#=============================================================================

#

# Flight Mechanics Analysis Tools Interoperability and Component Sharing (TI-18-01313)

#

# Monte-Copernicus Interface: Use Case 3.4, Copernicus to Monte transfer solutions

#

# By: Ricardo L. Restrepo (ricardo.l.restrepo@jpl.nasa.gov)

#

#=============================================================================

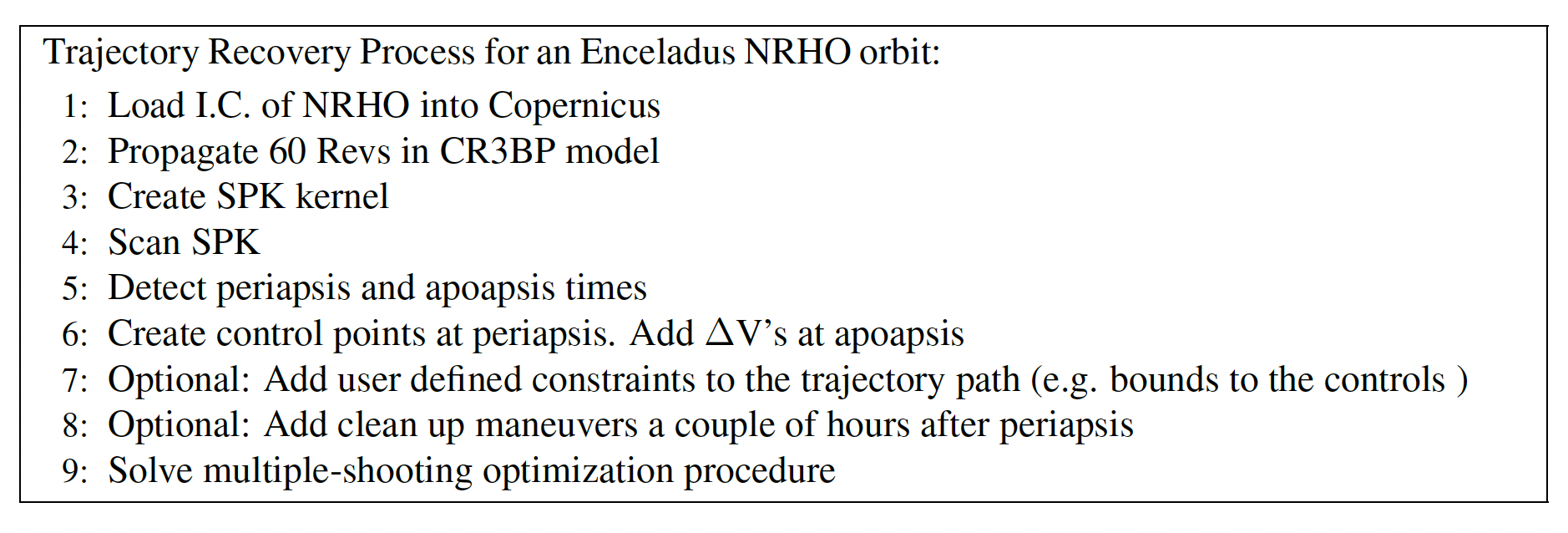
Use Case 3.4: Enceladus Orbiter. Transferring trajectory solutions from Copernicus to Monte. Enceladus Orbiters.

monteCop Script to be used: bsp2cosmic.py

Problem Description: The following use case demonstrates the use of the Monte-Copernicus interface (Trajectory Reverse Engineering strategy) for two orbiter trajectories around Enceladus: an NRHO and an Enceladus Low Orbit (ELO). Enceladus is a celestial body known for its complex gravitational environment, and designing trajectories under this environment is particularly challenging. The initial trajectories are generated using the Circular Restricted Three Body Problem (CR3BP) model in Copernicus. The Copernicus solutions serves as initial condition for reconstruction in Monte under a higher fidelity dynamical model. For details refer to Trajectory Reverse Engineering paper†.

Procedure: The initial trajectories are generated in Copernicus under the CRTBP model, and then saved into an SPK kernel. The main script that performs the conversion from SPK kernel to monte is bsp2cosmic.py A user customized version that execute all the steps described above, applied to each of the subcases are called bsp2cosmic\_ENC1.py, bsp2cosmic\_ENC2.py, and are included at ’monteCop/scripts/dev/’. In general, the script loads the SPK kernel into Monte, perform a scan over the contained trajectory, identify strategic places to place control points, and create a Cosmic timeline. The Cosmic file is the run then used as an initial condition for an optimization procedure, where discontinuities, due dynamical and numerical model differences between Copernicus and Monte are removed. The solutions considered in this test case are originally ballistic, but maneuvers along the trajectory are placed in order to help with the optimization procedure. For Use Case3.4.1 Enceladus Low Orbit, the trajectory is discretized in an even *N* number of segments, then, the multiple-shooting algorithm is applied to recover the CR3BP solution into the Monte high fidelity model. For case use Case3.4.2 NRHO, The SPK kernel is scanned to identify the osculating apsis of the orbit. Control points and maneuvers are placed at all the osculating apoapsis in order to help with the orbit reconstruction. At the periapsis of the orbit, located near the south pole, control points defined with classical orbital elements are added, and bounds to set up a minimum periapsis altitude rp\_min ≥ 3 km are included. The periapsis bounds are included as inequality constraints

A step-by-step description summarizing the trajectory transcription process, for the NRHO case, is as follow:



Scripts to be used: bsp2cosmic\_ENC1.py, bsp2cosmic\_ENC2.py

The scripts are found at ‘monteCop/scripts/dev/’

**Files Required:**

* MONTE-Copernicus interface scripts: (contained in the python module, see above)
* Initial Copernicus trajectories:
  + Enceladus\_NRHO.ideck
  + Enceladus\_Low\_Orbit.ideck
  + Enceladus\_NRHO.bsp
  + Enceladus\_Low\_Orbit.bsp

**Script execution:**

>> bsp2cosmic\_ENF1.py Enceladus\_Low\_Orbit.bsp

>> bsp2cosmic\_ENF2.py Enceladus\_NRHO.bsp

**For help on user inputs:**

>> bsp2cosmic\_ENF.py -h

**Output file:** Enceladus\_Low\_Orbit\_B2M.py, Enceladus\_NRHO\_BSP\_\_B2M.py, Enceladus\_ELO\_OPT.py, Enceladus\_NRHO\_OPT.py

**Optimization steps to finalize Recovery in Hifi (Monte):**

>> cosmic.py -i Enceladus\_NRHO\_B2M.py

>> run()

**Once converged, Save solutions:**

BOA file:

>> saveSol(‘myNewSolution.boa’)

SPK: Save solution as SPICE kernel

>> saveSpk(‘myNewSolution.bsp’)

New Cosmic Timeline: Save converged solution as a new Cosmic timeline

>> saveChkPt(‘myNewSolution.py’)

Find Figure associated to the Monte solutions and the ∆V scan process are located on “Plots” folder.

† “Trajectory Reverse Engineering: A General Strategy For Transferring Trajectories Between

Flight Mechanics Tools” AAS 23-312. 33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, Texas, January 15-19 2023. Ricardo L. Restrepo

\*\*Note: A detailed description of a Use Case, using a Windows machine while accessing the MONTE toolkit through a Docker Container, can be found at: *‘/monteCop/doc/NESC\_tool\_test\_summary.docx’*