Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum
Document ID: Basilisk-oeStateEphem

ORBIT ELEMENT STATE EPHEMERIS GENERATION MODULE

Prepared by	H. Schaub

Status: Released
Scope/Contents
This module takes the TDB time, current object time and computes trajectory position and velocity states using the time corrected by TDB and the stored Chebyshev orbit element coefficients. If the time provided is outside the specified range for which the stored Chebyshev coefficients are valid then the position vectors rail high/low appropriately.

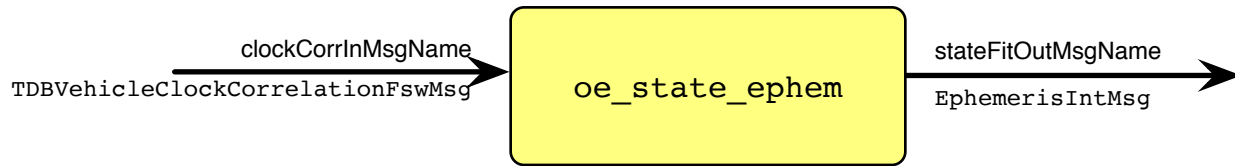| Rev	Change Description	By	Date
1.0	First Documentation Release	H. Schaub	2019-04-26

Contents

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

1   Model Description

The purpose of this module is to model a trajectory in space using a fitted set of Chebyshev coefficients. As illustrated in Figure 1, there is a single input message which provides the ephemeris time associated with the vehicle on the trajectory, as well as vehicle time code information. This is used to compute a corrected mission time used to compute the location on a given trajectory.

The output of the module is an ephemeris interface message which contains the inertial position vector $^{\mathcal{N}}\boldsymbol{r}$ and inertial velocity $^{\mathcal{N}}\dot{\boldsymbol{r}}$. To model the trajectory a Chebyshev polynomial approximation is used of the orbit elements. The elements modeled include the radius at periapses r_p, the eccentricity e, the inclination angle i, the ascending node angle Ω, the argument of periapses angle ω and the true anomaly angle f. The elements were chosen as they are defined finite quantities for ellipses, parabolas and hyperbolas.

The current time is used to first determine which Chebyshev coefficient record to use. Each polynomial fit is good for a discrete period of time determined through the interval mid-point $t_{m,i}$ and radius $t_{r,i}$. The algorithm determines which record interval mid-point $t_{m,i}$ is closest to the current simulation time t_i.

The next step computes a scaled time t^* where the time relative to $t_{m,i}$ is normalized that that it yields a value between [-1,1] where $+1$ would be the upper bounder at $t_i = t_{m,i} + t_{r,i}$. If there is a cap in the time intervals the module is set to fail high or low relative to the nearest time interval.

After selecting the proper set of Chebyshev coefficients for the current time interval the corresponding orbit elements are retrieved. The `elem2rv()` function of the `orbitalMotion.c/h` support library is

used to map these into the desired inertial position and velocity vectors. This requires mapping the radius of perapses into a semi-major axis a measure using

$$a = \frac{r_p}{1 - e} \tag{1}$$

If $e \approx 1$ then `elem2rv()` assumes $a = 0$.

2   Module Functions

- **Evluate a correct mission time**:

- **Compute trajectory position and velocity vectors**: Use stored Chebyshev polynomial coefficients to model the spacecraft trajectory.

3   Module Assumptions and Limitations

The model assumes that all orbit elements are modeled using the same order Chebyshev polynomial.

4   Test Description and Success Criteria

4.1   Check 1: time measure outside of Chebyshev time interval

The unit test makes sure that the correct response is obtained if a time step is used that is outside of the Chebyshev time intervals. This tests the expected failing high and low feature. The test is expected to pass.

4.2   Check 2: TDRSS Data Fit

The Spice trajectory of the TDRSS satellite is loaded. The trajectory is mapped into orbit elements which are then converted to Chebyshev coefficients. The simulation is run for 4 days and the compute trajectory is then compared to the actual Spice TDRSS trajectory. Both the inertial position and velocity vectors are checked.

5   Test Parameters

The Spice TDRSS trajectory data is loaded as the truth trajectory. A 14th order Chebyshev model is then created and uploaded to the BSK module. The unit tests verify that the module output guidance message vectors match expected values.

Table 2: Error tolerance for each test.

Output Value Tested	Tolerated Error
`r_BdyZero_N`	10000.0 m
`v_BdyZero_N`	1.0 m/s

6   Test Results

All of the tests passed:

Table 3: Test results

Check	Pass/Fail
1	PASSED
2	PASSED

7 User Guide

The following items must be set to operate this module:

- `muCentral`: set the central body gravity constant

- `ephArray`: Set the array of orbit element Chebyshef coefficients

- `stateFitOutMsgName`: module output message name

- `clockCorrInMsgName`: module input message name