Quaternions and the perifocal reference frame

Mont Blanc

July 7, 2022

1 *Introduction*

This is a short, (but somewhat technical) note on the rather arcane subject of converting a point in the perifocal reference frame. It is almost a verbatim copy a post written by this author on the Orbiter-forum website in July, 2019. Some tweaks to spelling, grammar and other oddities have been made.

To get the ball rolling, let's introduce the perifocal reference frame.

2 *The perifocal reference frame*

The perifocal reference frame is the natural (and traditional) reference frame in which to calculate the position of a body in a standard Keplerian (circular, elliptical, parabolic or hyperbolic) orbit. Specifically, in the perifocal reference frame, the periapsis is located at the 3-vector point:

$$\mathbf{X}_p = \left(+a\ (1-e)\,, 0, 0\right) \tag{1}$$

where, as usual, a denotes the orbital semi-major axis; and e the orbital eccentricity. Similarly, the apoapsis is located at the 3-vector point:

$$\mathbf{X}_a = \left(-a\ (1+e)\,, 0, 0\right) \tag{2}$$

More generally, all points of a Keplerian orbit in the perifocal reference frame lie in the x-y plane of that reference frame - such that the orbital periapsis lies on the positive x-axis; and such that the focus of the ellipse (*i.e.,* where the gravitating body is located) is located at the origin of the coordinate system. Hence the name of the coordinate system. In the perifocal reference frame, any point on the Keplerian orbot can be writtes as:

$$\mathbf{X} = (r\ \cos\nu\ r\ \sin\nu,\, 0) \tag{3}$$

where the true anomaly, ν, is usually determined frim the mean anomaly, M - a measure of ordinary clock time - via the application of Kepler's equation. Moreover, for a Keplerian orbit, the orbital radius, r, is given by the general expression:

$$r = \frac{a\,(1-e^2)}{1+e\,\cos\nu} \tag{4}$$

In other words, in the perifocal reference frame, we need just three of the six orbital elements to specify the posotion of a body in a Keplerian orbit - the semi-major axis, a; the orbital eccentricity, e; and the truen anomaly, ν.

3 Quaternions

ALL OF THIS IS PRETTY STANDARD STUFF. However, one often wants to convert a point in the perifocal reference frame to a more general reference frame - *e.g.*, Orbiter's Earth-centric equatorial or ecliptic reference frames. This transformation is defined in terms of the three remaining orbital elements - Ω, the Longitude of the Ascending Node (LAN); ι, the orbital inclination; and ω, the Argument of Periapsis. These three orbital elements are, of course, all angles and together form an Euler angle triplet.

Instead of using the rather cumbersome apparatus of Euler angle rotatin matrices, one can rather more elegantly carry out the same transformation using quaternions. In this section, then, I'll show how to construct (and apply) the quaternions needed to carry out the rotation from the perifocal reference frame.

BUT FIRST, A FEW COMMENTS ON QUATERNIONS. According to Wikipedia:

> *"In mathematics, the quaterniosn are a number system that extends the complex numbers. They were first described by Irish mathematician William Rowan Hamilton in 1843 and applied to mechanics in three-dimensional space."*

Quaternions are generally represented in the form:

$$\mathcal{Q} = q_0 + q_1\,\mathbf{i} + q_2\,\mathbf{j} + q_3\,\mathbf{k} \tag{5}$$

where q_0, q_1, q_2 and q_3 are real numbers; and $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are the fundamental unit quaternions such that:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \tag{6}$$

$$\mathbf{i}\,\mathbf{j} = \mathbf{k} \tag{7}$$

$$\mathbf{j}\,\mathbf{k} = \mathbf{i} \tag{8}$$

$$\mathbf{k}\,\mathbf{i} = \mathbf{j} \tag{9}$$

The general rules for adding, subtracting, multiplying and dividng quaternions is treated well in unnumerable books and articles, so I'm going to assume that you either know them or can readily Google them. It is worth pointing out, thoough, that the conjugate, $\mathcal{Q}^\dagger$, of the quaternion $\mathcal{Q}$ is given by the epxression:

$$\mathcal{Q}^\dagger = q_0 - q_1\,\mathbf{i} - q_2\,\mathbf{j} - q_3\,\mathbf{k} \tag{10}$$

GIVEN ALL OF THIS, a rotation of a 3-vector (x, y, z), say, is carried out using quaternions by first constructing a new quaternion from the 3-vector such thatt:

$$\mathcal{P} = 0 + x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} \tag{11}$$

and then carrying out the quaternion multiplication:

$$\mathcal{P}' = \mathcal{Q}\,\mathcal{P}\,\mathcal{Q}^\dagger \tag{12}$$

where we also impose the condition that $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The resulting quaternion is of the form:

$$\mathcal{P}' = 0 + X\,\mathbf{i} + Y\,\mathbf{j} + Z\,\mathbf{k} \tag{13}$$

with:

$$x^2 + y^2 + z^2 = X^2 + Y^2 + Z^2 \tag{14}$$

.

IN OTHER WORDS, this is a transformation that maps any 3-vector to an new 3-vector such that the length of the vector is preserved. Rotations fall into this class of transformations and so are defined by these quaternion arrangements. All of this actually rather elegant - and all one needs to know to carry out the rotation is how to take conjugates and how to multiply quaternions.

4 *Application to orbital coordinate transformations*

OF COURSE, WE ALSO NEED TO KNOW HOW TO CONSTRUCT THE ROTATION QUATERNION $\mathcal{Q}$ so that we can carry out the rotation - *i.e.*, the coordinate transformation. As it turns out, this is pretty simple. In terms of the three Euler angles - Ω, the Longitude of the Ascending Node; ι, the orbital inclination; and ω, the Argument of Periapsis, we construct the quaternion components of the rotation quaternion, $\mathcal{Q}$ as:

$$q_0 = \cos\left(\frac{\iota}{2}\right) \cos\left(\frac{\Omega + \omega}{2}\right) \tag{15}$$

$$q_1 = \sin\left(\frac{\iota}{2}\right) \cos\left(\frac{\Omega - \omega}{2}\right) \tag{16}$$

$$q_2 = \sin\left(\frac{\iota}{2}\right) \sin\left(\frac{\Omega - \omega}{2}\right) \tag{17}$$

$$q_3 = \cos\left(\frac{\iota}{2}\right) \sin\left(\frac{\Omega + \omega}{2}\right) \tag{18}$$

such that:

$$\mathcal{Q} = q_0 + q_1\,\mathbf{i} + q_2\,\mathbf{j} + q_3\,\mathbf{k} \tag{19}$$

Then, for an arbitrary 3-vector, (x, y, z), in the perifocal reference frame, we construct the quaternion, $\mathcal{P}$:

$$\mathcal{P} = 0 + x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} \tag{20}$$

and then carry out the quaternion multiplication:

$$\mathcal{P}' = \mathcal{Q}\,\mathcal{P}\,\mathcal{Q}^{\dagger} = 0 + X\,\mathbf{i} + Y\,\mathbf{j} + Z\,\mathbf{k} \tag{21}$$

And we read off the rotated 3-vector as (X, Y, Z).

5 *Invertting the transformation*

If we want to invert the transformation and to transform a point (X, Y, Z) from an equatorial or ecliptic reference frame back to the perifocal reference frame, this can easily achieved by constructing $\mathcal{P}'$ as:

$$\mathcal{P}' = 0 + X\,\mathbf{i} + Y\,\mathbf{j} + Z\,\mathbf{k} \tag{22}$$

and then carrying out the quaternion multiplication:

$$\mathcal{P} = \mathcal{Q}^{\dagger}\,\mathcal{P}'\,\mathcal{Q} = 0 + x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} \tag{23}$$

where we reading of the resulting 3-vector $(x, y\,z)$ in the perifocal reference frame from the quaternion $\mathcal{P}$. Simple!

6 *A worked example*

OK, let's work through a more or less realistic example of how this all works in practice. Let's take a specific example from the "Docked at the ISS" XR2 Ravenstar stock scenario. With the scenarion paused at $MJD = 51984.64994$, and using the Orbiter 2016 scenario editor, we

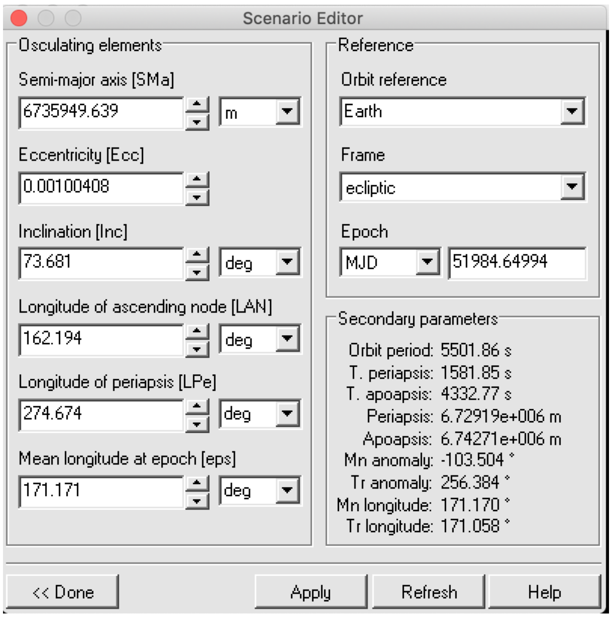

Figure 1: Screen capture of the Orbiter 2016 scenario editor for the XR2-Ravenstar showing orbital elements for a specific MJD

can read off directly the orbital elements of the ISS' orbit as shown in Figure 1:

Clearly, in the Earth-centric ecliptic reference frame:

$$
\begin{aligned}
a &= 7735949.639 \\
e &= 0.00100408 \\
\nu &= 256.384^\circ
\end{aligned}
$$

and also:

$$
\begin{aligned}
\Omega &= 162.195^\circ \\
\iota &= 73.681^\circ \\
\omega &= 112.480^\circ
\end{aligned}
$$

where we have used the idenity that the Longitude of Periapsis, 274.674°, is just the sume of the Argument of Periapsis, ω, and the Longitude of the Ascending Node, 162.194°.

From this, we calculate:

$$
\begin{aligned}
q_0 &= \cos\left(\frac{\iota}{2}\right)\,\cos\left(\frac{\Omega+\omega}{2}\right) = -0.5885082 \\
q_1 &= \sin\left(\frac{\iota}{2}\right)\,\cos\left(\frac{\Omega-\omega}{2}\right) = +0.5440433 \\
q_2 &= \sin\left(\frac{\iota}{2}\right)\,\sin\left(\frac{\Omega-\omega}{2}\right) = +0.2520404 \\
q_3 &= \cos\left(\frac{\iota}{2}\right)\,\sin\left(\frac{\Omega+\omega}{2}\right) = +0.5423565
\end{aligned}
$$

and so can construct the rotation quaternion as:

$$\mathcal{Q} = -0.5885082 + 0.5440433\,\mathbf{i} + 0.2520404\,\mathbf{j} + 0.5423565\,\mathbf{k}$$

and its conjugate as:

$$\mathcal{Q} = -0.5885082 - 0.5440433\,\mathbf{i} - 0.2520404\,\mathbf{j} - 0.5423565\,\mathbf{k}$$

Then, in the quaternion form of the perifocal coordiates, the position of the ISS is:

$$
\begin{aligned}
\mathcal{P} &= \frac{a\,(1-e^2}{1+e\,\cos\nu}\,\left(0+\cos\nu\,\mathbf{i}+\sin\nu\,\mathbf{j}+0\,\mathbf{k}\right) \\
&= 0 - 1586106.976\,\mathbf{i} - 6548179.005\,\mathbf{j} + 0\,\mathbf{k}
\end{aligned}
$$

We can carry out the rotation to the Earth-centric ecliptic reference frame from the perifocal reference by calculating:

$$
\begin{aligned}
\mathcal{P}' &= \mathcal{Q}\,\mathcal{P}\,\mathcal{Q}^\dagger \\
&= 0 - 6427381.91\,\mathbf{i} + 1757957.97\,\mathbf{j} + 996357.96\,\mathbf{k}
\end{aligned}
$$

In turn, from this, we can immediately read of the Earth-centric ecliptic reference frame positio coordinates of the ISS as:

$$\mathbf{P}' = (-6427381.91,\ 1757957.97,\ 996357.96)$$

Finally, to convert from a conventional right-handed coordinate system to Orbiter's somewhat quirkt left-handed coordinate system, we simply swap the y and z components of this vector to end up with an Orbiter 2016 compatible representation:

$$\bar{\mathbf{P}}' = (-6427381.91,\ 996357.96,\ 1757957.97)$$

How does this result compare with the state vector calculated by Orbiter 2016? Again using the Sceario Editor (see Figure 2), we can read off the position vector components directly as:

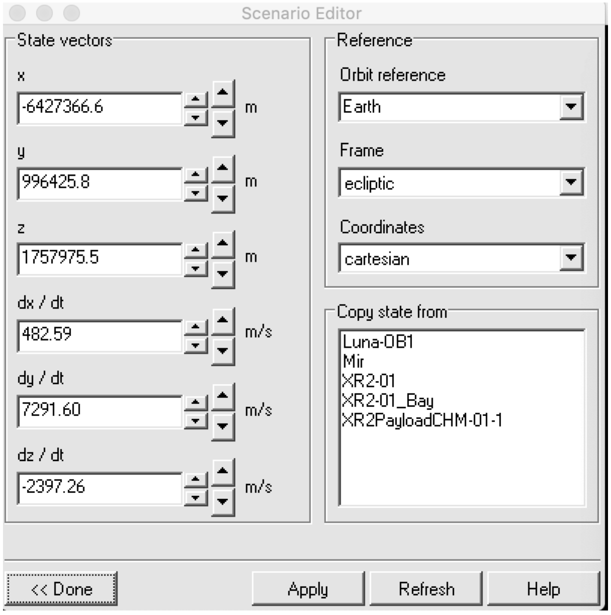

Figure 2: Screen capture of the Orbiter 2016 scenario editor for the XR2-Ravenstar showing state vectors for a specific *MJD*

$$\bar{\mathbf{P}}' = (-6427366.1, 996435.8, 175975.5)$$

This isn't exactly the same as the calculated result as set out above but it is close - with a hundred metres or so. Why the difference? Well, the Scenario Editor only reports orbital elements to a relatively low precision - about 5 or 6 significant digits. If we were to use the full machine precision of those parameters, which we could obtain by interrogating Orbiter 2016's internal system via the API, the calculated values would match those reported by Orbiter 2016 directly with very high precision.

7 *In summary*

QUATERNIONS DO THE SAME JOB AS ROTATIO MATRICES in mapping points from one rectilinear coordinate system to another. However, they do so in a more elegant way and, possibly, more efficiently. According to Martin Schweiger, founder an developer of the Orbiter Space Flight Simulator:

> *"One of the nice things about quaternions is that they describe the rotation more concisely. A rotation matrix consists of 9 scalar values, but there is some redundancy since the conditions of orthogonality and normalisation introduce constraints that don't allow for just any combination of 9 values. Indeed, rotation matrices that are continuously updated tend to need periodic*

re-orthoganalisation and re-normalisation, since numerical errors build up over time. Quaternions don't have that problem."