005. Achieving control and interoperability through unified model-based systems and software engineering. In *Infotech@ Aerospace*. 6918.

Roatsch, T.; Kersten, E.; Matz, K.-D.; Preusker, F.; Scholten, F.; Jaumann, R.; Raymond, C.; and Russell, C. 2016a. Ceres survey atlas derived from dawn framing camera images. *Planetary and Space Science* 121:115–120.

Roatsch, T.; Kersten, E.; Matz, K.-D.; Preusker, F.; Scholten, F.; Jaumann, R.; Raymond, C.; and Russell, C. 2016b. High-resolution ceres low altitude mapping orbit atlas derived from dawnframing camera images. *Planetary and Space Science* 121:115–120.

Russell, C., and Raymond, C. 2011. The dawn mission to vesta and ceres. In *The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres*. Springer. 3–23.

Simmons, R., and Apfelbaum, D. 1998. A task description language for robot control. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, volume 3, 1931–1937. IEEE.

Smith, M.; Donner, A.; Knapp, M.; Pong, C.; Smith, C.; Luu, J.; Di Pasquale, P.; and Campuzano, B. 2018. On-orbit results and lessons learned from the asteria space telescope mission. In *Proceedings of the AIAA/USU Conference on Small Satellites, The Year in Review, SSC18-I-08*.

Troesch, M.; Chien, S.; and Ferguson, E. 2017. Using automated scheduling to assess coverage for europa clipper and jupiter icy moons explorer. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

Verma, V.; Jónsson, A.; Simmons, R.; Estlin, T.; and Levinson, R. 2005. Survey of command execution systems for nasa spacecraft and robots.

Verma, V.; Gaines, D.; Rabideau, G.; Schaffer, S.; and Joshi, R. 2017. Autonomous science restart for the planned europa mission with lightweight planning and execution. In *Proceedings of the 10th International Workshop on Planning and Scheduling for Space (IWPSS2017)*. Pittsburgh, PA.

Yelamanchili, A.; Chien, S.; Moy, A.; Shao, E.; Trowbridge, M.; Cawse-Nicholson, K.; Padams, J.; and Freeborn, D. 2019. Automated science scheduling for the ecostress mission. Submitted for publication to IWPSS2019.