
# Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

## Basilisk Technical Memorandum

## EPHEMERIS_CONVERTER.PY

| Rev | Change Description | By | Date |
|---|---|---|---|
| 1.0 | Initial Revision | T. Teil | 07/2017 |
| 1.1 | Valdiation Revision | T. Teil | 07/2017 |

# Contents



---

# 1 Model Description

## 1.1 Introduction

The ephemeris converter module has the purpose of copying the spice sim messages into a flight software interface message. This allow flight software module to access planet ephemeris data.

## 1.2 I/O

The spice message, which is read in by the module, contains the following variables:

- J2000Current: the time of validity for the planet state
- PositionVector : the true position of the planet for the time
- VelocityVector[3]: the tue velocity of the planet for the time
- J20002Pfix: the orientation matrix of planet-fixed relative to inertial
- J20002Pfix_dot: the derivative of the orientation matrix of planet-fixed relative to inertial
- computeOrient: a flag indicating whether the reference should be computed
- PlanetName

Only the position and velocity vectors are transferred to the output ephemeris message, which therefore contains:

- r_BdyZero_N[3]: the position of orbital body in the inertial frame
- v_BdyZero_N[3]: the velocity of orbital body in the inertial frame
- timeTag: the vehicle Time-tag for state

# 2 Model Functions

This module allows the creation of a map function that maps the input message to the output message in which the content will be copied. The use of this can be seen in the user guide.

# 3 Model Assumptions and Limitations

There are no direct assumptions and limitations in this module which is simply a necessary piece for the code architecture.

# 4 Test Description and Success Criteria

This test sets up an appropriate simulation by creating a Spice Object, which will write messages containing ephemerides. A ephemeris converter object is also created with the map between the message names. This test guarantees that the data is properly copied.

This test is located in:
`simulation/environment/ephemerisConverter/_UnitTest/test_ephemerisconvert.py`.

## 4.1 Validation success criteria

The criteria for a successful testing of this module is driven by the correct copy of the spice messages. This is done simply by comparing the messages before the copy with the outcome of the copied message. The error tolerance is at $\epsilon = 10^{-5}$ which corresponds to 12 significant digits. This gives a healthy margin from machine precision all the while getting all of the physical information from the ephemerides.

# 5 Test Parameters

A spice object was created in order to write the messages that need to be copied. The spice object was set on the following date: 2015 February 10, 00:00:00.0 TDB, and the planets that were loaded where the Earth, Mars Barycenter, and the Sun.

### Successful link test

A boolean variable is added for logging and is verified to have successfully linked the desired messages. This is done by comparing the variables down to $\epsilon = 10^{-12}$, since we are looking for a value of 1.

### Successful copy test

For each of the celestial bodies who get a message output, we log the two messages that contain their position and velocities in the inertial frames. For mars the first message is the Spice message is `mars_planet_data` and the second message is the ephemeris converted data `mars_ephemeris_data`. If the norm of their relative difference (including the time component of the vector), at any time, is greater than our error tolerance $\epsilon = 10^{-12}$, then the test fails.

# 6 Test Results

## 6.1 Pass/Fail results

| Test | Link Test | Copy Test |
|---|---|---|
| Result | Passed | Passed |
| Tolerance | $10^{-12}$ | $10^{-12}$ |

Both components of the test pass. The copy is therefore a properly executed.

# 7 User Guide

The nominal set-up for the ephemeris converter is done as follows:

- `EphemObject = ephemeris_converter.EphemerisConverter()`
- `EphemObject.ModelTag = 'EphemData'`
- `EphemObject.addSpiceInputMsg(spiceInMsg)`

The output messages are then stored in the vector of output messages called `ephemOutMsgs`.