OrbitComp

An Orbit determination software

Lee Weidow

Abstract

The objective of this project was to create a software for preliminary orbit determination from ground station data. Its inputs are the ground station measurements and desired time of flight. The outputs are the initial and final orbital elements (OEs), position vectors, and velocity vectors. Using the sample data given the program calculated accurate values for these. Two cases were considered in the sample data. For case 1, the time of flight (TOF) was 32 minutes and for case 2 it was 10 hours. Earth's J2 is also accounted for in the program.

Introduction

The software, called OrbitComp from, takes the ground station measurements and the desired time of flight (TOF), and outputs the classical orbital elements and the initial and final velocity and position vectors of the object. The purpose of this program is to find where the satellite is at any point of time. The measurements taken are the local sideral time (λ), longitude (φ), range (ρ), azimuth (β), elevation (σ), range rate ($\dot{\rho}$), azimuth rate ($\dot{\beta}$), and elevation rate ($\dot{\sigma}$). The orbital elements are the semi-major axis (a), true anomaly (f), eccentricity (e), inclination (i), argument of the periapsis (ω) and longitude of the ascending node (Ω). Effects of Earths J2 are also included in the software for ω and Ω. These rates are called $\dot{\omega}$ and $\dot{\Omega}$, but they are not included in the outputs. To test OrbitComp, data from an elliptical satellite was used. The test data outputs are accurate to hand calculations.

Theoretical Development

This program is a combination of preliminary orbit determination and orbital element determination around Earth therefore, $\mu=398600\frac{km^3}{s^2}$. Because of this OrbitComp, and its development process, is split into three main parts. The first part takes the inputs and calculates the initial velocity and position vectors. The equations

$$\boldsymbol{\rho}_{sez} = -\rho\cos\sigma\cos\beta\hat{\boldsymbol{S}} + \rho\cos\sigma\sin\beta\hat{\boldsymbol{E}} + \rho\sin\sigma\hat{\boldsymbol{Z}} \text{ (1)}$$

and

$$\dot{\boldsymbol{\rho}}_{sez} = \left(-\dot{\rho}\cos\sigma\cos\beta + \rho\dot{\sigma}\sin\sigma\cos\beta + \rho\dot{\beta}\cos\sigma\sin\beta\right)\hat{\boldsymbol{S}} + \left(\rho\cos\sigma\sin\beta - \rho\dot{\sigma}\sin\sigma\sin\beta + \rho\dot{\beta}\cos\sigma\cos\beta\right)\hat{\boldsymbol{E}} + (\rho\sin\sigma + \rho\dot{\sigma}\cos\sigma)\hat{\boldsymbol{Z}} \text{ (1)}$$

are used first to determine the range and range rate vectors in the SEZ coordinate frame. It is assumed that the ground station is at sea level so $\mathbf{r}_{\mathbf{siteSEZ}}=6378\hat{\boldsymbol{Z}}$ km. Also, the rotational velocity of Earth in the ECI coordinate frame is $\boldsymbol{\omega}_E=7.2921*10^{-5}\,\hat{K}$ km/s. To convert the SEZ values to ECI a rotational matrix is used. To find the ECI values, the SEZ value is multiplied by

$$[ROT] = \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 \\ \sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin\varphi & 0 & \cos\varphi \\ 0 & 1 & 0 \\ -\cos\varphi & 0 & \sin\varphi \end{bmatrix} \text{ (1)}$$

For example

$$\rho_{ECI} = [ROT] * \rho_{SEZ} \text{ (1)}$$

and then the initial velocity and position vectors, in the ECI frame, are found with

$$\boldsymbol{r_i} = \boldsymbol{r_{siteECI}} + \boldsymbol{\rho}_{ECI} \text{ (1)}$$

$$\boldsymbol{v}_i = \dot{\boldsymbol{\rho}}_{ECI} + \boldsymbol{\omega}_E \times \boldsymbol{\rho}_{ECI} + \boldsymbol{\omega}_E \times \boldsymbol{r_{siteECI}} \text{ (1)}$$

The second part uses these two vectors to find the 6 OEs. These equations are:

$$r = \|\boldsymbol{r}\|$$
$$\hat{\boldsymbol{r}} = \boldsymbol{r}/r$$

$$\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v} \qquad \boldsymbol{e} = \frac{1}{\mu}\boldsymbol{v} \times \boldsymbol{h} - \hat{\boldsymbol{r}}$$
$$h = \|\boldsymbol{h}\| \qquad e = \|\boldsymbol{e}\|$$
$$\hat{\boldsymbol{h}} = \boldsymbol{h}/h \qquad \hat{\boldsymbol{e}} = \boldsymbol{e}/e$$

$$\hat{\boldsymbol{n}} = (\hat{\mathbf{K}} \times \boldsymbol{h}) \,/\, \|\hat{\mathbf{K}} \times \boldsymbol{h}\|$$

$$\varepsilon = \frac{1}{2}\boldsymbol{v} \cdot \boldsymbol{v} - \frac{\mu}{r} \qquad a = -\frac{\mu}{2\varepsilon}$$

$$i = \cos^{-1}(\hat{\boldsymbol{h}} \cdot \hat{\mathbf{K}})$$
$$\Omega = \tan^{-1}\left(\frac{\hat{n}_Y}{\hat{n}_X}\right)$$

$$\omega = \cos^{-1}(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{e}}) \qquad (\text{if} \quad \hat{\boldsymbol{e}} \cdot \hat{\mathbf{K}} \geq 0)$$
$$\omega = 2\pi - \cos^{-1}(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{e}}) \qquad (\text{if} \quad \hat{\boldsymbol{e}} \cdot \hat{\mathbf{K}} < 0)$$

$$f = \cos^{-1}(\hat{\boldsymbol{e}} \cdot \hat{\boldsymbol{r}}) \qquad (\text{if} \quad \boldsymbol{r} \cdot \boldsymbol{v} \geq 0)$$
$$f = 2\pi - \cos^{-1}(\hat{\boldsymbol{e}} \cdot \hat{\boldsymbol{r}}) \qquad (\text{if} \quad \boldsymbol{r} \cdot \boldsymbol{v} < 0)$$

(2)

used in order from top to bottom for simplicity. For equations based on conditions, both are accounted for in the program. The Third part of the software uses Newton,s method and J2 perturbation equations to determine the final OEs, position vector, and velocity vectors. The definition of Newton's method, taken from the lecture 5 slides is

Newton's method is the classical approach for solving Kepler's equation (finding a root of the function):

- Choose $E_0 = M$ as the initial guess.
- Iterate successive values

$$E_{i+1} = E_i - \frac{E_i - e\sin E_i - M}{1 - e\cos E_i},$$

as $i = 0,1,2\cdots$, until the solution no longer changes within the desired precision.

(5)

With

$$M = \sqrt{\frac{\mu}{a^3}} * TOF. \text{ (3)}$$

Newton's method is used until the difference between the two Es is less than .05. Next, f_f is found using:

$$f_2 = 2\tan^{-1}\left[\sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}\right] \text{(3)}$$

At the same time, the perturbations due to Earths J2 are calculated using

$$\dot{\Omega} = -\frac{3nJ_2}{2(1-e^2)^2}\left(\frac{R_E}{a}\right)^2 \cos i$$

$$\dot{\omega} = \frac{3nJ_2}{4(1-e^2)^2}\left(\frac{R_E}{a}\right)^2 (5\cos^2 i - 1) \quad (4)$$

These are then used in the following equations to determine the final values of Ω and ω

$$\Omega(t) = \Omega(t_0) + \dot{\Omega}(t - t_0),$$
$$\omega(t) = \omega(t_0) + \dot{\omega}(t - t_0). \quad (4)$$

Here, $t\text{-}t_0$=TOF, and the dot indicates time rate of change. Also, the t_0 is for initial values Lastly, the final position and velocity vectors are found using the new OEs in the PQW coordinate frame. The equations used are:

$$p = a * (1 - e^2) \text{ (3),}$$

assuming the orbit is elliptical. Otherwise, p=a.

$$h_2 = \sqrt{p * \mu}$$

$$r_2 = \frac{p}{1 + e\cos f_2}$$

$$\boldsymbol{r}_{2pqw} = r_2 \cos f_2\, \hat{\boldsymbol{p}} + r_2 \sin f_2\, \hat{\boldsymbol{q}}$$

$$\boldsymbol{v}_{2pqw} = \frac{\mu}{h}[-\sin f_2\, \hat{\boldsymbol{p}} + (e + \cos f_2)\hat{\boldsymbol{q}}] \text{ (2)}$$

Where p is the focal parameter and h is the angular momentum. While calculation for h_2 is not necessary, doing so makes the code easier to follow. Then, using a new rotational matrix for PQW to ECI like before, the ECI values are found. The PQW to ECI rotational matrix is:

$$[ROT]_{ECI} = \begin{bmatrix} \cos\Omega_f & -\sin\Omega_f & 0 \\ \sin\Omega_f & \cos\Omega_f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \begin{bmatrix} \cos\omega_f & -\sin\omega_f & 0 \\ \sin\omega_f & \cos\omega_f & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ (2)}$$

No further equations are used.

Simulation Results

The OrbitComp program is very simple to use. There are two ways to input values. They can either be hardcoded or the input functions can be used. To hardcode your values in, simply replace the sample hardcoded values and run the program. To use the input functions, remove the percentage signs, %, and put them on the hardcoded values. Run the program and it will prompt you for the necessary values. When inputting values make sure that they are in either degrees, km/s, or km. After running and inputting your values the program will show a screen like these:

```
Position and Velocity Vectors:
r0=4590.93I + -3560.67J + 4102.72K (km)
v0=0.26I + 5.41J + 5.3047K (km/s)
rf=3692.87I + 1285.81J + 7045.40K (km)
vf=-3.15I + 6.39J + 0.7377K (km/s)

Initial OEs:
a=7306.366136 (km)
f=72.783287 (deg)
e=0.073401
i=61.358702 (deg)
w=328.307809 (deg)
omega=-60.482192 (deg)

Final OEs:
a=7306.366136 (km)
f=118.776432 (deg)
e=0.073401
i=61.358702 (deg)
w=328.297461 (deg)
omega=-60.548876 (deg)
```

```
Position and Velocity Vectors:
r0=4590.93I + -3560.67J + 4102.72K (km)
v0=0.26I + 5.41J + 5.3047K (km/s)
rf=4342.31I + -2623.02J + 4728.06K (km)
vf=2.19I + -6.77J + -2.3433K (km/s)

Initial OEs:
a=7306.366136 (km)
f=72.783287 (deg)
e=0.073401
i=61.358702 (deg)
w=328.307809 (deg)
omega=-60.482192 (deg)

Final OEs:
a=7306.366136 (km)
f=-83.127435 (deg)
e=0.073401
i=61.358702 (deg)
w=328.113787 (deg)
omega=-61.732520 (deg)
```

These out are for both case 1 and 2 respectfully. A f denotes final values and 0 is initial.

Conclusions

From the test data, OrbitComp was able to calculate the OEs, position vectors, and velocity vectors. It did this while include effects from Earth's J2. From this we can see that the effect here is small, but not negligible over longer periods of time. This is because the sample data was for a satellite in Low Earth Orbit. These orbits are more affected by Earth's J2.

References

1. Nazari, Morad. *Lecture 9 Preliminary Orbit Determination*, *https://erau.instructure.com/courses/132400/files/27580348?module_item_id=7917927*
2. Nazari, Morad. *Lecture 7 Orbits in 3D*, *https://erau.instructure.com/courses/132400/files/27580349?module_item_id=7917929*
3. Curtis, Howard D.. *Orbital Mechanics for Engineering Students*, Elsevier Science & Technology, 2019. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/erau/detail.action?docID=5787892.
4. Kleuver, Craig. *Space Flight Dynamics*, https://erau.instructure.com/courses/132400/files/27580354?module_item_id=7917938
5. Nazari, Morad. *Lecture 5 Orbital Motion as a Function of Time*, *https://erau.instructure.com/courses/132400/files/27580340?module_item_id=7917931*