BASICS 1: COORDINATE TRANSFORMATIONS

AE4-878 Mission Geometry and Orbit Design

Aaron de Windt — 4134249

Delft University of Technology
Delft

2021

1 Assignment

1. Convert the following state-vector from Cartesian components to Kepler elements:

$$
\begin{aligned}
x &= 8751268.4691\ [m]\\
y &= -7041314.6869\ [m]\\
z &= 4846546.9938\ [m]\\
\dot{x} &= 332.2601039\ [ms^{-1}]\\
\dot{y} &= -2977.0815768\ [ms^{-1}]\\
\dot{z} &= -4869.8462227\ [ms^{-1}]
\end{aligned}
$$

2. Convert the following state-vector from Kepler elements to Cartesian components:

$$
\begin{aligned}
a &= 12158817.9615\ [m]\\
e &= 0.014074320051\ [m]\\
i &= 52.666016957\ [deg]\\
\Omega &= 323.089150643\ [deg]\\
\omega &= 148.382589129\ [deg]\\
M &= 112.192638384\ [deg]
\end{aligned}
$$

2 Mathematical description

The goal of this assignment is to develop the necessary functions to convert between Cartesian components and Kepler elements. This section explains the mathematics used in the implementation of these functions. These where all found in Mission geometry: orbit and constellation design and management by J. R. Wertz [1].

2.1 Cartesian components to Kepler elements

First step is to calculate the orbit angular momentum, **h**, and the vector to the ascending node, **N**.

$$\mathbf{h} = \mathbf{r} \times \mathbf{V} \tag{1}$$

$$\mathbf{N} = \frac{\hat{\mathbf{z}} \times \hat{\mathbf{h}}}{|\mathbf{h}|} \tag{2}$$

Here are **r** and **V** the position and velocity vectors and $\hat{z}$ is a unit vector normal to the equatorial plane.

The eccentricity vector can be calculated using,

$$\mathbf{e} = \frac{\mathbf{V} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{|\mathbf{r}|} \tag{3}$$

The semi-major axis can be found using the total energy.

$$E = \frac{|\mathbf{V}|^2}{2} - \frac{\mu}{|\mathbf{r}|} \tag{4}$$

$$a = -\frac{\mu}{2E} \tag{5}$$

The eccentricity, e, inclination i, right ascension of the ascending node, Ω, argument of periapsis, ω, and true anomaly θ can then be calculated using,

$$e = |\mathbf{e}| \tag{6}$$

$$\cos i = \frac{h_z}{|\mathbf{h}|} \tag{7}$$

$$\tan\Omega = \frac{N_y}{N_x} \tag{8}$$

$$\cos\omega = \frac{\mathbf{e}\cdot\hat{\mathbf{N}}}{e} \tag{9}$$

$$\cos\theta = \frac{\mathbf{r}\cdot\mathbf{e}}{e\,|\mathbf{r}|} \tag{10}$$

The argument of periapsis will be between 0° and 180° when $\left(\hat{\mathbf{N}} \times \mathbf{e}\right)\cdot\mathbf{h} > \mathbf{0}$ and between 180° and 360° when this value is negative. However the expression above will only give values between 0° and 180°. The correct value for the argument of periapsis may be calculated by subtracting the result of Equation 9 from 360° when $\left(\hat{\mathbf{N}} \times \mathbf{e}\right)\cdot\mathbf{h} < 0$.

A similar issue is also present for the true anomaly, it will be between 0° and 180° when $(\mathbf{e}\times\mathbf{r})\cdot\mathbf{h} > 0$, otherwise it will be between 180° and 360°. The solution is the same as with the argument of periapsis.

Finally the eccentric anomaly, E, and mean anomaly M are calculated as using,

$$\tan E = \frac{\sqrt{1-e^2}\sin\theta}{e+\cos\theta} \tag{11}$$

$$M = E - e\sin E \tag{12}$$

2.2 Kepler elements to Cartesian components

Beginning in the perifocal coordinate system in which the orbit plane is the x/y plane with the x axis in the direction of periapsis. The semiparameter, p, is the distance from the center of mass to the orbit, measured perpendicular to the mayor axis. It's given by the following equation in terms of the semi-mayor axis, a, and eccentricity, e.

$$p = a\left(1-e^2\right) \tag{13}$$

The position in the perifocal frame is given by,

$$\mathbf{r_{pf}} = \begin{bmatrix} p\frac{\cos\theta}{1+e\cos\theta} \\ p\frac{\sin\theta}{1+e\cos\theta} \end{bmatrix} \tag{14}$$

and the velocity,

$$\mathbf{V_{pf}} = \begin{bmatrix} -\sqrt{\frac{\mu}{p}}\sin\theta \\ \sqrt{\frac{\mu}{p}}(e+\cos\theta) \end{bmatrix} \tag{15}$$

The transformation matrix, C_{pf}, from the perifocal to inertial frame is as follows,

$$C_{pf} = \begin{bmatrix} -\sin(\Omega)\sin(\omega)\cos(i)+\cos(\Omega)\cos(\omega) & -\sin(\Omega)\cos(\omega)\cos(i)-\sin(\omega)\cos(\Omega) & \sin(\Omega)\sin(i) \\ \sin(\Omega)\cos(\omega)+\sin(\omega)\cos(\Omega)\cos(i) & -\sin(\Omega)\sin(\omega)+\cos(\Omega)\cos(\omega)\cos(i) & -\sin(i)\cos(\Omega) \\ \sin(\omega)\sin(i) & \sin(i)\cos(\omega) & \cos(i) \end{bmatrix} \tag{16}$$

The inertial position and velocity can then be calculate using,

$$\mathbf{r} = C_{pf}\cdot\mathbf{r_{pf}} \tag{17}$$

$$\mathbf{V} = C_{pf}\cdot\mathbf{V_{pf}} \tag{18}$$

The true anomaly is needed for these equations, but it may not necessarily be known. It is possible to calculate the true anomaly from the eccentric anomaly using,

$$\tan\frac{\theta}{2} = \left(\frac{1+e}{1-e}\right)^{\frac{1}{2}}\tan\frac{E}{2} \tag{19}$$

Unfortunately there is no analytical solution for calculating the eccentric or true anomaly from the mean anomaly, thus it necessary to solve it numerically. For this assignment it is done by solving the following equation for the eccentric anomaly, E, using a numerical root finding method. The specific numerical method used is left to be selected by the scientific software library (scipy) based on the inputs given to it.

$$0 = E - e\sin E - M \tag{20}$$

3 Testing

Unittests where written for testing the functions. The first few unittests use the examples given in the lecture slides. The functions are called with the example input values and the results are checked to be close within a certain tolerance to the expected values. Tables 1 and 2 show the results of these tests.

The rest of the unittests convert from Kepler elements to Cartesian components and back to Kepler elements which are then checked against the original Kepler elements. These cases are manually selected in order include edge cases and cases not covered by the examples from the lecture slides. Examples of some of these cases are zero and negative inclination, highly eccentric but still elliptical orbits, argument of periapsis and true anomalies between 180° and 360° and combinations of some of these conditions. Due to the large number of automated tests it is inconvenient to add the results directly in this report, however at the time of writing all tests where passing.

	Example 1			Example 2		
	Result	Expected	Error%	Result	Expected	Error%
a $[m]$	6.78775e+06	6.78775e+06	1.08393e-11	7.09614e+06	7.09614e+06	2.0703e-07
e $[-]$	0.000731104	0.000731104	1.76321e-05	0.0011219	0.0011219	8.8506e-05
i [°]	51.6871	51.6871	8.31844e-10	92.0316	92.0316	3.2867e-08
Ω [°]	127.549	127.549	1.95492e-08	296.138	296.138	6.5539e-09
ω [°]	74.2199	74.2199	2.49098e-09	120.688	120.688	3.85834e-05
θ [°]	24.1003	24.1003	2.67354e-09	239.544	239.544	3.26752e-05
E [°]	24.0832	24.0832	1.24563e-09	239.599	239.599	4.36334e-05
M [°]	24.0661	24.0661	5.29195e-09	239.655	239.655	1.94195e-05

Table 1: Unittest results for the Cartesian components to Kepler Elements function using the examples from the lecture slides.

	Example 1			Example 2		
	Result	Expected	Error%	Result	Expected	Error%
x $[m]$	-2.70082e+06	-2.70082e+06	-2.0929e-08	3.12697e+06	3.12697e+06	1.45115e-07
y $[m]$	-3.31409e+06	-3.31409e+06	-3.07529e-08	-6.37445e+06	-6.37445e+06	-6.19502e-08
z $[m]$	5.26635e+06	5.26635e+06	1.28802e-08	28673.6	28673.6	8.48746e-06
$\dot{x}$ $[m\ s^{-1}]$	5168.61	5168.61	3.1752e-08	-254.912	-254.912	-1.44902e-06
$\dot{y}$ $[m\ s^{-1}]$	-5597.55	-5597.55	-4.78342e-08	-83.3011	-83.3011	-1.51447e-06
$\dot{z}$ $[m\ s^{-1}]$	-868.878	-868.878	-5.5328e-08	7485.71	7485.71	2.72705e-08

Table 2: Unittest results for the Kepler Elements to Cartesian components function using the examples from the lecture slides.

4 Assignment results

All calculations where done using $\mu = 3.98600441e14$ $[m^3 s^{-2}]$.

4.1 Cartesian components to Kepler elements

These are the results from the first part of the assignment. As a sanity check these results where passed to the Kepler elements to Cartesian components function which yielded results within $1e-07$ relative tolerance to the original inputs. Thus it likely that these results are correct.

$$
\begin{aligned}
a &= 12273086.181\ [m] \\
e &= 0.0050221667\ [m] \\
i &= 109.81877383\ [deg] \\
\Omega &= 132.23369779\ [deg] \\
\omega &= 105.06673299\ [deg] \\
\theta &= 50.027991349\ [deg] \\
E &= 49.807826568\ [deg] \\
M &= 49.588019690\ [deg]
\end{aligned}
$$

4.2 Kepler elements to Cartesian components

These are the results from the second part of the assignment. As a sanity check these results where passed to the Cartesian components to Kepler elements function which yielded results within $1e-07$ relative tolerance to the original inputs. Thus it likely that these results are correct.

$$
\begin{aligned}
x &= -5760654.2301 \ [m] \\
y &= -4856967.4882 \ [m] \\
z &= -9627444.8622 \ [m] \\
\dot{x} &= 4187.6612513 \ [ms^{-1}] \\
\dot{y} &= -3797.5451854 \ [ms^{-1}] \\
\dot{z} &= -683.61512604 \ [ms^{-1}]
\end{aligned}
$$

Appendices

A Software

The requirements for executing the assignment code are:

- Python 3.8 or higher
  - JupyterLab
  - Numpy
  - Scipy
  - Tabulate

Please read the README file for a description of the repository file structure.

A.1 Docker

This is optional, but to ensure a consistent development environment a docker container is used to host the python interpreter and JupyterLab server used to develop and run the code. To start the container first ensure that both docker and docker-compose are installed on the host machine. Starting the JupyterLab instance can be done by running the following command in a terminal from the repository root directory (the directory containing the docker-compose.yml file).

```
docker-compose up
```

Instructions on how to open JupyterLab will appear in the terminal after the container has successfully started. This JupyterLab instance will have all requirements preinstalled and the assignment code should run without issues.

References

[1] James Richard Wertz. "Mission geometry; orbit and constellation design and management". In: Space Technology Library (2001).