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

#

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

#

# Monte-Copernicus Interface: Use Case 3.5, Europa Flybys

#

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

#

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

Use Case 3.5: Monte to Copernicus trajectory Transfers. Europa Clipper Flybys

monteCop Script to be used: cosmic2cop.py

This use case represents a direct trajectory transfer strategy from MONTE to Copernicus, using as a test case a Europa Clipper flyby sequence. The script used is cosmic2cop.py. The steps taken for this use case are to 1) Design the trajectory in MONTE in form of a Cosmic Timeline 2) identify all the relevant control points (e.g. maneuvers) and events (e.g. apoapsis/periapsis), 3) convert the trajectory into a Copernicus file. 4) utilize Copernicus’ GUI to adjust the segments and/or constraints/functions, optimizer, etc. To generate an ideck from a Cosmic timeline, a similar process to the one implemented for bsp2cosmic.py is performed, but the multiple-shooting strategy is manually set up on the python script by creating segments that are propagated forward and backward on time while imposing continuity true state constraints. For details refer to Trajectory Reverse Engineering paper†.

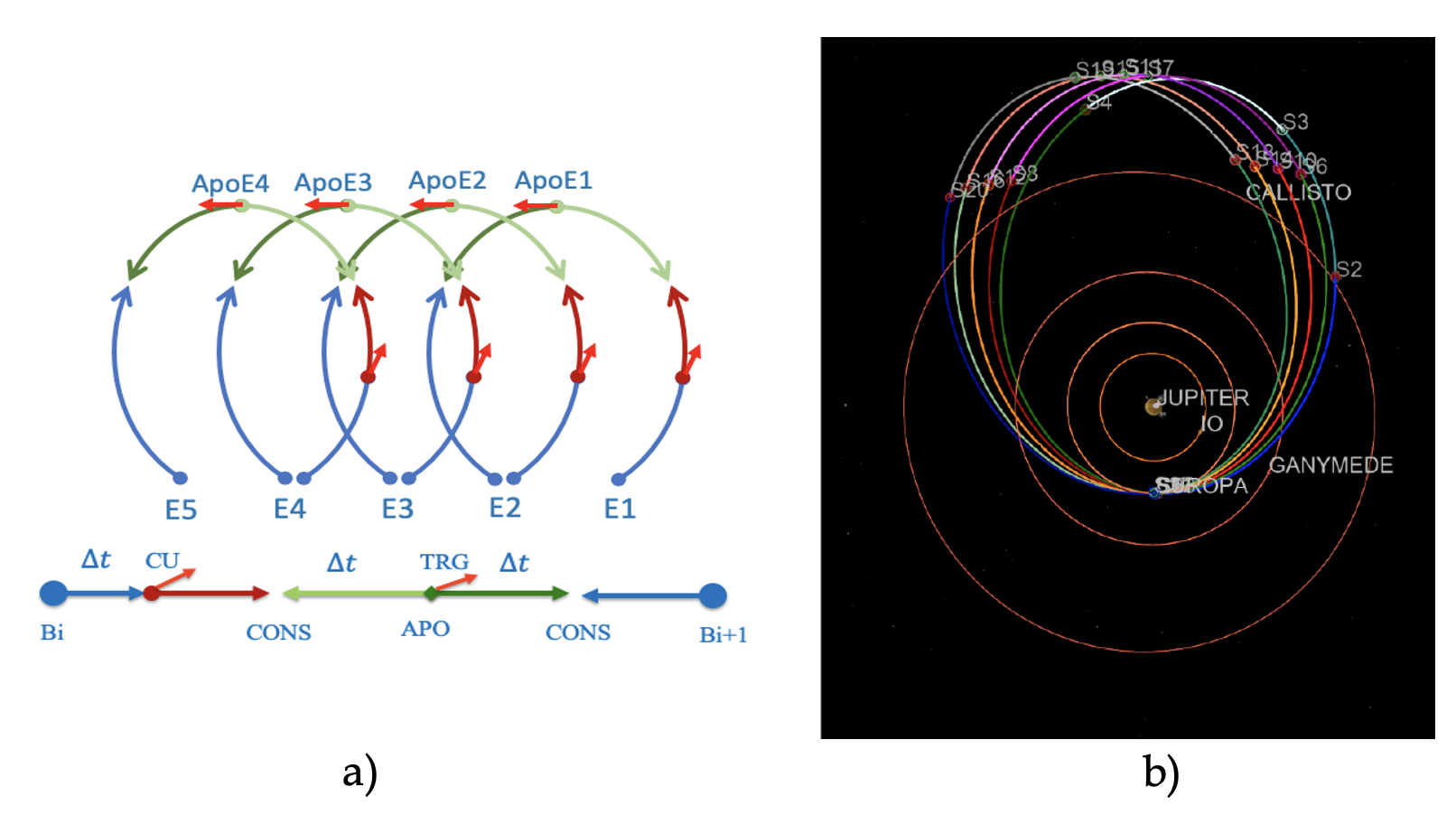


Figure - Europa Clipper Multiple-Flyby architecture. a) Segment structure implemented in Copernicus. b) Converged trajectory

Figure 1 shown in a) is parto of the Europa Clipper trajecotry structure in Cosmic, conform by a set of Control Points ans Break Points, maneuvers and constraints. As an additional note, the transcription from Monte to Copernicus make use of the Copernicus Python interface Robocopy, that allow to build an entire mission from a Python scripting layer.

**Procedure:**

The main script that performs the conversion from a Cosmic timeline into a Copernicus ideck is cosmic2cop.py.

**Files Required:**

* MONTE-Copernicus interface scripts (contained in the python module)
* Moon tour Initial conditions trajectory: euclip\_COT1.py

**Script execution:**

>> cosmic2cop.py euclip\_COT1.py

**For help on user inputs:**

>> cosmic2cop.py -h

**Output file:** euclip\_COT1\_M2C.ideck

Use Copernicus GUI to set up the problem to your own preferences, and perform an optimization (SNOPT recommended). The optimization procedure will close any discontinuities created by the dynamical and numerical models differences between Monte and Copernicus.

† “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’*