Conversion between quaternions and Euler angles

From Wikipedia, the free encyclopedia

Spatial rotations in three dimensions can be parametrized using both Euler angles and unit quaternions. This article explains how to convert between the two representations. Actually this simple use of "quaternions" was first presented by Euler some seventy years earlier than Hamilton to solve the problem of magic squares. For this reason the dynamics community commonly refers to quaternions in this application as "Euler parameters".

Contents

Definition

A unit quaternion can be described as:

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$$
$$|\mathbf{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

We can associate a quaternion with a rotation around an axis by the following expression

$$\mathbf{q}_0 = \cos(\alpha/2)$$
$$\mathbf{q}_1 = \sin(\alpha/2)\cos(\beta_x)$$
$$\mathbf{q}_2 = \sin(\alpha/2)\cos(\beta_y)$$
$$\mathbf{q}_3 = \sin(\alpha/2)\cos(\beta_z)$$

where α is a simple rotation angle (the value in radians of the angle of rotation) and $\cos(\beta_x)$, $\cos(\beta_y)$ and $\cos(\beta_z)$ are the "direction cosines" locating the axis of rotation (Euler's Theorem).

Rotation matrices

The orthogonal matrix (post-multiplying a column vector) corresponding to a clockwise/left-handed rotation by the unit quaternion $q = q_0 + iq_1 + jq_2 + kq_3$ is given by the inhomogeneous expression

$$\begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

or equivalently, by the homogeneous expression
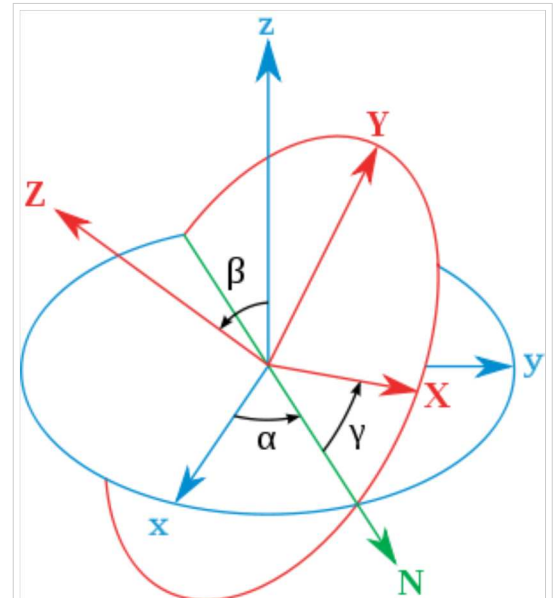
Euler angles - The xyz (fixed) system is shown in blue, the XYZ (rotated) system is shown in red. The line of nodes, labelled N, is shown in green.

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

If $q_0 + iq_1 + jq_2 + kq_3$ is not a unit quaternion then the homogeneous form is still a scalar multiple of a rotation matrix, while the inhomogeneous form is in general no longer an orthogonal matrix. This is why in numerical work the homogeneous form is to be preferred if distortion is to be avoided.

The orthogonal matrix (post-multiplying a column vector) corresponding to a clockwise/left-handed rotation with Euler angles φ,θ,ψ, with **x-y-z** convention, is given by:

$$\begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

Conversion

By combining the quaternion representations of the Euler rotations we get

$$\mathbf{q} = R_z(\psi)R_y(\theta)R_x(\phi) = [\cos(\psi/2) + \mathbf{k}\sin(\psi/2)][\cos(\theta/2) + \mathbf{j}\sin(\theta/2)][\cos(\phi/2) + \mathbf{i}\sin(\phi/2)]$$

$$\mathbf{q} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix}$$

For Euler angles we get:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)} \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \arctan\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \end{bmatrix}$$

arctan and arcsin have a result between -Pi/2 and Pi/2. With three rotation between -Pi/2 and Pi/2 you can't have all possible orientations. You need to replace the arctan by atan2 to generate all the orientation.

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}$$

Relationship with Tait-Bryan angles

Similarly for Euler angles, we use the Tait-Bryan angles (in terms of flight dynamics):

- Roll - φ: rotation about the X-axis
- Pitch - θ: rotation about the Y-axis
- Yaw - ψ: rotation about the Z-axis

where the X-axis points forward, Y-axis to the right and Z-axis downward and in the example to follow the rotation occurs in the order yaw, pitch, roll (about body-fixed axes).

Singularities

One must be aware of singularities in the Euler angle parametrization when the pitch approaches $\pm$ 90° (north/south pole). These cases must be handled specially. The common name for this situation is gimbal lock.

Tait-Bryan angles for an aircraft

Code to handle the singularities is derived on this site: www.euclideanspace.com (http://www.euclideanspace.com/maths/geometry/rotations/conversions/quaternionToEuler/)
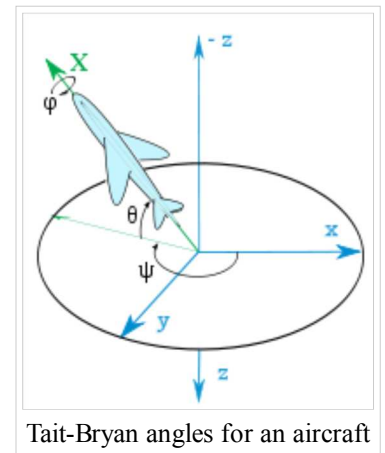
See also

- Rotation operator (vector space)
- Quaternions and spatial rotation
- Euler Angles
- Rotation matrix

External links

- Q60. How do I convert Euler rotation angles to a quaternion? (http://www.j3d.org/matrix_faq/matrfaq_latest.html#Q60) and related questions at The Matrix and Quaternions FAQ

Retrieved from "http://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles"
Categories: Rotational symmetry | Euclidean symmetries | 3D computer graphics

- This page was last modified on 28 November 2010 at 02:50.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
  Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.