Test Cases for frames.py

Author: Sanket Chirame
Date: 9/5/2018

1. **Test cases for latlon:**
    a. Test_cases_file: Take data from file “test-data/test_latlon.json”. Check the latitude and longitude for extreme cases of ecef vectors. Also verify that the returned data type is a tuple.
        i. North pole
        ii. South pole
        iii. Intersection of equator and 0 deg longitude
        iv. Intersection of equator and 180 deg longitude
        v. Intersection of equator and 90 deg longitude
        vi. Intersection of equator and and -90 deg longitude
    b. Test_z_symmetry_lat: Set ecef vector as [sin(k) , cos(k), 13 .0] Change k from 1 to 6. Latitude should remain same for all cases
    c. Test_independnt_of_radius: Use different value of radius with same direction cosines. Both latitude and longitude should remain unchanged.
2. **Test Cases for Ecif2Ecef:**
    a. Test_orthogonality_of_matrix: calculate transformation of x,y,z axis and construct matrix using it. Do this for different time values. Check if matrix is orthogonal.
    b. Test_at_zero_time: Set time such that (time from launch + time between equinox to launch) is zero. This achieves the situation when ecif and ecef are aligned. Thus transformed vector is identical to initial.
    c. Test_periodicity: Change time by 2*pi/w_earth. The transformed vector should be same for both t = t0 and t = t+ 2pi/w
    d. Test_cases_from_file_e2e: Randomly chosen vectors are given as input. The transformed vector is explicitly calculated using formula.
3. **Test cases for Ecef2Ecif:**
    a. Test_cases_from_file_e2e: Randomly chosen vectors are given as input. The transformed vector is explicitly calculated using formula.
    b. Test_inverse: Verify if ecef2ecif(ecif2ecef( )) gives back the same vector.
4. **TestEcif2Orbit:**
    a. Test_e2o_matrix_orthogonality: calculate transformation of x,y,z axis and construct matrix using it. Do this for different position, velocity values. Check if matrix is orthogonal
    b. Test_check_pos_vel_transformatio: Verify if position and velocity values are transformed according to definition.
        i. Z_orb = - unit(r)
        ii. Y_orb = unit(cross(v , r))
        iii. X_orb = cross(Y_orb , Z_orb)

5. Test_qBI_2qBO:
    a. Test_south_pole_ideal: Satellite is above south pole. Body, orbit and eci all aligned wrt each other. qBI = qIO = qBO = [1,0,0,0]
    b. Test_data: Test explicit data. Example3.7 from Junkins page 89. qIO = beta', beta'' = qBI , beta = qBO
        i. $$\boldsymbol{\beta}' = \left(0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right) \quad \boldsymbol{\beta} = \frac{1}{2\sqrt{2}}\left(\sqrt{3}, \sqrt{3}, 1, 1\right)$$
        $$\boldsymbol{\beta}'' = \left(\frac{1}{2}\sqrt{\frac{\sqrt{3}}{2}+1}, -\frac{1}{2}\sqrt{\frac{\sqrt{3}}{2}+1}, \frac{-\sqrt{2}}{4\sqrt{2+\sqrt{3}}}, \frac{\sqrt{2}}{4\sqrt{2+\sqrt{3}}}\right)$$

6. Test_wBIB_2_wBOB:
    a. Test_ideal_controlled: Controlled satellite where orbit frame is aligned with body.
    b. Test_stationary_body: Body is stationary wrt eci frame. Then wBO is wIO. Data file is test-data folder.