ExoOps: an analysis tool for electric propulsion powered missions

Orekit Day Presentation
23 05 2019

Mikael Fillastre / Lead Software Engineer
Andrea Fiorentino / Space Mechanics Engineer
INDEX

1. Exotrail
2. ExoOps
3. Why Orekit?
4. Orekit in ExoOps
5. Low-Thrust Maneuvers with Orekit – enhancement
6. Low-Thrust Maneuvers with Orekit – DSST add-on
7. System Engineering with Orekit
8. Technical issues and workarounds
9. Perspectives
Our vision & mission

Our mission
Delivering agility to small satellites

→ Reduce launch costs
→ Enhance launch flexibility
→ Improve picture resolution, coverage, link performance
→ Increase satellites lifespan
→ Reduce space pollution

Our vision
From propulsion to in-orbit servicing

First deliveries 2020

2022

2025

PROPULSION
Electric propulsion systems for 10-100kg satellites.
Mission software to design, optimise and operate propulsion manoeuvres.

PROXIMITY OPERATION
Proximity operation technologies for space surveillance and debris management.

ON-ORBIT SERVICING
On-orbit inspection, delivery & assembly of spacecraft.
Exotrail Today

**Massy, France**
- 15 people
- Electric propulsion systems

**Toulouse, France**
- 4 engineers + 3 interns
- Mission software

**ExoMG™** Hall Effect Thrusters for small satellites
- Modular
- Affordable
- High thrust

**ExoMS** Mission Study for the space industry
- Responsiveness
- Understand the impact of propulsion
- Study new use cases & trade-offs

**ExoOps™** Mission Software for propulsion
- Cloud-based, user-friendly
- Propulsion mission design
- ACOSS Operation
ExoOps™ Mission Design

Understand the impact of propulsion

- Parametrical analysis for Phase 0 / A / B / C Mission Design
- Broad range of missions:
  - Altitude change – Inclination correction – Local Time Phasing – Anomaly phasing – Station keeping – Deorbitation – Collision Avoidance – GEO relocation - …
- Broad range of results:
  - $\Delta V$ – Mission Duration – Power Consumption – Fuel used – Thrust & Attitude sequence – …

Comparing different solutions

- Compare the impact of different propulsion technologies on your system and your mission

Optimize launch strategies

- Trade-off between rideshare + electric propulsion & dedicated launchers becomes easy with our software.
- Analyze quickly the impact of various launch scenarios on your orbital deployment timing and your costs.

Cloud-based, easy-to-use software

- Cloud-based software
- Local data storage possible
- Easy-to-use, even for non experts
- Periodic license: no very large one-off fee
- SaaS model: regular updates included
Based on validated literature

Validated against Celestlab toolbox

Validated against numerical propagation models

Event-based propagation and mission planning

Full modeling of perturbations: Earth potential, atmospheric drag, third-body, etc.

Power generation and depletion

AOCS inputs and commands
ExoOps™ Mission Design

Do not disclose without the explicit consent of Exotrail
Why Orekit?

A few reasons behind our choice

- Longstanding open-source project
- High profile users from both the academic and industrial worlds
- Developers/users interaction
- Thoroughly validated
- Orekit day...😊
Orekit in ExoOps

How does it help us achieving our goal

- Space Mechanics
  - Orbit/Attitude

- Propagators
  - Numerical/Semi analytical

- Mathematics
  - Hipparchus library

OREKIT

Missions

- Positioning
- RAAN Phasing
- AOL Phasing
- Station Keeping

ExoOps
Low-Thrust Maneuvers with Orekit
Enhancing Orekit for continuous maneuvers’ design

**OREKIT**

- **ConstantThrustManeuver**
  - Fixed Direction in SAT frame
  - Start/Stop with Date Detectors
  - No available model for DSST

**ExoOps**

- **ConstantBurn**
  - Arbitrary direction in any frame
  - Any detector can be used as trigger
  - Added DSST low-thrust model
Low-Thrust Maneuvers with Orekit

A low-thrust module for the DSST propagator

AbstractGaussianContribution

• Compute mean elements
• Define thrusting interval

DSSTContinuousThrustManeuver

• Validated against numerical propagation
• Possibility of simulating satellite’s duty cycle
• Allows for fast estimation of maneuver cost
• Allows for fast dynamics propagation when computing optimized maneuvers
System Engineering with Orekit

Taking into account system-level constraints for maneuver planning

System Constraints
- Battery Level
- AOCS

Thrust Direction Computation

Maneuver Strategies
(optimize use of available power for propulsion)

Maneuvers Planning
(thrust opportunities can be discarded if they violate mission or system constraint)
Technical issues and workarounds

A few tips from our experience

- Osculating to mean elements conversion
  - Needs: convert osculating to mean elements to check maneuver convergence or compute control
  - Issues: slow and convergence sometimes fails,
  - Workaround: for detectors, use a mean step computed in a step handler. Customize DSST conversion.
Technical issues and workarounds

A few tips from our experience

- **Osculating to mean elements conversion**
  - Needs: in detectors to stop the maneuver or to output ephemerides,
  - Issues: slow and convergence sometimes fails,
  - Workaround: for detectors, use a mean step computed in a step handler. Customize DSST conversion.

- **System constraints defined by detectors**
  - Needs: stop the propulsion when an event occurs (e.g. low battery),
  - Issues: the detector stay at 0 when the event occurs,
  - Workaround: introduce a noise term to make the value vary.
Technical issues and workarounds

A few tips from our experience

- **Osculating to mean elements conversion**
  - Needs: in detectors to stop the maneuver or to output ephemerides,
  - Issues: slow and convergence sometimes fails,
  - Workaround: for detectors, use a mean step computed in a step handler. Customize DSST conversion.

- **System constraints defined by detectors**
  - Needs: stop the propulsion when an event occurs (e.g. low battery),
  - Issues: the detector stay at 0 when the event occurs,
  - Workaround: introduce a noise term to make the value vary.

- **Additional states in simulation’s output**
  - Needs: useful infos for the user (e.g. battery charge, power collected by SAs),
  - Issues: the evaluation of additional states “offline” might give unexpected results,
  - Workaround: investigation on-going.
Perspectives

Exchanging with the Orekit community

- Low thrust layer improvements
- Constrained Optimization in Hipparchus (NLP and TPBVP)
- Spacecraft and system modelling
- Augmented robustness of osculating to mean conversion and in detectors
Thanks for your attention!