C2C Shankar Kulumani

PREDICT Algorithm

Given: Satellite TLEs and Observation site data

Find: Range, azimuth, elevation of visible satellites at 2min intervals during the viewing window

Process:

1. Open constants – WGS84Data
2. Open input and output files
3. Input observation site information
   1. Site latitude – degrees
   2. Site longitude – degrees
   3. Site altitude – kilometers
   4. Viewing Start Time – Julian Day
   5. Viewing Stop Time – Julian Day
4. Convert observation site information into working units
   1. Convert all units in degrees to radians
5. Create Satellite Loop to process multiple satellite TLEs- continue to process until end of input file
   1. Read satellite TLE information from input file
      1. Input input file id and output file id to input\_predict.m
         1. Read the following information directly from the TLE input file
            1. Satellite name – string
            2. Satellite number – integer
            3. Half Mean motion rate – rev/day2
            4. Initial inclination – degrees
            5. Initial right ascension of the ascending node – degrees
            6. Initial eccentricity – unit less
            7. Initial argument of periapsis – degrees
            8. Initial mean anomaly – degrees
            9. Initial mean motion – rev/day
         2. Convert to working units
            1. Convert all units in degrees to radians

* + - * 1. Convert rev/day to radians/day
      1. Convert the epoch year and day fraction into Julian Day units
         1. TLE only supplies last two digits of year

07 =2007

06 = 2006

* + - * 1. Covert full year and day fraction into Julian Date using JulianDay.m
      1. Input initial inclination, initial eccentricity, initial mean motion, half mean motion rate into J2DragPert.m
         1. Declare global variables J2 and Radius of earth
         2. Calculate nodal rate



* + - * 1. Calculate argument of periapsis rate



* + - * 1. Calculate eccentricity rate



1. Create a Time loop to process each satellite’s location during the viewing window
   1. Input elapsed time since epoch, initial mean motion, half mean motion rate, initial eccentricity, eccentricity rate, initial right ascension of ascending node, right ascension of ascending node rate, initial argument of periapsis, argument of periapsis rate, and initial mean anomaly into Update.m to find new Orbital elements
      1. Calculate mean motion at the elapsed time
         1. 
      2. Calculate eccentricity at the elapsed time
         1. 
      3. Calculate RAAN at the elapsed time
         1. 
      4. Calculate argument of periapsis at the elapsed time
         1. 
      5. Calculate true anomaly at the elapsed time
         1. Convert initial true anomaly to initial Eccentric Anomaly
            1. 
         2. Convert initial eccentric anomaly to initial mean anomaly
            1. 
         3. Convert initial mean anomaly to future mean anomaly
            1. i
         4. Input mean anomaly and eccentricity into Newton.m
            1. Calculate future eccentric anomaly



* + - * 1. Continue to perform iterative calculations until the difference between successive answers is less than 1\*10-9
      1. Convert future eccentric anomaly to future true anomaly at the elapsed time
         1. 
  1. Input mean motion, eccentricity , inclination, right ascension of the ascending node, argument of periapsis, and true anomaly into RVector.m
     1. Calculate Satellite position Vector – kilometers
        1. Define global variable MU
        2. Convert mean motion to semi major axis
           1. ****
        3. Calculate the parameter
           1. 
        4. Calculate the position vector in PQW system
           1. 
        5. Rotate the position vector from PQW to IJK
           1. Multiply PQW vector by ROT3 then ROT1 then ROT3
           2. where = -argument of periapsis
           3. where  = -inclination
           4. where = -RAAN
           5. Rijk=ROT3\*ROT1\*ROT3\*Rpqw
  2. Find GST and LST
     1. Input Julian Day and site longitude into GSTLST.m
        1. Declare global SidePerSol from WGS84Data.m
        2. Unpack Julian Day back into Year, Month, Day, Hour, Minute, Second by inputting Julian Day into InvJulianDay.m
        3. Find GST0 by inputting Year into GSTim0
        4. Find the fractional day by inputting Year, Month, Day, Hour, Minute, and Second into FindDays.m
        5. Calculate GST
        6. Calculate LST
        7. Perform a rev check to ensure the GST and LST are between 0 and 2π
  3. Calculate the site vector of observation site in IJK
     1. Input LST, Site latitude(L), site altitude(H) into site.m
  4. Input site vector, satellite vector, site latitude, and LST into RhoAzEl.m
     1. Calculate Range vector from observation site to satellite
        1. =Rsat-Rijk

Convert range vector from IJK to SEZ system

Multiply IJK by ROT2 then ROT3

1. where β = -(π/2-site latitude)
2. where = -LST
   * 1. Calculate Azimuth, Elevation , and Range using geometry
        1. Range=Magnitude of range vector
        2. Azimuth =
        3. Elevation =
   1. Test for visibility – if test fails do not output information and continue to step 7
      1. Test if observation site is in darkness
         1. Input Julian day into Sun.m
         2. Use vecangle.m to determine angle between sun vector and site vector
            1. Angle must be greater than 90° to be visible
         3. If it passes continue to next test
      2. Test to see if range<1500 km and Elevation >10°
         1. If it passes continue to next test
      3. Test if satellite is in the sunlight
         1. Use vecangle.m to determine angle(β) between sun vector and satellite vector
         2. If the following is true continue
            1. OR β >90°
   2. If satellite passes all tests output the information
      1. Input range, azimuth, elevation, observer in darkness flag, visibility of satellite flag, Julian date, and output file id into output\_predict.m
         1. Convert radians back to degrees
            1. Same equation as above
3. Continue time loop until reaching end of viewing window
4. Continue satellite loop until end of input file
5. Close input and output files