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

#

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

#

# Monte-Copernicus Interface: Use Case 3.3, Icy Moon tours

#

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

#

=============================================================================

Use Case 3.4: Icy Moons Tour

monteCop Script to be used: bsp2cosmic\_ENC.py,

Problem Description: This use case represents the SPK to Monte/Cosmic data transfer scenario where trajectories developed in Copernicus or any other astrodynamics capable to store trajectories in form of SPK kernels, are recreated in Monte. The use case considered is the design of Moon tours to explore the Icy worlds. In particular, a Saturn Moon tour that enable an Enceladus capture orbit is used. A traditional tour consists a sequence of flybys of the moons Titan, Rhea, Dione, Tethys, and Enceladus to pump down the energy of the initial Saturn post-capture orbit, enabling an affordable ∆V Enceladus capture insertion. For simplicity, in the consider example, only the Enceladus phase, the last part of the tour, es considered, although the process of generating other phases of the tour is similar. For details refer to Trajectory Reverse Engineering paper†.

**Procedure:**

A low fidelity trajectory was first designed in a tool called Dyno, from the Jet Propulsion Laboratory. Dyno is a low fidelity moon tour design tool, and was used to produce initial conditions in a Conic model. The solution, was then save as an SPK kernel (Enceladus\_tour\_2048.bsp). The Enceladus leg of the tour is composed by the resonance set [7:6, 15:13, 8:7, 17:15, 9:8, 10:9, 11:10], where M and N represent the number of revolutions of Enceladus and the spacecraft around Saturn, respectively. The entire Enceladus phase, which includes seven Enceladus flybys (E1, E2, ..., E7) ending with an Enceladus orbit insertion interface (E8) and spans 105.5 days over 68 revolutions around Saturn. The reconstruction of the Enceladus tour in Monte starts by loading and scanning the SPK kernel. The first step after loading the kernel is to read initial and final states of the trajectory. Two cases were run, one leaving the bounding states fixed, and a second leaving the final state fixed but free to move in time. Then, any close approaches around Enceladus are detected, where control points are placed in order to have control of the flybys. The state of the close approaches is represented as hyperbolic orbital elements and bounds to the minimum altitude of the flybys are added. The SPK kernel is then scanned in search of impulsive burns in the form of velocity discontinuities. Nine burns were found and added to the reconstruction process. 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 called bsp2cosmic\_ENC.py is included at ’monteCop/scripts/dev/’.

**Files Required:**

* MONTE-Copernicus interface scripts (contained in the python module)
* Moon tour Initial conditions trajectory: Enceladus\_tour\_2048.bsp

**Script execution:**

>> bsp2cosmic\_ENF.py Enceladus\_tour\_2048.bsp -id -303 -to 1 -tl 30 -dt 10

**For help on user inputs:**

>> bsp2cosmic\_ENF.py -h

**Output file:** Enceladus\_tour\_2048\_B2M.py

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

>> cosmic.py -I Enceladus\_tour\_2048\_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: *‘/monCop/doc/NESC\_tool\_test\_summary.docx’*