Autonomous Vehicle Simulation (AVS) Laboratory,
University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-planetEphemeris

PLANET EPHEMERIS FROM CLASSICAL ORBIT ELEMENTS

Prepared by	H. Schaub

Status: Released
Scope/Contents
The planetEphemeris module uses classical heliocentric orbit elements to specify a planet's position and velocity vectors. Optionally, the planet's orientation can also be specified through a right ascension angle, a declination angle, and a location sidereal time at epoch. The planet rotation about its third (polar) axis can be specified as well. If the planet attitude information is not complete or missing then an inertially fixed zero planet orientation is modeled. The module is able to receive a stack or vector of classical orbit elements and orientation information to output a series of planet ephemeris messages.

Rev	Change Description	By	Date
1.0	First release	H. Schaub	2019-05-04

Contents

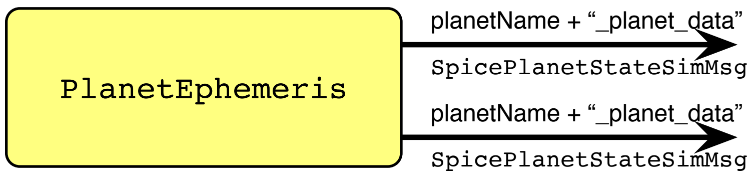

Fig. 1: Illustration of the module input and output messages.

1 Model Description

The purpose of this module is to create planetary ephemeris messages where the planetary motion is modeled through classical orbit elements. The module output messages are illustrated in Figure 1. Note that there are no input messages. The planetary translational motion, and optionally the rotational motion, are specified through the module parameters directly.

1.1 Specifying the required planetary translational information

The first step is to specify the planet names that are to be modeled. This is done by creating a C++ vector of strings and setting them using the module `setPlanetNames()` method. The length of this vector determines how many output messages are created in the vector `planetOutMsgs`. The planet name string provided is used to fill the output message `PlanetName` variable.

Next the translational planetary motion is specified through a C++ vector of classical orbit element structures called `planetElements`. The true anomaly provided is assumes to be the anomaly angle at epoch. This is the same time as the zero'th simulation time step.

1.2 Specifying the optional planetary orientational information

The planet orientation information is optional. However, if it is specified for one planet it must be specified for all planets. The planet polar rotation axis $\hat{e}$, i.e. the 3rd axis of the planet fixed frame, is specified through the right ascension angle RAN and declination angle DEC using

$$ {}^{\mathcal{N}}\hat{\boldsymbol{e}} = {}^{\mathcal{N}}\begin{bmatrix} \cos(\text{RAN})\cos(\text{DEC}) \\ \sin(\text{RAN})\cos(\text{DEC}) \\ \sin(\text{DEC}) \end{bmatrix} \tag{1} $$

The initial angular rotation about this axis is given by the local sidereal time (LST) angle γ_0 at the epoch time. The LST at the current time is determined using the planet's polar rotation rate $\omega_{P/N}$:

$$ \gamma(t) = \gamma_0 + \omega_{P/N}(t - t_{\text{epoch}}) \tag{2} $$

From these states the planet's DCM $[PN]$ and DCM rate $[\dot{PN}]$ are evaluated.

If the planet orientation information is computed, then the output message `computeOrient` is set to $+1$. If not, then `computeOrient` is 0. If the orientation is not specified, then the planet DCM is set to the identity matrix with $[PN] = [I_{3\times3}]$. The DCM rate is set to zero. If any orientation states are not specified, then the module will output this default zero orientation attitude.

2 Module Functions

- **Compute planet heliocentric position and velocity vectors**: Classical heliocentric orbit elements are used to determine the inertial position and velocity vectors
- **Optionally model a constant polar axis rotation**: The module can model a constant inertial rotation of the planet and evaluate the corresponding DCM and DCM rate information.

3 Module Assumptions and Limitations

The module assumes the heliocentric motion is unperturbed. The optional rotational motion is assumed to be an inertially fixed rotation of the planet.

4 Test Description and Success Criteria

The unit test configures the module to model general orbital motions of Earth and Venus. In each case the translational motion is compute for 3 times steps from 0 to 1 second using a 0.5 second time step. The planet orientation information is only set if the appropriate simulation parameter is set. The following sub-sections discuss these flags. If none of these flags are set, then the module default orientation behavior is expected. If all the flags are set, then a constant rotation is set. If only a partial set of orientation information is provided then the reset routine will force the module to throw an error message and only output a default constant zero orientation of the planet.

4.1 `setRAN`

This flag specifies if a set of right ascension angles are specified in the unit test.

4.2 `setDEC`

This flag specifies if a set of declination angles are specified in the unit test.

4.3 `setLST`

This flag specifies if a set of local sidereal time angles are specified in the unit test.

4.4 `setRate`

This flag specifies if a set of planetary polar rotation rates are specified in the unit test.

5 Test Parameters

The unit test verify that the module output guidance message vectors match expected values.

Table 2: Error tolerance for each test.

Output Value Tested	Tolerated Error
`J2000Current`	0.001 s
`PositionVector`	0.001 m
`VelocityVector`	0.001 m/s
`J20002Pfix`	0.001
`J20002Pfix_dot`	10^{-10} rad/s
`computeOrient`	0.001

6 Test Results

All orientation flag permutations are tested and are expected to pass.

Table 3: Test results

`setRAN`	`setDEC`	`setLST`	`setRate`	**Pass/Fail**
True	True	True	True	PASSED
True	True	True	False	PASSED
True	True	False	True	PASSED
True	True	False	False	PASSED
True	False	True	True	PASSED
True	False	True	False	PASSED
True	False	False	True	PASSED
True	False	False	False	PASSED
False	True	True	True	PASSED
False	True	True	False	PASSED
False	True	False	True	PASSED
False	True	False	False	PASSED
False	False	True	True	PASSED
False	False	True	False	PASSED
False	False	False	True	PASSED
False	False	False	False	PASSED

7 User Guide

7.1 Required Module Parameters

- `setPlanetNames(names)` – provide the module a list of strings containing the planet names
- `planetElements` – C++ vector of `classicElements` structures containing the planet ephemeris information

7.2 Optional Module Parameters

The planet orientation information is optional. If not specified, then the planet output message default the planet orientation to an identity matrix with a zero DCM rate matrix. If any of the following orientation parameters are missing, then the module will throw an error message and output the default zero orientation.

- `rightAscension` – C++ vector of doubles containing the planet rotation axis right ascension angles in radians

- `declination` – C++ vector of doubles containing the planet rotation axis declination angles in radian
- `lst0` – C++ vector of doubles containing the planet epoch local sidereal time angles in radians
- `rotRate` – C++ vector of doubles containing the planet polar axis rotation rate in rad/sec