

Rotation formalisms in three dimensions

In geometry, various **formalisms** exist to express a rotation in three dimensions as a mathematical transformation. In physics, this concept extends to classical mechanics where rotational (or angular) kinematics is the science of describing with numbers the purely rotational motion of an object.

According to Euler's rotation theorem the general displacement of a rigid body (or three-dimensional coordinate system) with one point fixed is described by a rotation about some axis. This allows the use of rotations to express orientations as a single rotation from a reference placement in space of the rigid body (or coordinate system). Furthermore, such a rotation may be uniquely described by a minimum of three parameters. However, for various reasons, there are several ways to represent it. Many of these representations use more than the necessary minimum of three parameters, although each of them still has only three degrees of freedom.

An example where rotation representation is used is in computer vision, where an automated observer needs to track a target. Let's consider a rigid body, with an orthogonal right-handed triad $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, and $\hat{\mathbf{w}}$ of unit vectors fixed to its body (representing the three axes of the object's coordinate system). The basic problem is to specify the orientation of this triad, and hence the rigid body, in terms of the reference coordinate system (in our case the observer's coordinate system).

Rotation matrix

The above mentioned triad of unit vectors is also called a basis. Specifying the coordinates (scalar components) of this basis in its current (rotated) position, in terms of the reference (non-rotated) coordinate axes, will completely describe the rotation. The three unit vectors $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ which form the rotated basis each consist of 3 coordinates, yielding a total of 9 parameters. These parameters can be written as the elements of a 3×3 matrix \mathbf{A} , called a **rotation matrix**. Typically, the coordinates of each of these vectors are arranged along a column of the matrix (however, beware that an alternative definition of rotation matrix exists and is widely used, where the vectors coordinates defined above are arranged by rows^[1])

$$\mathbf{A} = \begin{bmatrix} \hat{\mathbf{u}}_x & \hat{\mathbf{v}}_x & \hat{\mathbf{w}}_x \\ \hat{\mathbf{u}}_y & \hat{\mathbf{v}}_y & \hat{\mathbf{w}}_y \\ \hat{\mathbf{u}}_z & \hat{\mathbf{v}}_z & \hat{\mathbf{w}}_z \end{bmatrix}$$

The elements of the rotation matrix are not all independent € as Euler's rotation theorem dictates, the rotation matrix has only three degrees of freedom. The rotation matrix has the following properties:

- , \mathbf{A} is a real, orthogonal matrix, hence each of its rows or columns represents a unit vector.
- , The eigenvalues of \mathbf{A} are

$$\{1, e^{\pm i\theta}\} = \{1, \cos(\theta) + i \sin(\theta), \cos(\theta) - i \sin(\theta)\}$$

where i is the standard imaginary unit with the property $i^2 = -1$

- , The determinant of \mathbf{A} is +1, equivalent to the product of its eigenvalues.
- , The trace of \mathbf{A} is $1 + 2 \cos(\theta)$, equivalent to the sum of its eigenvalues.

The angle θ which appears in the eigenvalue expression corresponds to the angle of the Euler axis and angle representation. The eigenvector corresponding with the eigenvalue of 1 is the accompanying Euler axis, since the axis is the only (nonzero) vector which remains unchanged by left-multiplying (rotating) it with the rotation matrix.

The above properties are equivalent to:

$$\begin{aligned} |\hat{\mathbf{u}}| &= |\hat{\mathbf{v}}| = 1 \\ \hat{\mathbf{u}} \cdot \hat{\mathbf{v}} &= 0 \\ \hat{\mathbf{u}} \times \hat{\mathbf{v}} &= \hat{\mathbf{w}} \end{aligned}$$

which is another way of stating that $\hat{u}, \hat{v}, \hat{w}$ form a 3D orthonormal basis. Note that the statements above constitute a total of 6 conditions (the cross product contains 3), leaving the rotation matrix with just 3 degrees of freedom as required.

Two successive rotations represented by matrices \mathbf{A}_1 and \mathbf{A}_2 are easily combined as follows: $\mathbf{A}_{\text{total}} = \mathbf{A}_2 \mathbf{A}_1$ (Note the order, since the vector being rotated is multiplied from the right). The ease by which vectors can be rotated using a rotation matrix, as well as the ease of combining successive rotations, make the rotation matrix a very useful and popular way to represent rotations, even though it is less concise than other representations.

Euler axis and angle (rotation vector)

From Euler's rotation theorem we know that any rotation can be expressed as a single rotation about some axis. The axis is the unit vector (unique except for sign) which remains unchanged by the rotation. The magnitude of the angle is also unique, with its sign being determined by the sign of the rotation axis.

The axis can be represented as a three-dimensional unit vector $\hat{e} = [e_x \ e_y \ e_z]^T$, and the angle by a scalar θ .

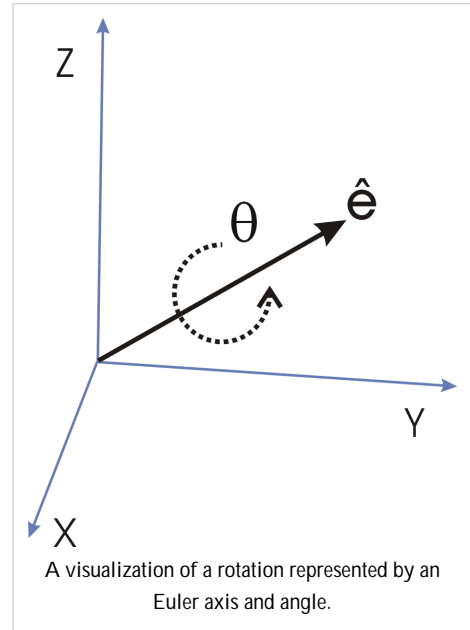
Since the axis is normalized, it has only two degrees of freedom. The angle adds the third degree of freedom to this rotation representation.

One may wish to express rotation as a **rotation vector**, a non-normalized three-dimensional vector the direction of which specifies the axis, and the length of which is θ :

$$\mathbf{v} = \theta \hat{e}$$

The rotation vector is in some contexts useful, as it represents a three-dimensional rotation with only three scalar values (its scalar components), representing the three degrees of freedom. This is also true for representations based on sequences of three Euler angles (see below).

If the rotation angle θ is zero, the axis is not uniquely defined. Combining two successive rotations, each represented by an Euler axis and angle, is not straightforward, and in fact does not satisfy the law of vector addition, which shows that finite rotations are not really vectors at all. It is best to employ the rotation matrix or quaternion notation, calculate the product, and then convert back to Euler axis and angle.



Euler rotations

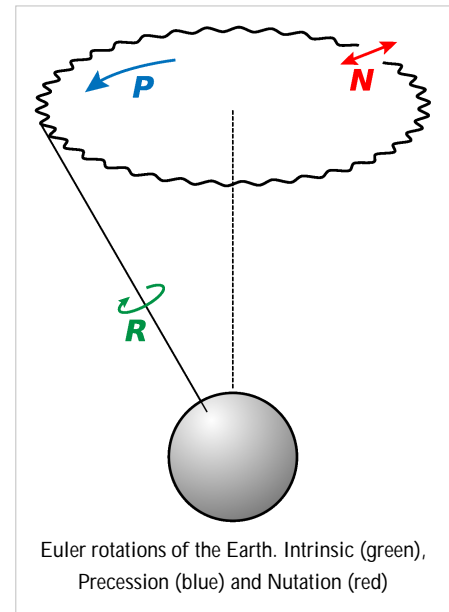
The idea behind Euler rotations is to split the complete rotation of the coordinate system into three simpler constitutive rotations, called Precession, Nutation, and intrinsic rotation, being each one of them an increment on one of the Euler angles. Notice that the outer matrix will represent a rotation around one of the axes of the reference frame, and the inner matrix represents a rotation around one of the moving frame axis. The middle matrix represent a rotation around an intermediate axis called **line of nodes**.

Unfortunately, the definition of Euler angles is not unique and in the literature many different conventions are used. These conventions depend on the axes about which the rotations are carried out, and their sequence (since rotations are not commutative).

The convention being used is usually indicated by specifying the axes about which the consecutive rotations (before being composed) take place, referring to them by index (1, 2, 3) or letter (X, Y, Z). The engineering and robotics communities typically use 3-1-3 Euler angles.

Notice that after composing the independent rotations, they do not rotate about their axis anymore. The most external matrix rotates the other two, leaving the second rotation matrix over the line of nodes, and the third one in a frame comoving with the body. There are $3 \cdot 3 \cdot 3 = 27$ possible combinations of three basic rotations but only $3 \cdot 2 \cdot 2 = 12$ of them can be used for representing arbitrary 3D rotations as Euler angles. These 12 combinations avoid consecutive rotations around the same axis (such as XXY) which would reduce the degrees of freedom that can be represented.

Therefore Euler angles are never expressed in terms of the external frame, or in terms of the co-moving rotated body frame, but in a mixture. Other conventions (e.g., rotation matrix or quaternions) are used to avoid this problem.



Quaternions

Quaternions (Euler symmetric parameters) have proven very useful in representing rotations due to several advantages above the other representations mentioned in this article.

A quaternion representation of rotation is written as a normalized four-dimensional vector

$$\hat{\mathbf{q}} = [q_1 \ q_2 \ q_3 \ q_4]^T$$

In terms of the Euler axis

$$\hat{\mathbf{e}} = [e_x \ e_y \ e_z]^T$$

and angle

$$\theta$$

this vector's elements are expressed as follows:

$$\begin{aligned}
q_1 &= e_x \sin\left(\frac{\theta}{2}\right) \\
q_2 &= e_y \sin\left(\frac{\theta}{2}\right) \\
q_3 &= e_z \sin\left(\frac{\theta}{2}\right) \\
q_4 &= \cos\left(\frac{\theta}{2}\right)
\end{aligned}$$

The above definition follows the convention as used in (Wertz 1980) and (Markley 2003). An alternative definition used in some publications defines the "scalar" term as the first quaternion element, with the other elements shifted down one position. (Coutsias 1999), (Schmidt 2001)

Inspection shows that the quaternion parametrization obeys the following constraint:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$$

The last term (in our definition) is often called the scalar term, which has its origin in quaternions when understood as the mathematical extension of the complex numbers, written as

$$a + bi + cj + dk \text{ with } \{a, b, c, d\} \in \mathbb{R}$$

and where $\{i, j, k\}$ are the hypercomplex numbers satisfying

$$\begin{aligned}
i^2 &= j^2 = k^2 = -1 \\
ij &= -ji = k \\
jk &= -kj = i \\
ki &= -ik = j
\end{aligned}$$

Quaternion multiplication is performed in the same manner as multiplication of complex numbers, except that the order of elements must be taken into account, since multiplication is not commutative. In matrix notation we can write quaternion multiplication as

$$\tilde{\mathbf{q}} \otimes \mathbf{q} = \begin{bmatrix} q_4 & -q_3 & -q_2 & q_1 \\ -q_3 & q_4 & -q_1 & q_2 \\ -q_2 & -q_1 & q_4 & q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \\ \tilde{q}_4 \end{bmatrix}$$

Combining two consecutive quaternion rotations is therefore just as simple as using the rotation matrix. Remember that two successive rotation matrices, \mathbf{A}_1 followed by \mathbf{A}_2 , are combined as follows:

$$\mathbf{A}_3 = \mathbf{A}_2 \mathbf{A}_1$$

We can represent this quaternion parameters in a similarly concise way. Please note the *inverse ordering* of quaternion multiplication when compared to matrix multiplication.

$$\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2$$

Quaternions are a very popular parametrization due to the following properties:

- , More compact than the matrix representation and less susceptible to round-off errors
- , The quaternion elements vary continuously over the unit sphere in \mathbb{R}^4 , (denoted by S^3) as the orientation changes, avoiding discontinuous jumps (inherent to three-dimensional parameterizations)
- , Expression of the rotation matrix in terms of quaternion parameters involves no trigonometric functions
- , It is simple to combine two individual rotations represented as quaternions using a quaternion product

Like rotation matrices, quaternions must sometimes be re-normalized due to rounding errors, to make sure that they correspond to valid rotations. The computational cost of re-normalizing a quaternion, however, is much less than for normalizing a 3×3 matrix.

Rodrigues parameters

Rodrigues parameters can be expressed in terms of Euler axis and angle as follows:

$$\mathbf{r} = \hat{\mathbf{e}}\theta$$

This has a discontinuity at 180° (π radians): each vector, \mathbf{r} , with a norm of π radians represent the same rotation as $-\mathbf{r}$.

Similarly, the Gibbs representation can be expressed as follows:

$$\mathbf{g} = \hat{\mathbf{e}} \tan\left(\frac{\theta}{2}\right)$$

A rotation \mathbf{g} followed by a rotation \mathbf{f} in Gibbs representation has the form

$$(\mathbf{g}, \mathbf{f}) = \frac{\mathbf{g} + \mathbf{f} - \mathbf{g} \times \mathbf{f}}{1 - \mathbf{g} \cdot \mathbf{f}}$$

The **Gibbs vector** has the advantage (or disadvantage, depending on context) that 180° rotations cannot be represented. (Even using floating point numbers that include infinity, rotation direction cannot be well-defined; for example, naively a 180° rotation about the axis $(1, 1, 0)$ would be $(\infty, \infty, 0)$, which is the same representation as 180° rotation about $(1, 0.0001, 0)$.)

Modified Rodrigues parameters (MRPs) can be expressed in terms of Euler axis and angle by:

$$\mathbf{p} = \hat{\mathbf{e}} \tan\left(\frac{\theta}{4}\right)$$

The modified Rodrigues parameterization shares many characteristics with the rotation vector parametrization, including the occurrence of discontinuous jumps in the parameter space when incrementing the rotation.

Cayley-Klein parameters

See definition at Wolfram Mathworld ^[2]

Rotors in a geometric algebra

The formalism of geometric algebra (GA) provides an extension and interpretation of the quaternion method. Central to GA is the geometric product of vectors, an extension of the traditional inner and cross products, given by

$$ab = a \cdot b + a \wedge b$$

where the symbol \wedge denotes the outer product. This product of vectors a, b produces two terms: a scalar part from the inner product and a bivector part from the outer product. This bivector describes the plane perpendicular to what the cross product of the vectors would return.

Bivectors in GA have some unusual properties compared to vectors. Under the geometric product, bivectors have negative square: the bivector $\hat{x}\hat{y}$ describes the xy -plane. Its square is $(\hat{x}\hat{y})^2 = \hat{x}\hat{y}\hat{x}\hat{y}$. Because the unit basis vectors are orthogonal to each other, the geometric product reduces to the antisymmetric outer product \hat{x} and \hat{y} can be swapped freely at the cost of a factor of -1 . The square reduces to $-\hat{x}\hat{x}\hat{y}\hat{y} = -1$ since the basis vectors themselves square to $+1$.

This result holds generally for all bivectors, and as a result the bivector plays a role similar to the imaginary unit. Geometric algebra uses bivectors in its analogue to the quaternion, the *rotor*, given by $R = \exp\left(\frac{-\hat{B}\theta}{2}\right) = \cos\frac{\theta}{2} - \hat{B}\sin\frac{\theta}{2}$, where \hat{B} is a unit bivector that describes the plane of rotation. Because \hat{B} squares to -1 , the power series expansion of R generates the trigonometric functions. The rotation formula that maps a vector a to a rotated vector b is then

$$b = RaR^\dagger$$

where $R^\dagger = \exp(\frac{1}{2}\hat{B}\theta) = \cos \frac{\theta}{2} + \hat{B} \sin \frac{\theta}{2}$ is the *reverse* of R (reversing the order of the vectors in B is equivalent to changing its sign).

Example. A rotation about the axis $\hat{v} = \frac{1}{\sqrt{3}}(\hat{x} + \hat{y} + \hat{z})$ can be accomplished by converting \hat{v} to its dual bivector, $\hat{B} = \hat{x}\hat{y}\hat{z}\hat{v} = i\hat{v}$, where $i = \hat{x}\hat{y}\hat{z}$ is the unit volume element, the only trivector (pseudoscalar) in three-dimensional space. The result is $\hat{B} = \frac{1}{\sqrt{3}}(\hat{y}\hat{z} + \hat{z}\hat{x} + \hat{x}\hat{y})$. In three-dimensional space, however, it is often simpler to leave the expression for $\hat{B} = i\hat{v}$, using the fact that i commutes with all objects in 3D and also squares to $\bullet 1$. A rotation of the \hat{x} vector in this plane by an angle θ is then

$$\hat{x}' = R\hat{x}R^\dagger = e^{-i\hat{v}\frac{\theta}{2}}\hat{x}e^{i\hat{v}\frac{\theta}{2}} = \hat{x} \cos^2 \frac{\theta}{2} + i(\hat{x}\hat{v} - \hat{v}\hat{x}) \cos \frac{\theta}{2} \sin \frac{\theta}{2} + \hat{v}\hat{x}\hat{v} \sin^2 \frac{\theta}{2}$$

Recognizing that $i(\hat{x}\hat{v} - \hat{v}\hat{x}) = 2i(\hat{x} \wedge \hat{v})$ and that $-\hat{v}\hat{x}\hat{v}$ is the reflection of \hat{x} about the plane perpendicular to \hat{v} gives a geometric interpretation to the rotation operation: the rotation preserves the components that are parallel to \hat{v} and changes only those that are perpendicular. The terms are then computed:

$$\begin{aligned} \hat{v}\hat{x}\hat{v} &= \frac{1}{3}(-\hat{x} + 2\hat{y} + 2\hat{z}) \\ 2i\hat{x} \wedge \hat{v} &= 2i\frac{1}{\sqrt{3}}(\hat{x}\hat{y} + \hat{x}\hat{z}) = \frac{2}{\sqrt{3}}(\hat{y} - \hat{z}) \end{aligned}$$

The result of the rotation is then

$$\hat{x}' = \hat{x} \left(\cos^2 \frac{\theta}{2} - \frac{1}{3} \sin^2 \frac{\theta}{2} \right) + \frac{2}{3} \hat{y} \sin \frac{\theta}{2} \left(\sin \frac{\theta}{2} + \sqrt{3} \cos \frac{\theta}{2} \right) + \frac{2}{3} \hat{z} \sin \frac{\theta}{2} \left(\sin \frac{\theta}{2} - \sqrt{3} \cos \frac{\theta}{2} \right)$$

A simple check on this result is the angle $\theta = \frac{2}{3}\pi$. Such a rotation should map the \hat{x} to \hat{y} . Indeed, the rotation reduces to

$$\begin{aligned} \hat{x}' &= \hat{x} \left(\frac{1}{4} - \frac{1}{3} \frac{3}{4} \right) + \frac{2}{3} \hat{y} \frac{\sqrt{3}}{2} \left(\frac{\sqrt{3}}{2} + \sqrt{3} \frac{1}{2} \right) + \frac{2}{3} \hat{z} \frac{\sqrt{3}}{2} \left(\frac{\sqrt{3}}{2} - \sqrt{3} \frac{1}{2} \right) \\ &= 0\hat{x} + \hat{y} + 0\hat{z} = \hat{y} \end{aligned}$$

exactly as expected. This rotation formula is valid not only for vectors but for any multivector. In addition, when Euler angles are used, the complexity of the operation is much reduced. Compounded rotations come from multiplying the rotors, so the total rotor from Euler angles is

$$R = R_{\gamma'} R_{\beta'} R_{\alpha} = \exp \left(\frac{-i\hat{z}'\gamma}{2} \right) \exp \left(\frac{-i\hat{x}'\beta}{2} \right) \exp \left(\frac{-i\hat{z}\alpha}{2} \right)$$

but $\hat{x}' = R_{\alpha}\hat{x}R_{\alpha}^\dagger$ and $\hat{z}' = R_{\beta'}\hat{z}R_{\beta'}^\dagger$. These rotors come back out of the exponentials like so:

$$R_{\beta'} = \cos \frac{\beta}{2} - iR_{\alpha}\hat{x}R_{\alpha}^\dagger \sin \frac{\beta}{2} = R_{\alpha}R_{\beta}R_{\alpha}^\dagger$$

where R_{β} refers to rotation in the original coordinates. Similarly for the γ rotation, $R_{\gamma'} = R_{\beta'}R_{\gamma}R_{\beta'}^\dagger = R_{\alpha}R_{\beta}R_{\alpha}^\dagger R_{\gamma}R_{\alpha}R_{\beta}^\dagger R_{\alpha}^\dagger$. Noting that R_{γ} and R_{α} commute (rotations in the same plane must commute), and the total rotor becomes

$$R = R_{\alpha}R_{\beta}R_{\gamma}$$

Thus, the compounded rotations of Euler angles become a series of equivalent rotations in the original fixed frame.

While rotors in geometric algebra work almost identically to quaternions in three dimensions, the power of this formalism is its generality: this method is appropriate and valid in spaces with any number of dimensions. In 3D, rotations have three degrees of freedom, a degree for each linearly independent plane (bivector) the rotation can take place in. It has been known that pairs of quaternions can be used to generate rotations in 4D, yielding six degrees of freedom, and the geometric algebra approach verifies this result: in 4D, there are six linearly independent bivectors that can be used as the generators of rotations.

Conversion formulae between formalisms

Rotation matrix • Euler angles

The Euler angles (ϕ, θ, ψ) can be extracted from the rotation matrix \mathbf{A} by inspecting the rotation matrix in analytical form.

Using the x-convention, the 3-1-3 Euler angles ϕ , θ and ψ (around the Z , X and again the Z -axis) can be obtained as follows:

$$\begin{aligned}\phi &= \arctan2(A_{31}, A_{32}) \\ \theta &= \arccos(A_{33}) \\ \psi &= -\arctan2(A_{13}, A_{23})\end{aligned}$$

Note that $\arctan2(a, b)$ is equivalent to $\arctan(a/b)$ where it also takes into account the quadrant that the point (a, b) is in; see `atan2`.

When implementing the conversion, one has to take into account several situations.^[3]

- , There are generally two solutions in $(\bullet, \epsilon_\bullet)^3$ interval. The above formula works only when θ is from the interval $[0, \epsilon_\theta)^3$.
- , For special case $A_{33} = 0$, ϕ, ψ shall be derived from A_{11}, A_{12} .
- , There is infinitely many but countably many solutions outside of interval $(\bullet, \epsilon_\bullet)^3$.
- , Whether all mathematical solutions apply for given application depends on the situation.

The rotation matrix \mathbf{A} is generated from the Euler angles by multiplying the three matrices generated by rotations about the axes.

$$\mathbf{A} = \mathbf{A}_3 \mathbf{A}_2 \mathbf{A}_1 = \mathbf{A}_Z \mathbf{A}_Y \mathbf{A}_X$$

The axes of the rotation depend on the specific convention being used. For the x-convention the rotations are about the X , Y and Z axes with angles ϕ , θ and ψ , the individual matrices are as follows:

$$\begin{aligned}\mathbf{A}_X &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \\ \mathbf{A}_Y &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\ \mathbf{A}_Z &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

This yields

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

Note: This is valid for a right-hand system, which is the convention used in almost all engineering and physics disciplines.

Rotation matrix • Euler axis/angle

If the Euler angle θ is not a multiple of π , the Euler axis $\hat{\mathbf{e}} = [e_1 \ e_2 \ e_3]^T$ and angle θ can be computed from the elements of the rotation matrix \mathbf{A} as follows:

$$\theta = \arccos \left(\frac{1}{2} [A_{11} + A_{22} + A_{33} - 1] \right)$$

$$e_1 = \frac{A_{32} - A_{23}}{2 \sin \theta}$$

$$e_2 = \frac{A_{13} - A_{31}}{2 \sin \theta}$$

$$e_3 = \frac{A_{21} - A_{12}}{2 \sin \theta}$$

Alternatively, the following method can be used:

Eigen-decomposition of the rotation matrix yields the eigenvalues 1, and $\cos \theta \pm i \sin \theta$. The Euler axis is the eigenvector corresponding to the eigenvalue of 1, and the θ can be computed from the remaining eigenvalues.

The Euler axis can be also found using Singular Value Decomposition since it is the normalized vector spanning the null-space of the matrix $\mathbf{I} - \mathbf{A}$.

To convert the other way the rotation matrix corresponding to an Euler axis $\hat{\mathbf{e}} = [e_1 \ e_2 \ e_3]^T$ and angle θ can be computed according to the Rodrigues' rotation formula (with appropriate modification) as follows:

$$\mathbf{A} = \mathbf{I}_3 \cos \theta + (1 - \cos \theta) \hat{\mathbf{e}} \hat{\mathbf{e}}^T + [\hat{\mathbf{e}}]_{\times} \sin \theta$$

with \mathbf{I}_3 the 3×3 identity matrix, and

$$[\hat{\mathbf{e}}]_{\times} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

is the cross-product matrix.

Rotation matrix • quaternion

When computing a quaternion from the rotation matrix there is a sign ambiguity, since \mathbf{q} and $-\mathbf{q}$ represent the same rotation.

One way of computing the quaternion $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$ from the rotation matrix \mathbf{A} is as follows:

$$q_4 = \frac{1}{2} \sqrt{1 + A_{11} + A_{22} + A_{33}}$$

$$q_1 = \frac{1}{4q_4} (A_{32} - A_{23})$$

$$q_2 = \frac{1}{4q_4} (A_{13} - A_{31})$$

$$q_3 = \frac{1}{4q_4} (A_{21} - A_{12})$$

There are three other mathematically equivalent ways to compute \mathbf{q} . Numerical inaccuracy can be reduced by avoiding situations in which the denominator is close to zero. One of the other three methods looks as follows:^[4]

$$\begin{aligned}
q_1 &= \frac{1}{2} \sqrt{1 + A_{11} - A_{22} - A_{33}} \\
q_2 &= \frac{1}{4q_1} (A_{12} + A_{21}) \\
q_3 &= \frac{1}{4q_1} (A_{13} + A_{31}) \\
q_4 &= \frac{1}{4q_1} (A_{32} - A_{23})
\end{aligned}$$

The rotation matrix corresponding to the quaternion $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$ can be computed as follows:

$$\mathbf{A} = (q_4^2 - \check{\mathbf{q}}^T \check{\mathbf{q}}) \mathbf{I}_3 + 2\check{\mathbf{q}}\check{\mathbf{q}}^T + 2q_4 \mathcal{Q}$$

with \mathbf{I}_3 the 3 • 3 identity matrix, and

$$\check{\mathbf{q}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \mathcal{Q} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

which gives

$$\mathbf{A} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_1q_4 + q_2q_3) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

or equivalently

$$\mathbf{A} = \begin{bmatrix} -1 + 2q_1^2 + 2q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -1 + 2q_2^2 + 2q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_1q_4 + q_2q_3) & -1 + 2q_3^2 + 2q_4^2 \end{bmatrix}$$

Euler angles • quaternion

We will consider the x-convention 3-1-3 Euler Angles for the following algorithm. The terms of the algorithm depend on the convention used.

We can compute the quaternion $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$ from the Euler angles (ϕ, θ, ψ) as follows:

$$\begin{aligned}
q_1 &= -\cos\left(\frac{\phi - \psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \\
q_2 &= -\sin\left(\frac{\phi - \psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \\
q_3 &= -\sin\left(\frac{\phi + \psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \\
q_4 &= \sin\left(\frac{\phi + \psi}{2}\right) \cos\left(\frac{\theta}{2}\right)
\end{aligned}$$

Given the rotation quaternion $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$, the x-convention 3-1-3 Euler angles (ϕ, θ, ψ) can be computed by

$$\begin{aligned}
\phi &= \arctan((q_1q_3 + q_2q_4), (q_2q_3 - q_1q_4)) \\
\theta &= \arccos(-q_1^2 - q_2^2 + q_3^2 + q_4^2) \\
\psi &= -\arctan((q_1q_3 - q_2q_4), (q_2q_3 + q_1q_4))
\end{aligned}$$

Euler axis/angle • quaternion

Given the Euler axis $\hat{\mathbf{e}}$ and angle θ , the quaternion

$$\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$$

can be computed by

$$q_1 = \hat{e}_1 \sin\left(\frac{\theta}{2}\right)$$

$$q_2 = \hat{e}_2 \sin\left(\frac{\theta}{2}\right)$$

$$q_3 = \hat{e}_3 \sin\left(\frac{\theta}{2}\right)$$

$$q_4 = \cos\left(\frac{\theta}{2}\right)$$

Given the rotation quaternion $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$, define $\tilde{\mathbf{q}} = [q_1 \ q_2 \ q_3]^T$. Then the Euler axis $\hat{\mathbf{e}}$ and angle θ can be computed by

$$\hat{\mathbf{e}} = \frac{\tilde{\mathbf{q}}}{\|\tilde{\mathbf{q}}\|}$$

$$\theta = 2 \arccos(q_4)$$

Conversion formulae between derivatives

Rotation matrix • angular velocities

The angular velocity vector $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ can be extracted from the derivative of the rotation matrix $\frac{d\mathbf{A}}{dt}$ by the following relation:

$$[\boldsymbol{\omega}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \frac{d\mathbf{A}}{dt} \mathbf{A}^T$$

The derivation is adapted from ^[5] as follows:

For any vector \mathbf{r}_0 consider $\mathbf{r}(t) = \mathbf{A}(t)\mathbf{r}_0$ and differentiate it:

$$\frac{d\mathbf{r}}{dt} = \frac{d\mathbf{A}}{dt} \mathbf{r}_0 = \frac{d\mathbf{A}}{dt} \mathbf{A}^T(t) \mathbf{r}(t)$$

The derivative of a vector is the linear velocity of its tip. Since \mathbf{A} is a rotation matrix, by definition the length of $\mathbf{r}(t)$ is always equal to the length of \mathbf{r}_0 , and hence it does not change with time. Thus, when $\mathbf{r}(t)$ rotates, its tip moves along a circle, and the linear velocity of its tip is tangential to the circle; i.e., always perpendicular to $\mathbf{r}(t)$. In this specific case, the relationship between the linear velocity vector and the angular velocity vector is

$$\frac{d\mathbf{r}}{dt} = \boldsymbol{\omega}(t) \times \mathbf{r}(t) = [\boldsymbol{\omega}]_{\times} \mathbf{r}(t) \text{ (see circular motion and Cross product).}$$

By the transitivity of the above mentioned equations,

$$\frac{d\mathbf{A}}{dt} \mathbf{A}^T(t) \mathbf{r}(t) = [\boldsymbol{\omega}]_{\times} \mathbf{r}(t)$$

which implies (Q.E.D.),

$$\frac{d\mathbf{A}}{dt} \mathbf{A}^T(t) = [\boldsymbol{\omega}]_{\times}$$

Quaternion • angular velocities

The angular velocity vector $\omega = (\omega_x, \omega_y, \omega_z)$ can be obtained from the derivative of the quaternion $\frac{d\mathbf{q}}{dt}$ as follows^[6]:

$$\begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = 2 \frac{d\mathbf{q}}{dt} \otimes \tilde{\mathbf{q}}$$

where $\tilde{\mathbf{q}}$ is the inverse of \mathbf{q} .

Conversely, the derivative of the quaternion is

$$\frac{d\mathbf{q}}{dt} = \frac{1}{2} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \otimes \mathbf{q}$$

References

- [1] Rotation Matrix, <http://mathworld.wolfram.com/RotationMatrix.html>
- [2] <http://mathworld.wolfram.com/Cayley-KleinParameters.html>
- [3] Direct and inverse kinematics (<http://cmp.felk.cvut.cz/cmp/courses/ROB/roblec/serial-noteeng.pdf>) lecture notes, page 5
- [4] Mebius, Johan (2007). "Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations". arXiv:math/0701759.
- [5] (<http://www.euclideanspace.com/physics/kinematics/angularvelocity/mark.htm>) Physics - Mark Ioffe - W(t) in terms of matrices
- [6] (<http://www.euclideanspace.com/physics/kinematics/angularvelocity/>) Physics - Kinematics - Angular Velocity
- , Evangelos A. Coutsias and Louis Romero, (1999) *The Quaternions with an application to Rigid Body Dynamics*, Department of Mathematics and Statistics, University of New Mexico.
- , F. Landis Markley, (2003) *Attitude Error Representations for Kalman Filtering*, Journal of Guidance, Control and Dynamics.
- , H. Goldstein, (1980) *Classical Mechanics*, 2nd. ed., Addison-Wesley. ISBN 0-201-02918-9
- , James R. Wertz, (1980) *Spacecraft Attitude Determination and Control*, D. Reidel Publishing Company. ISBN 90-277-1204-2
- , J. Schmidt and H. Niemann, (2001) *Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization*, Vision, Modeling and Visualization (VMV01).
- , Lev D. Landau and E. M. Lifshitz, (1976) *Mechanics*, 3rd. ed., Pergamon Press. ISBN 0-08-021022-8 (hardcover) and ISBN 0-08-029141-4 (softcover).
- , Klumpp, A. R., *Singularity-Free Extraction of a Quaternion from a Direction-Cosine Matrix*, Journal of Spacecraft and Rockets, vol. 13, Dec. 1976, p.754, 755.
- , C. Doran and A. Lasenby, (2003) *Geometric Algebra for Physicists*, Cambridge University Press. ISBN 978-0-521-71595-9

External links

- , EuclideanSpace (<http://www.euclideanspace.com/maths/geometry/rotations/index.htm>) has a wealth of information on rotation representation
- , Q36. How do I generate a rotation matrix from Euler angles? (http://www.j3d.org/matrix_faq/matrfaq_latest.html#Q36) and Q37. How do I convert a rotation matrix to Euler angles? (http://www.j3d.org/matrix_faq/matrfaq_latest.html#Q37) , The Matrix and Quaternions FAQ
- , Imaginary numbers are not Real € the Geometric Algebra of Spacetime (<http://www.mrao.cam.ac.uk/~clifford/introduction/intro/node9.html#SECTION00030000000000000000>) € Section "Rotations and Geometric Algebra" derives and applies the rotor description of rotations

- , Starlino's DCM Tutorial (http://www.starlino.com/dcm_tutorial.html) € Direction cosine matrix theory tutorial and applications. Space orientation estimation algorithm using accelerometer, gyroscope and magnetometer IMU devices. Using complimentary filter (popular alternative to Kalman filter) with DCM matrix.

Article Sources and Contributors

Rotation formalisms in three dimensions *Source:* <http://en.wikipedia.org/w/index.php?oldid=511199611> *Contributors:* Aesopos, AndrewDressel, Auntof6, Ballinamona, Ben pcc, BenFrantzDale, Calwiki, Catslash, Charles Matthews, Chessfan, Derek Andrews, EdJohnston, Emily Jensen, Eulerian vibe, Excirial, Fmalan, Giftlite, Hansee, Harold Philby, Haseldon, Headbomb, Ida Shaw, Imnotminkus, Incnis Mrsi, Iorsh, Jefago, Jemebius, Jheald, Jlenthe, JohnBlackburne, Jorunn, Juansempere, Keilandreas, KerichIII, Kmatzen, Laadams85, LilHelpa, Lliu2008, Lourakis, Mandarax, Michael Hardy, Mukkakukaku, Muphrid15, Nageh, Niac2, Otypon, Paolo.dL, Papadim.G, Parvons, Patrick O'Leary, Qetuth, Quondum, Radagast83, Riccio, Rjwilmsi, Ryankrameretc, SchreiberBike, Shlomo Levinger, Smutny, Soler97, Stamcose, Tennenrishin, The Anome, Thelastcraftman, Tkhaxton, Vliegeois, Waldir, Wesha, WillowW, Winston365, Yoderj, 101 anonymous edits

Image Sources, Licenses and Contributors

Image:Euler AxisAngle.png *Source:* http://en.wikipedia.org/w/index.php?title=File:Euler_AxisAngle.png *License:* Public Domain *Contributors:* DF Malan
Image:Praezession.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:Praezession.svg> *License:* GNU Free Documentation License *Contributors:* User Herbye (German Wikipedia).
Designed by Dr. H. Sulzer

License

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)