Introduction into quaternions for spacecraft attitude representation

Dipl. -Ing. Karsten Großekatthöfer, Dr. -Ing. Zizung Yoon
Technical University of Berlin
Department of Astronautics and Aeronautics
Berlin, Germany

May 31, 2012

Abstract

The purpose of this paper is to provide a straight-forward and practical introduction to quaternion operation and calculation for rigid-body attitude representation. Therefore the basic quaternion definition as well as transformation rules and conversion rules to or from other attitude representation parameters are summarized. The quaternion computation rules are supported by practical examples to make each step comprehensible.

1 Introduction

Quaternions are widely used as attitude represenation parameter of rigid bodies such as spacecrafts. This is due to the fact that quaternion inherently come along with some advantages such as no singularity and computationally less intense compared to other attitude parameters such as Euler angles or a direction cosine matrix. Mainly, quaternions are used to

- Parameterize a spacecraft's attitude with respect to reference coordinate system,
- Propagate the attitude from one moment to the next by integrating the spacecraft equations of motion,
- Perform a coordinate transformation: e.g. calculate a vector in body fixed frame from a (by measurement) known vector in inertial frame.

However, different references use several notations and rules to represent and handle attitude in terms of quaternions, which might be confusing for newcomers [5], [4]. Therefore this article gives a straight-forward and clearly notated introduction into the subject of quaternions for attitude representation.

The attitude of a spacecraft is its rotational orientation in space relative to a defined reference coordinate system.

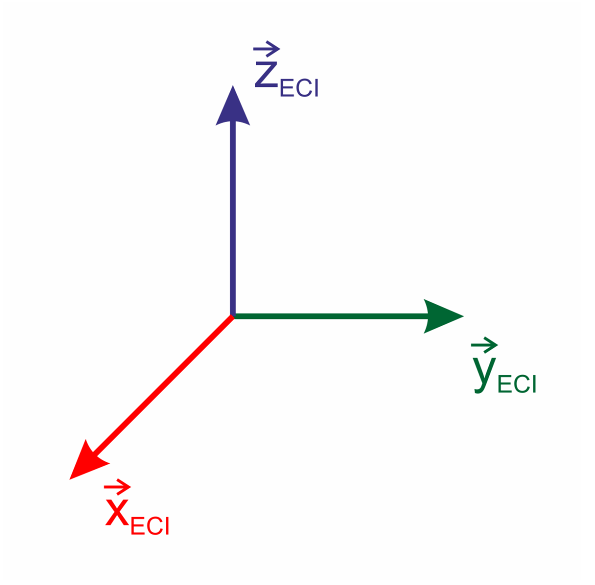

Figure 1: Reference coordinate system, e.g. Earth centered inertial (ECI) frame

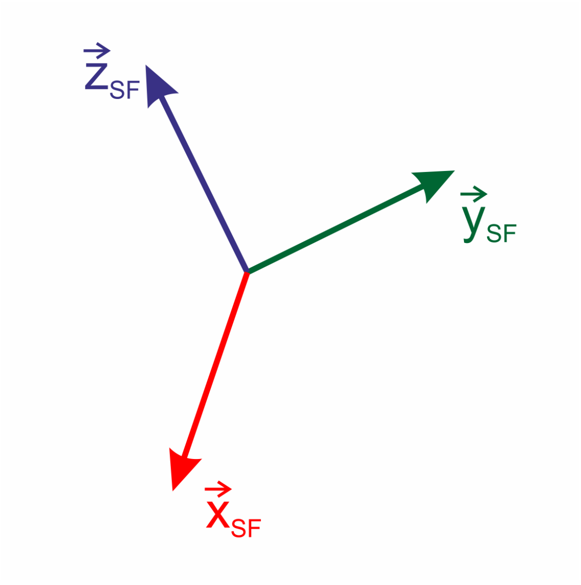

Figure 2: Spacecraft fixed (SF) coordinate system

2 Vector rotation vs. vector transformation

In the field of attitude representation, the terms *transformation* and *rotation* are often not clearly distinguished, which results in unexpected calculation results or wrong attitude information. This section will give a short definiton of both terms and illustrate the need for their differentiation in a simple example.

Rotation A rotation of a vector $\vec{v}$ in a coordinate system A is an operation, which modifies $\vec{v}$'s representation in A. Given the representation of A and $\vec{v}$ in a coordinate system S, a rotation changes the orientation of $\vec{v}$ in both A and S.

Transformation A (coordinate) transformation is an operation, which describes a vector $\vec{v}$'s representation with respect to a second coordinate system, B. Given the representation of A, B and $\vec{v}$ in a coordinate system S, a transformation does not change the orientation of $\vec{v}$ in S.

Example 1 *Figure 3 and Figure 4 show two coordinate systems, A and B. To get the base vectors of B, the base vectors of A must be rotated by 90° about the $\vec{x}_A$ axis.*

At first, we define a vector $\vec{v}_A$ in A (Figure 5) and apply a vector rotation by 90° about the $\vec{x}_A$ axis. The result of this rotation is called $\vec{w}_A$ and still expressed in A, as shown in Figure 6.

If $\vec{v}_A = \begin{pmatrix}1 & 2 & 3\end{pmatrix}^T$ in A, the rotated vector in A is $\vec{w}_A = \begin{pmatrix}1 & -3 & 2\end{pmatrix}^T$.

Now we transform both $\vec{v}_A$ and $\vec{w}_A$ to the coordinate system B, i.e. the orientation of the vectors in space stays the same, whereas the coordinate system describing them changes. The transformed vectors are called $\vec{v}_B$ and $\vec{w}_B$ respectively.

W.r.t. B, $\vec{v}_B = \begin{pmatrix}1 & 3 & -2\end{pmatrix}^T$ (Figure 7) and $\vec{w}_B = \begin{pmatrix}1 & 2 & 3\end{pmatrix}^T$ (Figure 8).

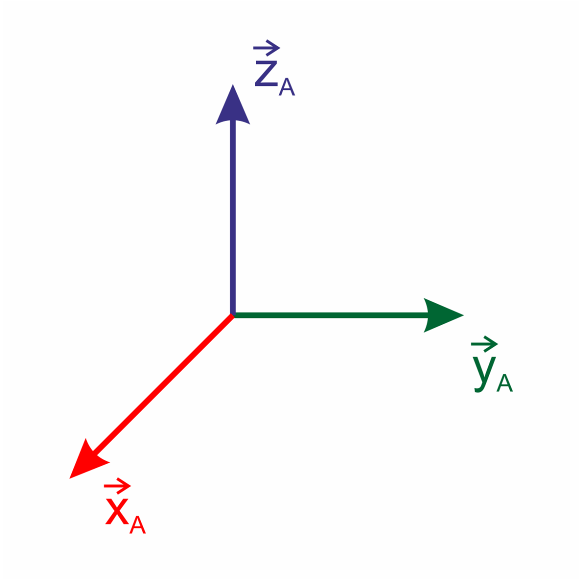

Figure 3: Coordinate system A

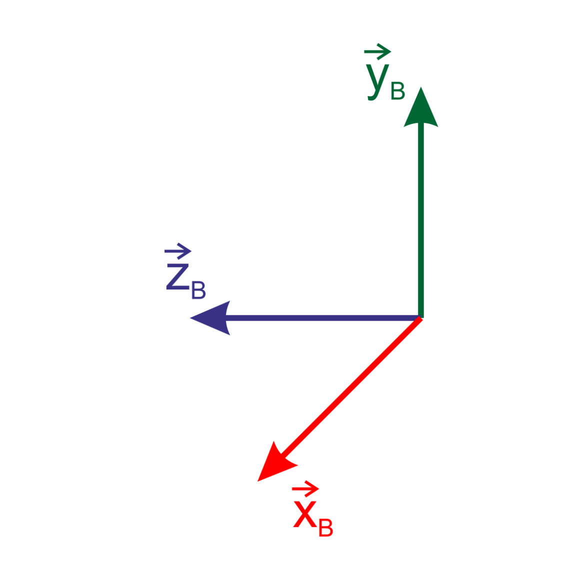

Figure 4: Coordinate system B

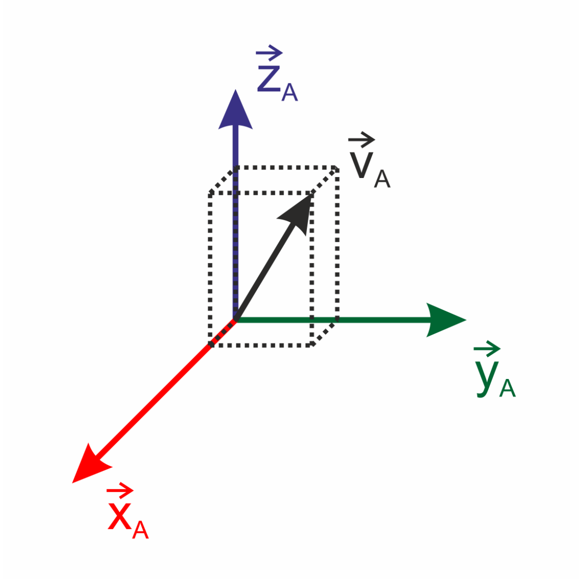

Figure 5: vector $\vec{v}_A$ in A

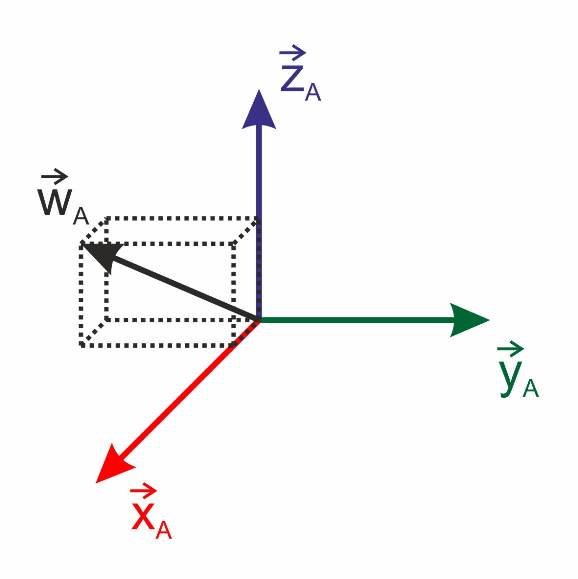

Figure 6: vector $\vec{w}_A$ in A

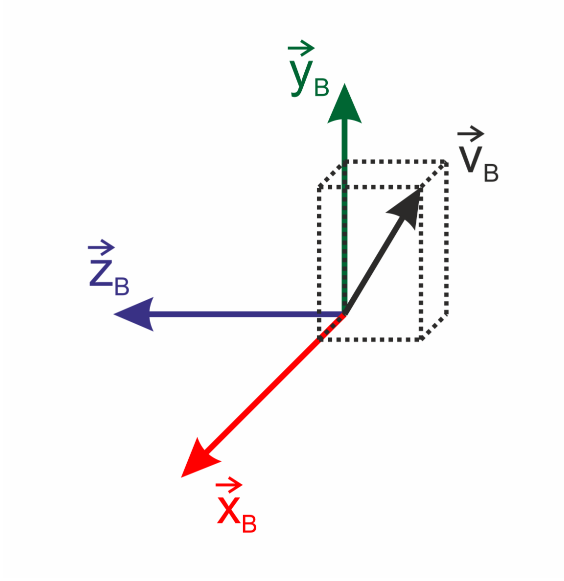

Figure 7: vector $\vec{v}_B$ in B

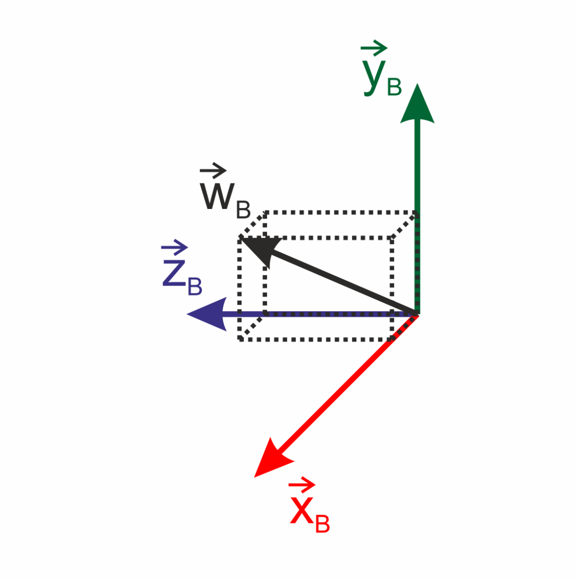

Figure 8: vector $\vec{w}_B$ in B

Hence, we have shown that a vector rotation (Figure 5 to Figure 6) is entirely different from a vector transformation (Figure 5 to Figure 7). Furthermore, we can see that transformation and rotation are inverse operations, i.e. the rotated vector $\vec{w}_B$ w.r.t. B has the same coordinates as the original vector $\vec{v}_A$ w.r.t. A.

The following section describes different types of representations for coordinate transformation. Mathematical operations for the transformation are given in Section 5.

3 Attitude representations

Three parameters are required to define a rotational orientation (attitude) of a rigid body in a three-dimensional Euclidean space. There exist various parametrization methods for the mathematical representation of a rigid body's attitude transformation or rotation. This section shall give an overview about the most common ones: direction cosine matrices (DCM), Euler angles and quaternions. Each methods bears advantages and drawbacks, which will be discussed within the following sections.

3.1 Direction cosine matrix

A direction cosine matrix is a transformation matrix which is composed of the direction cosine values between the initial coordinate system and the target coordinate system.

Let A be the initial coordinate system and B be the target coordinate system of a transformation. The base vectors of A are given by $\vec{x}_A$, $\vec{y}_A$ and $\vec{z}_A$, whereas $\vec{x}_B$, $\vec{y}_B$ and $\vec{z}_B$ are the base vectors of system B. The direction cosine matrix which transforms a vector from system A to system B shall be called $T_{B \leftarrow A}$ and is defined by

$$T_{B\leftarrow A} = \begin{pmatrix} cos\sphericalangle(\vec{x}_A, \vec{x}_B) & cos\sphericalangle(\vec{y}_A, \vec{x}_B) & cos\sphericalangle(\vec{z}_A, \vec{x}_B) \\ cos\sphericalangle(\vec{x}_A, \vec{y}_B) & cos\sphericalangle(\vec{y}_A, \vec{y}_B) & cos\sphericalangle(\vec{z}_A, \vec{y}_B) \\ cos\sphericalangle(\vec{x}_A, \vec{z}_B) & cos\sphericalangle(\vec{y}_A, \vec{z}_B) & cos\sphericalangle(\vec{z}_A, \vec{z}_B) \end{pmatrix} = \begin{pmatrix} \vec{x}_A \cdot \vec{x}_B & \vec{y}_A \cdot \vec{x}_B & \vec{z}_A \cdot \vec{x}_B \\ \vec{x}_A \cdot \vec{y}_B & \vec{y}_A \cdot \vec{y}_B & \vec{z}_A \cdot \vec{y}_B \\ \vec{x}_A \cdot \vec{z}_B & \vec{y}_A \cdot \vec{z}_B & \vec{z}_A \cdot \vec{z}_B \end{pmatrix} \tag{3.1}$$

The DCM $T_{A\leftarrow B}$ is an orthonormal matrix because the base vectors of A and B are orthogonal unit vectors. Therefore, the transpose of a DCM is the same as the DCM representing the inverse transformation. For all transformation matrices, the transpose is equal to the inverse of the matrix:

$$T^T_{B\leftarrow A} = T^{-1}_{B\leftarrow A} = T_{A\leftarrow B} = 1 \tag{3.2}$$

and

$$det(T_{A\leftarrow B}) = det(T_{B\leftarrow A}) \tag{3.3}$$

The transformation of a coordinate system about each basis vector with an rotation (not transformation!) angle θ is described by the following elementary transformation matrices:

$$R^x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \theta & sin\ \theta \\ 0 & -sin\ \theta & cos\ \theta \end{bmatrix} \quad R^y(\theta) = \begin{bmatrix} cos\ \theta & 0 & -sin\ \theta \\ 0 & 1 & 0 \\ sin\ \theta & 0 & cos\ \theta \end{bmatrix} \quad R^z(\theta) = \begin{bmatrix} cos\ \theta & sin\ \theta & 0 \\ -sin\ \theta & cos\ \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

3.2 Euler angles

The orientation of a rigid body with respect to an inertial coordinate system can be described by three successive transformations about the body fixed axis. The three angles used for the successive transformation are the Euler angles. Usually they are used for graphical display of the spacecraft orientation, since they are relatively easy to interprete. Any body fixed axis can be used for the first transformation. The second rotation must be performed by any of the two axes not taken for the first transformation. The final transformation is about any axes not employed by the second transformation. Therefore 12 different transformation sequences (sets of Euler angles) exist for this scheme to describe the attitude of a rigid body. The transformation matrix for the transformation sequences is obtained by the multiplication of three elementary transformation matrices.

Example 2 *The example describes the "$z \to x \to y$" transformation sequence to present the orientation of coordinate system B (basis vectors $\begin{bmatrix} \vec{x}_B & \vec{y}_B & \vec{z}_B \end{bmatrix}$) relative to A (basis vectors $\begin{bmatrix} \vec{x}_A & \vec{y}_A & \vec{z}_A \end{bmatrix}$).*

- *1st transformation about body fixed z-axis with angle θ_z : $R^z(\theta_z)$ transforms A to A'*
- *2nd transformation about body fixed x-axis with angle θ_x : $R^x(\theta_x)$ transforms A' to A''*
- *3rd transformation about body fixed y-axis with angle θ_y : $R^y(\theta_y)$ transforms A'' to B*

The three angles θ_x, θ_y, θ_z are called Euler angles. A' and A'' are intermediate coordinate systems with the basis vectors $\begin{bmatrix}\vec{x}_{A'} & \vec{y}_{A'} & \vec{z}_{A'}\end{bmatrix}$ and $\begin{bmatrix}\vec{x}_{A''} & \vec{y}_{A''} & \vec{z}_{A''}\end{bmatrix}$ respectively. The transformation sequences can be multiplied as following:

$$B = R^y(\theta_y) \cdot A'' = R^y(\theta_y) \cdot R^x(\theta_x) \cdot A' = R^y(\theta_y) \cdot R^x(\theta_x) \cdot R^z(\theta_z) \cdot A \tag{3.5}$$

Note that the matrices are sequentially multiplied from the left side, so that the first transformation matrix $R^z(\theta_z)$ comes last. Substituting Equation 3.4 into Equation 3.5 leads to

$$R^{zxy}_{B\leftarrow A} = \begin{bmatrix} -sin\theta_x sin\theta_y sin\theta_z + cos\theta_x cos\theta_z & sin\theta_x sin\theta_y cos\theta_z + cos\theta_x sin\theta_z & -sin\theta_x cos\theta_y \\ -cos\theta_y sin\theta_z & cos\theta_y cos\theta_z & sin\theta_y \\ cos\theta_x sin\theta_y sin\theta_z + sin\theta_x cos\theta_z & -cos\theta_x sin\theta_y cos\theta_z + sin\theta_x sin\theta_z & cos\theta_x cos\theta_y \end{bmatrix} \tag{3.6}$$

In the same manner, the 12 transformation matrix using Euler angles can be obtained. For smaller Euler angles the transformation matrix for all 12 rotation sequences can be linearized into one matrix:

$$R = \begin{bmatrix} 1 & \theta_z & -\theta_x \\ -\theta_z & 1 & \theta_y \\ \theta_x & -\theta_y & 1 \end{bmatrix} \tag{3.7}$$

which makes the transformation sequence insignificant for infinitesimal Euler angles.

3.3 Quaternions

The representation of relative orientation using Euler angles is easy to develop and to visualize, but computationally intense. Also a singularity problem occurs when describing attitude kinematics in terms of Euler angles and therefore it is not an effective method for spacecraft attitude dynamics. The widely used quaternion representation is based on Euler's rotational theorem which states that the relative orientation of two coordinate systems can be described by only one rotation about a fixed axis.

A Quaternion is a 4 × 1 matrix which elements consists of a scalar part s and a vector part $\vec{v}$. Note the scalar part is the first element of the matrix.

$$q = \begin{bmatrix} s \\ \vec{v} \end{bmatrix} = \begin{bmatrix} s \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{3.8}$$

As seen before, according to Euler's rotational theorem a quaternion is defined by a rotational axis and a rotation angle. A quaterionion respresenting a coordinate transformation from system A to system B, $q_{B\leftarrow A}$, is defined by Equation 3.9:

$$q = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ ||\vec{e}|| \cdot \sin\frac{\theta}{2} \end{bmatrix} \tag{3.9}$$

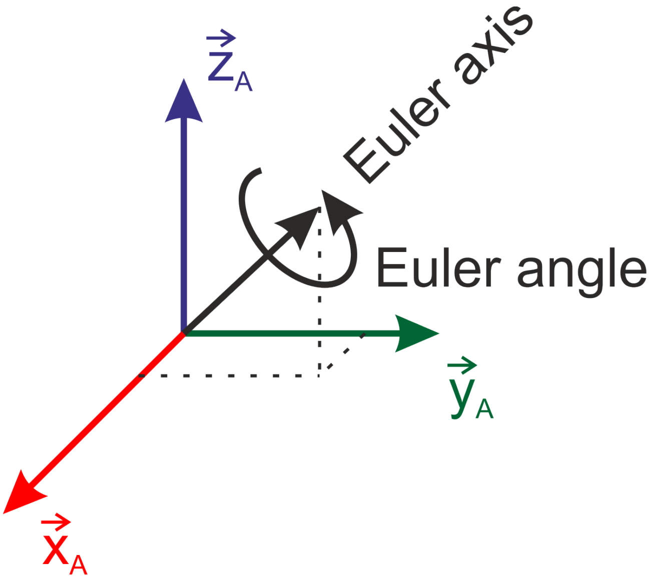

Figure 9: concept of Euler's rotational theoreom of a quaternion

where $\|\vec{e}\|$ is the normalized rotational axis and θ is not the rotational angle but the transformation angle.

Note: Every rotation or transformation can be expressed by two quaternions q and $\overline{q}$, where

$$\overline{q} = \begin{bmatrix} -q_s \\ -q_x \\ -q_y \\ -q_z \end{bmatrix} = \begin{bmatrix} \cos\frac{2\pi-\theta}{2} \\ \|-\vec{e}\| \cdot \sin\frac{2\pi-\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos(\pi - \frac{\theta}{2}) \\ \|-\vec{e}\| \cdot \sin(\pi - \frac{\theta}{2}) \end{bmatrix} \tag{3.10}$$

This becomes apparent, if we imagine $\overline{q}$ as a rotation (or transformation) with angle of $2\pi - \theta$ **and** opposite axis $-\vec{e}$ to q. Thus we have a 1:2 mapping for a relative orientation and its representation as a quaternion.

Example 3 *Let us assume in Figure 10 coordinate system TOD to be the inertial frame and SAT the body fixed frame. SAT is rotated by an angle θ about the axis $\vec{z}_{TOD} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ for an angle of θ=+30°. Thus the transformation angle is $-\theta$ respectively. The attitude of SAT with respect to TOD is given by the quaternion*

$$q_{SAT\leftarrow TOD} = \begin{bmatrix} \cos\frac{-\theta}{2} \\ \|\vec{z}_A\| \cdot \sin\frac{-\theta}{2} \end{bmatrix} = \begin{bmatrix} 0.9659 \\ 0 \\ 0 \\ -0.2588 \end{bmatrix} \tag{3.11}$$

Concerning the algebraic sign of θ, if a positive value describes the rotation of SAT's axes about $\vec{z}_{TOD}$, the transformation of a vector from TOD to SAT is computed via a negative value, as we have seen before that transformation and rotation are inverse to each other.

Mathematical operations to apply coordinate transformations for different attitude representations will be discussed in Section 5. To have the ability to do so, we must first take a closer look on quaternion mathematics, as done in the next section.

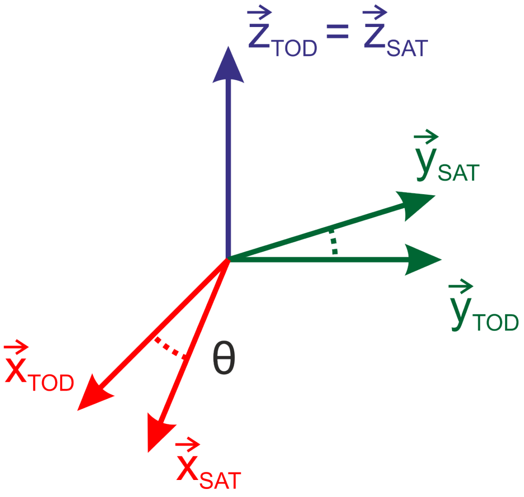

Figure 10: building of a transformation quaternion

Note: The notation of quaternion may differ among references and programming languages in terms of vector element sequence and angle sign. The quaternion definition of some widely used references are listet below:

- MATLAB Aerospace Toolbox:
  q $= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} \cos\frac{\theta}{2} \\ \|\vec{e}\| \cdot \sin\frac{\theta}{2} \end{bmatrix}$ where θ is not the transformation but the **rotation** angle.
- [5] is widely used as a standard reference on spacecraft dynamics.
  q $= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} \|\vec{e}\| \cdot \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2} \end{bmatrix}$ where θ is not the transformation but the **rotation** angle.
- [4] handles the attitude dynamics in a relatively practical point of view. The quaternion definition is identical to [5].

4 Quaternion Mathematics

Quaternions are a number system including a wide and complex mathematical theory. In this article, only computational rules required for the purpose of attitude representation are introduced briefly.

4.1 Norm

The norm of a quaternion is given by Equation 4.1:

$$|q| = \sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2} \tag{4.1}$$

A Quaternion with the norm $|q| = 1$ is called *unit quaternion.* All quaternions for attitude representation are unit quaternions.

4.2 Normalization

To normalize a quaternion, i.e. transform into a unit quaternion, it is devided by its norm:

$$\|q\| = \frac{q}{|q|} \tag{4.2}$$

4.3 Conjugate

The conjugate quaternion has an inverted vector part:

$$q^* = \begin{bmatrix} q_s \\ -q_x \\ -q_y \\ -q_z \end{bmatrix} \tag{4.3}$$

4.4 Inverse

To obtain the inverse of a quaternion, its conjugate is normalized.

$$q^{-1} = \frac{q^*}{|q|} \tag{4.4}$$

For all unit quaternions, the inverse is equal to the conjugate, as they have the norm one.

4.5 Multiplication

The product q of two quaternions q_1 and q_2 is defined by:

$$q_1 \otimes q_2 = q = \begin{bmatrix} s \\ \vec{v} \end{bmatrix} = \begin{bmatrix} s_1 \cdot s_2 - \vec{v}_1 \circ \vec{v}_2 \\ s_1 \cdot \vec{v}_2 + s_2 \cdot \vec{v}_1 + \vec{v}_1 \times \vec{v}_2 \end{bmatrix} \tag{4.5}$$

where $\circ$ is the dot product of vectors $\vec{v}_1$ and $\vec{v}_2$ and $\times$ is their cross product.

5 Mathematical operations of coordinate transformations

The transformation of a vector from one to another coordinate frame using the introduced attitude representation methods is given in the following.

5.1 Direction cosine matrix (DCM)

The transformation of a vector $\vec{v}$ for system A to system B via a direction cosine matrix $T_{B\leftarrow A}$ is simply achieved by

$$\vec{v}_B = T_{B\leftarrow A} \cdot \vec{v}_A \tag{5.1}$$

Example 4 *We come back to the coordinate systems A and B from Example 1. As defined before, to get the base vectors of B, the base vectors of A must be rotated by 90° about the* $\vec{x}_A$ *axis. We determine the DCM according to Equation 3.1:*

$$T_{B\leftarrow A} = \begin{bmatrix} cos\ 0^\circ & cos\ 90^\circ & cos\ 90^\circ \\ cos\ 90^\circ & cos\ 90^\circ & cos\ 0^\circ \\ cos\ 90^\circ & cos\ 180^\circ & cos\ 90^\circ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{5.2}$$

Now we transform the vector $\vec{v}_A$ *from system A to system B via Equation 5.1 and obtain the result* $\vec{v}_B$*:*

$$\vec{v}_B = T_{B\leftarrow A} \cdot \vec{v}_A = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{5.3}$$

5.2 Euler angles

Analog to the vector transformation using DCM given in Section 5.1, the transformation of a vector from A to B using the Euler Angles is done by

$$\vec{v}_B = R^{seq}_{B\leftarrow A} \cdot \vec{v}_A \tag{5.4}$$

where seq is one of the twelve possible transformation sequences.

Example 5 *Now lets do the same vector transformation done in Example 4 using Euler angles. Because there is only a single rotation sequence around the x-Axis, the matrix*

$$R^{seq}_{B\leftarrow A} = R^{x}_{B\leftarrow A} = R^x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \theta & sin\ \theta \\ 0 & -sin\ \theta & cos\ \theta \end{bmatrix},$$

as given in Equation 3.4. Therefore, for $\theta = 90^\circ$*,*

$$\vec{v}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \theta & sin\ \theta \\ 0 & -sin\ \theta & cos\ \theta \end{bmatrix} \cdot \vec{v}_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}.$$

5.3 Quaternions

In terms of quaternions, the transformation of $\vec{v}$ from A to B is defined as:

$$\vec{v}_B = q_{B\leftarrow A} \otimes \begin{bmatrix} 0 \\ \vec{v}_A \end{bmatrix} \otimes q_{B\leftarrow A}^{-1} \tag{5.5}$$

where $\otimes$ is the operator of a quaternion multiplication (cf. Equation 4.5).

Example 6 *For our systems A and B, we first of all determine the transformation quaternion using Equation 3.9:*

$$q_{B\leftarrow A} = \begin{bmatrix} \cos\frac{-90^\circ}{2} \\ \|\vec{x}_A\| \cdot \sin\frac{-90^\circ}{2} \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} \\ 0 \\ 0 \end{bmatrix} \tag{5.6}$$

Applying Equation 5.5 results in $\vec{v}_B$:

$$\vec{v}_B = q_{B\leftarrow A} \otimes \left(\begin{bmatrix} 0 \\ \vec{v}_A \end{bmatrix} \otimes q_{B\leftarrow A}^{-1} \right) = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} \\ 0 \\ 0 \end{bmatrix} \otimes \left(\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \otimes \begin{bmatrix} \sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} \\ 0 \\ 0 \end{bmatrix} \right) \tag{5.7}$$

Note: *The quaternion build with an angle of $\theta = +90^\circ$ can either be regarded as the transformation quaternion from system B to system A, $q_{A\leftarrow B}$, or a rotation quaternion q_R which rotates a vector within A. By inserting q_R in Equation 5.5, we obtain the rotated vector $\vec{w}_A$ w.r.t A, which is also known from the example in Section 2.*

6 Conversion of attitude representations

Since there is no "standard" representation method, but the method is chosen depending on application, the conversion among each other is necessary. Hence this chapter deals with the conversion of the introduced representation methods.

6.1 DCM to quaternion

This again seems to be from the reverse perspective/transformation

The single elements of a quaternion can be calculated from the main diagonal of the corresponding direction cosine matrix DCM. In case an element is found which is not zero, all other elements can be calculated from this element and the sub-diagonals. As elements close to zero can cause inexact solutions, one can first calculate all four elements from the main diagonal and then take the greatest value as basis for further calculation. As all equations have a square root on the right side, the maximum value is also the maximum absolute value.

$$
\begin{aligned}
q_s &= \sqrt{\frac{1}{4} \cdot (1 + T_{11} + T_{22} + T_{33})} \\
q_x &= \sqrt{\frac{1}{4} \cdot (1 + T_{11} - T_{22} - T_{33})} \\
q_y &= \sqrt{\frac{1}{4} \cdot (1 - T_{11} + T_{22} - T_{33})} \\
q_z &= \sqrt{\frac{1}{4} \cdot (1 - T_{11} - T_{22} + T_{33})}
\end{aligned}
\tag{6.1}
$$

The maximum value from Equation 6.1 is chosen directly and all others are re-calculated as shown in Table 6.1. The first column lists all values from Equation 6.1, while the columns two to five list the re-caluclated values for all other elements.

Table 1: Building quaternion elements from subdiagonals

maximum	q_s	q_x	q_y	q_z
q_s	q_s	$\frac{T_{32}-T_{23}}{4 \cdot q_s}$	$\frac{T_{13}-T_{31}}{4 \cdot q_s}$	$\frac{T_{21}-T_{12}}{4 \cdot q_s}$
q_x	$\frac{T_{32}-T_{23}}{4 \cdot q_x}$	q_x	$\frac{T_{21}+T_{12}}{4 \cdot q_x}$	$\frac{T_{13}+T_{31}}{4 \cdot q_x}$
q_y	$\frac{T_{13}-T_{31}}{4 \cdot q_y}$	$\frac{T_{21}+T_{12}}{4 \cdot q_y}$	q_y	$\frac{T_{32}+T_{23}}{4 \cdot q_y}$
q_z	$\frac{T_{21}-T_{12}}{4 \cdot q_z}$	$\frac{T_{13}+T_{31}}{4 \cdot q_z}$	$\frac{T_{32}+T_{23}}{4 \cdot q_z}$	q_z

Example 7 *To convert the DCM from Example 4,*

$$
T_{B \leftarrow A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}
$$

to a quaternion representation, we first use Equation 6.1 to find the maximum element:

$$
\begin{aligned}
q_s &= \sqrt{\frac{1}{4} \cdot (1 + 1 + 0 + 0)} = \sqrt{\frac{1}{2}} \\
q_x &= \sqrt{\frac{1}{4} \cdot (1 + 1 - 0 - 0)} = \sqrt{\frac{1}{2}} \\
q_y &= \sqrt{\frac{1}{4} \cdot (1 - 1 + 0 - 0)} = 0 \\
q_z &= \sqrt{\frac{1}{4} \cdot (1 - 1 - 0 + 0)} = 0
\end{aligned}
$$

For the next step, we have to pick the element with the greatest absolute value. In this case, *both q_s and q_x have the value $\sqrt{\frac{1}{2}}$. We take q_s and apply the formulas of Table 6.1, row one:*

$$\begin{aligned}
q_x &= \frac{T_{32} - T_{23}}{4 \cdot q_s} = \frac{-1-1}{4 \cdot \sqrt{\frac{1}{2}}} = -\sqrt{\frac{1}{2}} \\
q_y &= \frac{T_{13} - T_{31}}{4 \cdot q_s} = \frac{0-0}{4 \cdot \sqrt{\frac{1}{2}}} = 0 \\
q_z &= \frac{T_{21} - T_{12}}{4 \cdot q_s} = \frac{0-0}{4 \cdot \sqrt{\frac{1}{2}}} = 0
\end{aligned}$$

Finally, we build the quaternion from these four values:

$$q_{B \leftarrow A} = \begin{bmatrix} \sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} \\ 0 \\ 0 \end{bmatrix}$$

***Note:** If we had picked q_x instead of q_s, the resulting quaternion would have been*

$$\overline{q}_{B \leftarrow A} = \begin{bmatrix} -\sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} \\ 0 \\ 0 \end{bmatrix}$$

Both quaternions represent the same relative orientation, so this example also shows the 1:2 mapping which results from Equation 3.10.

6.2 Quaternion to DCM

Note this appears to be C_v/b, this is the transpose of Matlab which gives C_b/v

The conversion from a transformation quaternion q to a direction cosine matrix T is given in Equation 6.2:

$$T = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2 \cdot (q_x \cdot q_y - q_z \cdot q_s) & 2 \cdot (q_x \cdot q_z + q_y \cdot q_s) \\ 2 \cdot (q_x \cdot q_y + q_z \cdot q_s) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2 \cdot (q_y \cdot q_z - q_x \cdot q_s) \\ 2 \cdot (q_x \cdot q_z - q_y \cdot q_s) & 2 \cdot (q_y \cdot q_z + q_x \cdot q_s) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{6.2}$$

6.3 Euler to DCM

To convert Euler angles into a DCM, one has to consider the sequence of the transformation. As pointed out in Section 3.2, there are 12 different possibilities.

Example 8 *For the sequence used in Example 2, we can use Equation 3.6 for the conversion:*

$$T = \begin{bmatrix} -sin\theta_x sin\theta_y sin\theta_z + cos\theta_x cos\theta_z & sin\theta_x sin\theta_y cos\theta_z + cos\theta_x sin\theta_z & -sin\theta_x cos\theta_y \\ -cos\theta_y sin\theta_z & cos\theta_y cos\theta_z & sin\theta_y \\ cos\theta_x sin\theta_y sin\theta_z + sin\theta_x cos\theta_z & -cos\theta_x sin\theta_y cos\theta_z + sin\theta_x sin\theta_z & cos\theta_x cos\theta_y \end{bmatrix} \tag{6.3}$$

6.4 DCM to Euler

Given an equation matrix to convert Euler angles to a DCM, like Equation 6.3, we can derive equations to calculate the Euler angles for this sequence.

Example 9 *Equation 6.3 shows the conversion from Euler angles to a DCM for the transformation sequence "$z \to x \to y$". Now we form new equations which allow us to calculate Euler angles from any given direction cosine matrix T. We devide T_{23} by T_{23} to get θ_z, deviding T_{13} by T_{33}, we obtain θ_x, and finally, element T_{23} provides θ_y.*

$$\begin{aligned} \theta_z &= \tan^{-1}(-\frac{T_{21}}{T_{22}}) && (6.4) \\ \theta_x &= \tan^{-1}(-\frac{T_{13}}{T_{33}}) && (6.5) \\ \theta_y &= \sin^{-1}(-T_{23}) && (6.6) \end{aligned}$$

6.5 Euler to Quaternion

To obtain a quaternion from Euler angles, one can take an intermediate step and convert into a DCM first (Section 6.3). The elements of the transformation matrix are then used to compute the quaternion elements as given in Section 6.1.

6.6 Quaternion to Euler

Converting a quaternion to Euler angles, we use the inverse procedure of Section 6.5. So we convert into a DCM first (Section 6.2) and then calculate the Euler angles using equations similiar to Equation 6.6.

7 Application

This chapter describes (or refers to) some commonly used quaternion operations / applications for attitude determination and control.

7.1 Attitude kinematics

A spacecraft's kinematics equation gives the dependency of the time derivative of its relative orientation in space from the angular rate, as shown in Equation 7.2. Note that the matrix Ω depends on the quaternion notation, which may differ in literature. Here, the notation given by Equation 3.8 is used.

$$
\begin{aligned}
\dot{q}_{TOD\leftarrow MOI} &= \frac{1}{2}\cdot\Omega\cdot q_{TOD\leftarrow MOI} \\
&= \frac{1}{2}\cdot\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}\cdot q_{TOD\leftarrow MOI} \\
&= \frac{1}{2}\cdot q_{TOD\leftarrow MOI}\otimes\begin{bmatrix} 0 \\ \vec{\omega} \end{bmatrix}
\end{aligned}
\tag{7.1}
$$

The satellite's attitude can then be determined by integrating $\dot{q}_{TOD\leftarrow MOI}$.

7.2 Attitude determination

Since the attitude determination problem is extensively investigated in numerous papers, the references are given in the list respectively. Since the references may use different quaternion definition, be aware of the notation.

- TRIAD: Simple solution to compute spacecraft attitude from vector measurement. [1]
- Q-Method: The Q-Method provides a closed solution to obtain the optimal attitude by minimizing the error function from multiple vector measurement. [3]
- QUEST: The QUEST provides an approximate but fast solution of the attitude by minimizing the error function from multiple vector measurement. [2]

7.3 Consecutive transformation

Simililar to multiplying the Euler matrices for successive transformation, the quaternions can be easily multiplied to achieve consecutive transformation of the attitude.

Example 10 *If the attitude is known as $q_{SAT\leftarrow TOD}$ and the transformation between the earth centered earth fixed WGS84 frame and the inertial frame is given by the onboard computer as $q_{WGS\leftarrow TOD}$, the attitude of the SAT frame w.r.t the WGS is obtained by:*
$q_{WGS\leftarrow SAT} = q_{WGS\leftarrow TOD}\otimes q^{-1}_{SAT\leftarrow TOD} = q_{WGS\leftarrow TOD}\otimes q_{TOD\leftarrow SAT}$

References

[1] H. Black. A Passive System for Determining the Attitude of a Satellite. *AIAA Journal 2 (7): 1350–1351*, 1963.

[2] M. Shuster. The Quest for Better Attitudes. *The Journal of the Astronautical Sciences, Vol. 54, Nos. 3&4, July-December*, 54:657–683, 2006.

[3] M. D. Shuster and S. D. Oh. Three-Axis Attitude Determination from Vector Observations. *Journal of Guidance and Control*, 4:70–77, 1981.

[4] M. Sidi. *Spacecraft Dynamcis And Control: a practical engineering approach.* Cambridge University Press, 1997.

[5] J. Wertz. *Spacecraft Attitude Determination and Control.* Springer Netherlands, 1987.