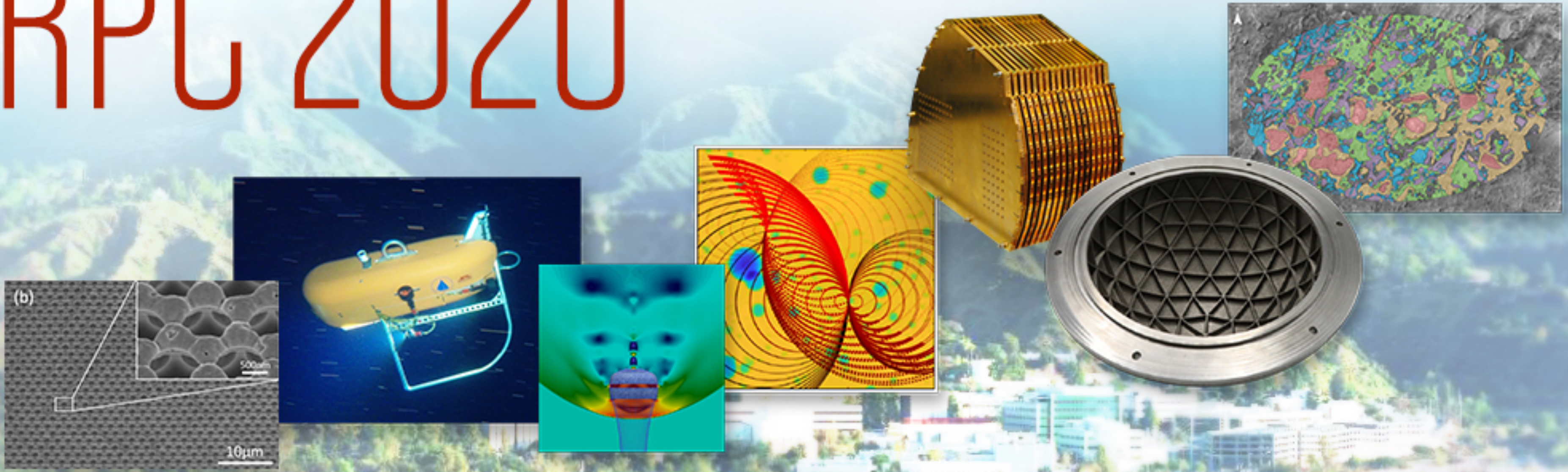
RPC 2020



Virtual Research Presentation Conference

Task-Based Commanding

Principal Investigator: Martina Troesch (397I)
Co-Is: Faiz Mirza (397I), Martin Feather (5125), Allen Nikora (5125)
Program: Strategic Initiative

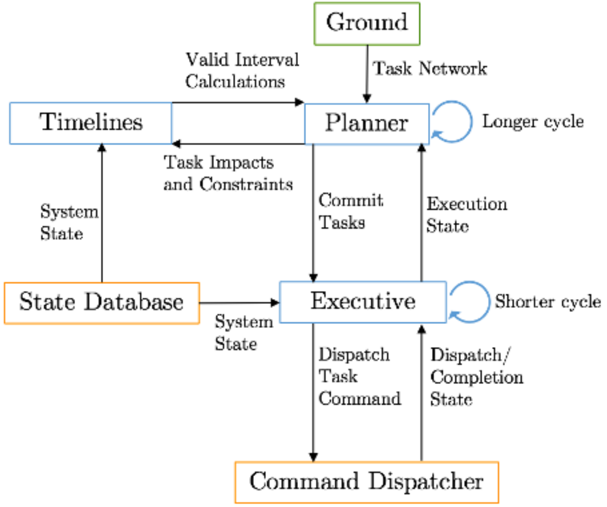
RPC-133

Jet Propulsion Laborato y
California Institute of Techn y

Tutorial Introduction

Abstract

Traditional spacecraft commanding is performed with time-based command sequences. However, these sequences limit the autonomy that can be achieved. Some sequences allow for control logic, but this provides only limited flexibility. Task-based commanding allows for opportunities for more autonomous behavior by maintaining within tasks the intentions used to generate a schedule. One way to perform task-based commanding is with task networks, which model spacecraft behavior and preserve information about constraints on tasks instead of only defining execution times for commands, as seen in sequences. Using an on-board planner and executive in conjunction with task networks allows for on-board constraint checking and scheduling with the most up-to-date information possible and also allows for the possibility of re-planning in reaction to unexpected events. We follow this approach in MEXEC (Multi-mission EXECutive) [9], a multi-mission, task-based planner and executive that performs flexible, on-board decision making based on current system state. It propagates state information to build and maintain conflict-free schedules and replans if there are unexpected changes.



A diagram of MEXEC components (shown in blue) and its interactions

Problem Description – Context

1. Time-based command sequences have limited flexibility if the on-board state is not as expected
    - By defining tasks according to their intentions with execution windows and resource and state constraints, MEXEC's on-board planner is able to re-plan when there are deviations in execution.
    - MEXEC's executive monitors real-time state to ensure constraints are satisfied.
2. Time-based command sequences do not allow for dynamic insertion, removal, or re-ordering of activities in response to on-board events
    - MEXEC performs on-board re-planning using task templates and can adjust the schedule in response to an event
3. Future missions will require more autonomous capabilities, which can be enabled and developed in MEXEC
4. The ASTERIA CubeSat testbed and workstation provide a flight-like environment for development and demonstrations

Objective: Demonstrate the benefits of and increase the confidence in task-based commanding and onboard planning using MEXEC, show that MEXEC can be made flight-ready, and to develop MEXEC capabilities for future missions. To that end, experiments were performed on the ASTERIA CubeSat testbed and workstation.

Problem Description – Comparison to SOA

Some sequencing languages can provide higher-level programming capabilities:

- VML used on Spitzer [7]
  - Higher-level programming with sequences
  - Pre-identified observations added from a list based only if relative timing shows there is space
  - No re-planning

A shift in paradigm to goal-based commanding has been shown in:

- Remote Agent on Deep Space One [5, 6]
  - Goal-based commanding using model-based programming and onboard search
  - Flew for 48 hours
  - Allocation for RAX: 32MB RAM, 16MB file space, 45% of the CPU
- Autonomous Sciencecraft Experiment on Earth Observing One [3]
  - CASPER
  - Maximize science return by creating new goals from processed data and re-planning
  - Flew for over 12 years
  - Allocation for ASE: 128MB RAM, 4MIPS
- M2020 Onboard Planner [8]
  - Planner and executive that uses tasks (activities)
  - Timeline library shared with MEXEC
  - Scheduled for operational use in late 2021

MEXEC similarities with Remote Agent, ASE, and M2020 OBP:

- Separate Planner and Executive
- Periodic re-planning up to a fixed time horizon
- Use of a representation like timelines

MEXEC differences:

- Has a consistent, tighter integration between the Planner and Controller with a shared model and representation
- Designed to be a component in low-risk, heritage flight software
- Allows operators to control the level of autonomy
- Designed to be multi-mission (M2020 OBP is mission specific)

Problem Description – Relevance and Impact

Onboard, task-based planning, re-planning, and smart execution with MEXEC:

- Maintains intentions within tasks
- Models spacecraft behavior and preserves information about constraints
- Performs onboard constraint checking and scheduling with the most up-to-date information
- Reacts to unexpected events

Which:

- Allows opportunities for more autonomous behavior
- Enables missions that are not feasible with humans in the loop
- Can increase science return
- Can increase robustness
- Can reduce cost

By advancing MEXEC, integrating it with the ASTERIA F' FSW, uploading it to the ASTERIA testbed or workstation, running task networks, and using V&V techniques, we demonstrated MEXEC's capabilities in a flight-like scenario, which matured the technology, raised the TRL, and provided confidence for future missions to adopt it.

Methodology

We used a use case and scenario driven approach with 3 experiments:

1. **Momentum Management**
2. **Autonomous Orbit Determination (OD)**
3. **Model-based Fault Protection**

Methodology – Momentum Management

Background

- External torques cause momentum buildup, which can cause attitude control loss
- Reaction wheel zero crossings reduce science quality

With traditional commanding

- Ground tools predict momentum buildup and zero crossings
- Observations and momentum dumping are scheduled conservatively
- Fault Protection’s response to momentum increase is to safe and reset the spacecraft, which may lead to loss of science

With MEXEC

- Use states onboard to respond and re-plan
- Schedule observations less conservatively to get *more or better science*
    - Monitor momentum *on board*
    - Dump momentum *only when need to*
- Unexpected momentum increase *detected and responded to*
    - Avoids drastic safing or reset
    - Resume observations when OK to do so
    - Maximizes science

Objectives:

- Use MEXEC’s planner and executive to react to events
- Provide a benefit to ASTERIA
- Use task networks with state and resource constraints and impacts
- Priority-based scheduling
- Perform re-planning
- Allow for less conservative scheduling

New/Improved Capabilities:

- internal states
- cleanup impacts
- skip condition
- controller timing tolerance
- impact consolidation

Methodology – Autonomous OD

Background

- ASTERIA was an Earth orbiting spacecraft taking science observations
- Accurate orbit knowledge is required for science
- Knowledge of orbit decays due to uncertainty of effects of external forces
- AutoNav is an autonomous navigation technology demonstrated on Deep Space One and various other missions [1, 2]

With traditional commanding

- Ground keeps track of orbit

With MEXEC and AutoNav

- Autonomous upkeep of orbit knowledge onboard
- Monitor onboard system state to proactively schedule *only when needed* by selecting appropriate tasks to address impending conflicts
- Plan to maintain goals in the future and deconflict resource constraints based on priorities

Objectives:

- Objectives from Momentum Management
- Use hierarchical tasks
- Achieve a goal by planning using templates
- Plan considering exogenous events
- Integrate with an onboard expert (AutoNav)
- Use repair and optimize techniques
- Priority-based scheduling with conflicting resources

New/Improved Capabilities:

- hierarchical tasks
- interface with expert components

Methodology – Model-Based Fault Protection

Background

- Builds on the Autonomous OD scenario
- Some tasks require a healthy ACS throughout execution
- Pointing must be held while taking an image for AutoNav, otherwise the image should not be used and follow-on tasks to perform processing should not occur due to the lack of valid data
- The Model-Based Off-Nominal State Identification Detection (MONSID) software propagates sensor data through a behavioral model to determine the health status of a component [4]

With traditional commanding

- Do not know onboard whether the ACS component is functioning properly and data can be trusted

With MEXEC, AutoNav, and MONSID

- Use ACS health status as reported from MONSID to determine validity of ACS state
- Fail or skip tasks based on constraints

Objectives:

- Objectives from Momentum Management and Autonomous OD
- Incorporate MONSID's reporting of ACS health to provide information about validity of state values
- Handle ACS faults to show that the overall architecture design and task network responded as desired

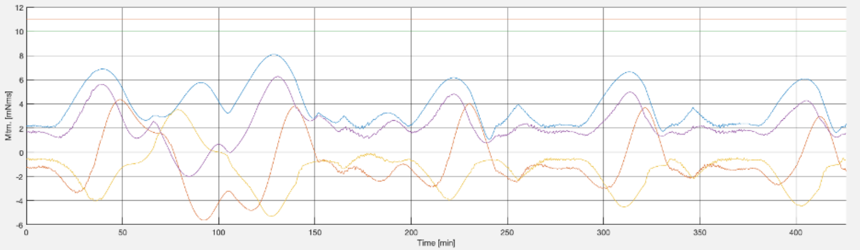
Results – Momentum Management

Performed on the **testbed** in February, 2020

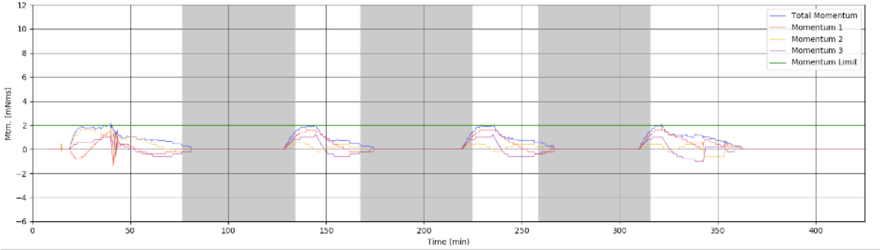
- Used the Realtime Dynamics Processor (RDP) to simulate sensor input
- Selected observations that were expected to cause momentum buildup past the response threshold
- Passed the momentum threshold 4 times and recovered successfully each time to resume operations

Significance

- MEXEC successfully responds to onboard state in the planner and executive with task failures, cleanup commands, contingencies, and re-planning
- MEXEC could potentially have helped ASTERIA by preventing spacecraft resets if observations were scheduled less conservatively to increase science quality, if the ground tools modeled momentum buildup incorrectly, or in the case of operator error

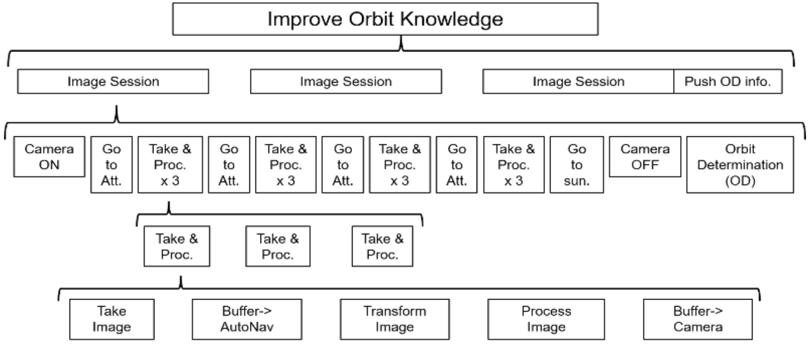


Ground predicted momentum versus time.



RDP experienced momentum versus time.
Grey areas show times of observations

Results – Autonomous OD



Improve Orbit Knowledge

Image Session
Image Session
Image Session
Push OD info.

Camera ON
Go to Att.
Take & Proc. x 3
Go to Att.
Take & Proc. x 3
Go to Att.
Take & Proc. x 3
Go to Att.
Take & Proc. x 3
Go to sun.
Camera OFF
Orbit Determination (OD)

Take & Proc.
Take & Proc.
Take & Proc.

Take Image
Buffer-> AutoNav
Transform Image
Process Image
Buffer-> Camera

Overview of task network structure used to achieve the goal of orbit knowledge confidence

Performed on the **development PC** in August, 2020

- Timelines, templates for the orbit knowledge goal, and tasks for high priority comm passes and low priority science observations were loaded
- MEXEC reacted to impending violation of orbit confidence goal by inserting a hierarchical task to address the issue
- Decomposed the hierarchy, including querying AutoNav for additional task information
- Successfully scheduled all tasks, deconflicting constraints between different priority tasks
- All tasks executed as expected
- New improved orbit knowledge confidence and expected decay determined by AutoNav

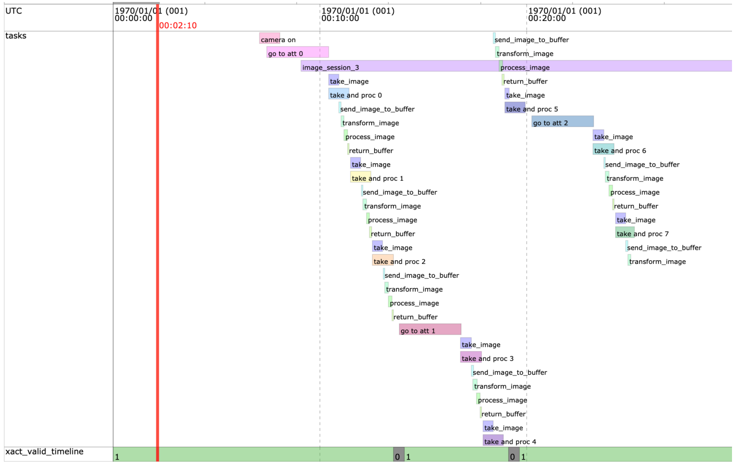
Significance

- Plan and maintain goals with conflict-free schedules and execute tasks while respecting constraints
- Monitoring of and reaction to onboard system state meant that orbit knowledge was recalculated only when needed through MEXEC's selection appropriate tasks to address impending conflicts
- Successful information exchange during MEXEC's interaction with an onboard subsystem

Results – Model-Based Fault Protection

Performed on the **development PC** in September, 2020

- ACS telemetry data was fed into the system through files generated from flight data
- Faults injected at 3 different times by commanding a change in model gains in MONSID
- 1) Fault injected during a task not requiring healthy ACS and cleared before tasks requiring healthy ACS were scheduled to execute resulting in tasks correctly executing as nominal
- 2) Fault injected during an AutoNav imaging task, correctly causing the task to terminate and follow-up analysis tasks to be skipped
- 3) Fault injected during a slew, which requires a healthy ACS to complete, but the fault cleared before the end of the task, so the slew was successful, the task completed normally, and following tasks continued as nominal, as expected



Execution of a run of the model-based fault protection scenario where two faults were injected, as seen on the xact_valid_timeline when the value transitions to 0

Significance

- Successful cooperative interaction of MEXEC, AutoNav, and MONSID working together and operating concurrently in the F’ FSW
- Confirms that MEXEC can react to unexpected states and events

Next Steps

- Continue to develop MEXEC capabilities to facilitate autonomous behavior
- Develop tooling to write task networks
- Use explainable AI techniques to aid in communicating MEXEC's behavior
- Foster confidence in MEXEC for integration in larger flight missions

Publications and References

PUBLICATIONS

[A] Troesch, M.; Mirza, F.; Hughes, K.; Rothstein-Dowden, A.; Bocchino, R.; Donner, A.; Feather, M.; Smith, B.; Fesq, L.; Barker, B.; Campuzano, B. (2020, October). “MEXEC: An Onboard Integrated Planning and Execution Approach for Spacecraft Commanding.” To be published in *Workshop on Integrated Execution (IntEx) / Goal Reasoning (GR), International Conference on Automated Planning and Scheduling (ICAPS IntEx/GP 2020)*, October 2020. Also to appear *at International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS 2020)* as an abstract.

[B] Feather, M.S., et al, (2021) "System-Level Autonomy for Spacecraft-a Demonstration”, In submission *IEEE AeroSpace Conference*, Big Sky, MT.

REFERENCES

[1] Bhaskaran, S. (2012). "Autonomous Navigation for Deep Space Missions." In *SpaceOps 2012*.

[2] Bhaskaran, S.; Riedel, J.; Synnott, S.; and Wang, T. (2000, August). "The Deep Space 1 Autonomous Navigation System: A Post-flight Analysis." In *Astrodynamics Specialist Conference*.

[3] Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davis, A.; Mandl, D.; 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,* 2(4), pp.196-216.

[4] Kolcio K.; and Fesq, L. (2016). "Model-based Off-nominal State Isolation and Detection System for Autonomous Fault Management." In *2016 IEEE Aerospace Conference*, IEEE.

[5] 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), pp.5-47.

[6] Nayak, P.; Kurien, J.; Dorais, G.; Millar, W.; Rajan, K.; Kanefsky, R.; Bernard, E.D.; Gamble Jr, B.E.; Rouquette, N.; Smith, D.B.; and Tung, Y.W. (1999, August). “Validating the DS1 Remote Agent Experiment.” In *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS '99*, held 1-3 June, 1999 in ESTEC, Noordwijk, the Netherlands. Edited by M. Perry. ESA SP-440. Paris: European Space Agency, 1999., p.349.

[7] Peer, S., and Grasso, C. A. (2005). “Spitzer Space Telescope Use of the Virtual Machine Language.” In *2005 IEEEAerospace Conference*. IEEE.

[8] Rabideau, G.; and Benowitz, E. (2017, June) “Prototyping an Onboard Scheduler for the Mars 2020 Rover.” In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*, Pittsburgh, PA.

[9] Verma, V.; Gaines, D.; Rabideau, G.; Schaffer, S.; and Joshi, R. (2017, June). “Autonomous Science Restart for the Planned Europa Mission with Lightweight Planning and Execution.” In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*, Pittsburgh, PA.