JSC-61777-models/utils/quaternion

JSC Engineering Orbital Dynamics Quaternion Model

Simulation and Graphics Branch (ER7)
Software, Robotics, and Simulation Division
Engineering Directorate

Package Release JEOD v5.2

Document Revision 1.0
September 2024

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

JSC-61777-models/utils/quaternion

JSC Engineering Orbital Dynamics
Quaternion Model

Document Revision 1.0
September 2024

David Hammen

Simulation and Graphics Branch (ER7)
Software, Robotics, and Simulation Division
Engineering Directorate

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

Executive Summary

Sir William Rowan Hamilton developed a four dimensional extension to the complex numbers called quaternions in the mid 1800s. Like the reals and the complex numbers, quaternions can be added, subtracted, multiplied, and divided. Unlike the reals and complex numbers, quaternion multiplication and division is not commutative. Care must be taken in specifying the order of the factors when multiplying quaternions.

Quaternions, like complex numbers, can be represented as comprising real imaginary parts. In the case of quaternions, there are three distinct roots of -1, i, j, and k. The imaginary part of a quaternion can be viewed as a three vector. (In fact, the use of $\hat{i}$, $\hat{j}$, and $\hat{k}$ as canonical unit vectors in mathematics and physics was motivated by Hamilton's quaternions.)

One widely used application of the quaternions is to represent rotations and transformations in three dimensional space. That application is the subject of the JEOD Quaternion model. The full power of the quaternions is not needed for this application. Unit quaternions (quaternions whose magnitude is identically one) suffice for representing rotations in Euclidean three-space. The principal operations of interest for this use of quaternions are

- Data representation. Quaternions must be represented by some means to enable their use. The JEOD quaternion model represents quaternions as comprising a scalar real part and an imaginary vector part.
- Multiplication. A sequence of rotations or transformations in three space maps to a product of quaternion representations of those rotations / transformations.
- Normalization. The quaternions used to represent rotations / transformations in three space should be unit quaternions. Normalizing a quaternion makes it a unit quaternion.
- Conjugation. One advantage of using unit quaternions is that the inverse is particularly easy to compute: It is the quaternion's conjugate.
- Conversion. Quaternions are but one of many ways to represent rotations and transformations in three space. Conversions between quaternions and other schemes can be quite useful.

Contents

Chapter 1

Introduction

1.1 Purpose and Objectives of the Quaternion Model

Sir William Rowan Hamilton developed a four dimensional extension to the complex numbers called quaternions in the mid 1800s. Like the reals and the complex numbers, quaternions can be added, subtracted, multiplied, and divided. Unlike the reals and complex numbers, quaternion multiplication and division is not commutative. Care must be taken in specifying the order of the factors when multiplying quaternions.

Quaternions, like complex numbers, can be represented as comprising real imaginary parts. In the case of quaternions, there are three distinct roots of -1, i, j, and k. The imaginary part of a quaternion can be viewed as a three vector. (In fact, the use of $\hat{i}$, $\hat{j}$, and $\hat{k}$ as canonical unit vectors in mathematics and physics was motivated by Hamilton's quaternions.)

One widely used application of the quaternions is to represent rotations and transformations in Euclidean three dimensional space. That application is the subject of the Quaternion Model.

1.2 Context within JEOD

The following document is parent to this document:

- *JSC Engineering Orbital Dynamics* [4]

The Quaternion Model forms a component of the utilities suite of models within JEOD v5.2. It is located at models/utils/quaternion.

1.3 Documentation History

Author	Date	Revision	Description
David Hammen	December, 2009	1.0	Initial Version

Quaternions are used throughout JEOD. These uses include:

- The *Reference Frame Model* [7], which uses left transformation unit quaternions (along with transformation matrices) to represent the rotational relationships between reference frames.
- The *Mass Body Model* [8], which uses left transformation unit quaternions (along with transformation matrices) to represent the rotational relationships between mass bodies.
- The *Dynamic Body Model* [3], which uses quaternionic propagation to model the time evolution of a dynamic body's rotational state.
- The *Orientation Model* [9], which uses left transformation unit quaternions as one of several mechanisms by which the user can specify a rotational relationship.

1.4 Documentation Organization

This document is formatted in accordance with the NASA Software Engineering Requirements Standard [6].

The document comprises chapters organized as follows:

Chapter 1: Introduction - This introduction describes the objective and purpose of the Quaternion Model.

Product Requirements - The requirements chapter describes the requirements on the Quaternion Model.

Chapter 3: Product Specification - The specification chapter describes the architecture and design of the Quaternion Model.

Chapter 4: User Guide - The user guide chapter describes how to use the Quaternion Model.

Chapter 5: Verification and Validation - The inspection, verification, and validation (IV&V) chapter describes the verification and validation procedures and results for the Quaternion Model.

Chapter 2

Product Requirements

Requirement Quaternion_1: Project Requirements

Requirement:

This model shall meet the JEOD project-wide requirements specified in the JEOD v5.2 top-level document requirements[4].

Rationale:

This is a project-wide requirement.

Verification:

Inspection

Requirement Quaternion_2: Quaternion Encapsulation

Requirement:

The Quaternion model shall encapsulate the data items needed to represent a quaternion and the mathematical operations on a quaternion in the form of an instantiable class.

Rationale:

JEOD 2.0 is written in C++. A class is the standard encapsulation mechanism used throughout JEOD. The overarching purpose of the Quaternion model is to represent quaternions as a class.

Verification:

Inspection

Requirement Quaternion_3: Data Representation

Requirement:

The Quaternion class shall contain the data items needed to represent a quaternion as an object.

Rationale:
Self-evident.

Verification:
Inspection

Requirement Quaternion_4: Represent Orientation

Requirement:
The Quaternion class shall provide the mathematical operations needed to enable the use of quaternions to represent the orientation of objects and reference frames in three-space.

Rationale:
The ability to represent orientation of objects and reference frames is the key reason for the existence of the Quaternion model.

Verification:
Inspection, test

Requirement Quaternion_5: Conversion

Requirement:
The Quaternion model shall provide means to convert left transformation unit quaternions to and from

5.1 Column vector transformation matrices ,

5.2 eigen rotations , and

5.3 four vectors .

Rationale:

- Conversion to and from transformation matrices is an essential capability used by many other JEOD models.
- There are many other charts on SO(3). Eigen rotations is one of them.
- A quaternion needs to be converted to a primitive type to enable communication with outside agents and simulations.

Verification:
Inspection, test

Chapter 3

Product Specification

3.1 Conceptual Design

3.1.1 Basic Concepts

The JEOD Quaternion class represents quaternions as comprising a real scalar and an imaginary vectorial part, thereby satisfying requirements Quaternion_2 and Quaternion_3. The Quaternion class defines methods needed to satisfy requirements Quaternion_4 and Quaternion_5.

3.1.2 Trades

Data Representation

Quaternions can be represented by a variety of data representation schemes. These include:

- A scalar real part and imaginary vectorial part. This approach explicitly acknowledges that quaternions are an extension of the complex numbers.
- Four separate scalars, typically labeled w, x, y, and z. This approach has the advantage of simplicity. The key problem with this approach is that the real and imaginary parts are not well- distinguished.
- A four vector. This approach has the advantage of encapsulating the data in a single item that can easily be shared with other simulations, including those that do not use the JEOD Quaternion class. The key problem with this approach is that it is not clear which element denotes the real part of a quaternion. Some place the real part as the first (zeroth) element of the four vector while others place it as the last element.
- A unit vector and an angle. This approach has the advantage of ease of visualization. The key problem with this approach is one must invoke trigonometric functions to compute the product of two quaternions.

The JEOD Quaternion class represents quaternions as comprising a scalar real part and imaginary vectorial part. As noted above, this approach explicitly acknowledges that quaternions are an extension of the complex numbers.

Use of Quaternions to Represent Orientation

There are two coin-toss types of decisions to be made in using quaternions to represent orientation in three-space:

- Rotation versus transformation. Rotation is the conjugate of transformation. JEOD uses quaternions to represent transformations rather than rotations.
- Left versus right quaternions. Any quaternion can be used to represent a rotation or transformation of a three vector via either $\begin{bmatrix}0\\x'\end{bmatrix} = Q \begin{bmatrix}0\\x\end{bmatrix} Q^*$ or $\begin{bmatrix}0\\x'\end{bmatrix} = Q^* \begin{bmatrix}0\\x\end{bmatrix} Q$. The only difference between these two forms is whether the quaternion appears in an unconjugated form to the left or right of the vector to be rotated / transformed. Both forms are mathematically correct.

In addition to the above concerns is the issue of using quaternions in general versus unit quaternions to represent transformations. Taking the conjugate of a quaternion in general requires negating the quaternion's imaginary part and scaling by the inverse of the quaternion's magnitude. Because the magnitude of a unit quaternion is one by definition, the conjugate of a unit quaternion is easily constructed by negating the imaginary part of the quaternion. Converting a unit quaternion to the equivalent transformation matrix is also much simpler if the quaternion is a unit quaternion.

The JEOD Quaternion class uses left unit transformation quaternions.

3.1.3 Functionality

The minimal functionality needed to encapsulate quaternions as an object is the ability to construct a quaternion. The JEOD Quaternion class provides a default constructor that initializes a quaternion to all zeros. The class also provides convenience constructors whose functionality could be achieved by using the default constructor in combination with one of the conversion methods.

The functionality needed to enable use of quaternions to represent the orientation of objects and reference frames in three-space includes:

- Normalization,
- Multiplying by a real scalar,
- Multiplying by a vector (interpreted as a pure imaginary quaternion), and
- Multiplying by another quaternion, in various forms.

The JEOD Quaternion class provides methods to convert to/from:

- Transformation matrices;

- Eigen rotations; and
- Four vector representation of quaternions, with the scalar part being the zeroth element of the four vector.

Future versions of the Quaternion package may provide additional conversion capabilities.

3.2 Mathematical Formulation

This section summarizes key equations used in the implementation of the Quaternion Model. See appendix A for a detailed description of the quaternions of the mathematical formulation of the quaternions.

3.2.1 Quaternion Representation

The Quaternion Model represents quaternions as comprising a real scalar and an imaginary vectorial part:

$$\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \tag{3.1}$$

See section A.2.2 for details.

3.2.2 Quaternion Multiplication

The product of two quaternions $\mathcal{Q}_1$ and $\mathcal{Q}_2$,

$$\mathcal{Q}_1 = \begin{bmatrix} q_{1_s} \\ \boldsymbol{q_{1_v}} \end{bmatrix}$$

$$\mathcal{Q}_2 = \begin{bmatrix} q_{1_2} \\ \boldsymbol{q_{1_2}} \end{bmatrix}$$

is

$$\mathcal{Q}_1\,\mathcal{Q}_2 = \begin{bmatrix} q_{1_s}q_{2_s} - \boldsymbol{q_{1_v}} \cdot \boldsymbol{q_{2_v}} \\ q_{1_s}\boldsymbol{q_{2_v}} + q_{2_s}\boldsymbol{q_{1_v}} + \boldsymbol{q_{1_v}} \times \boldsymbol{q_{2_v}} \end{bmatrix} \tag{3.2}$$

See section A.3 for details.

3.2.3 Quaternion Norm

The norm, or magnitude of a quaternion $\mathcal{Q}$,

$$\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$$

is given by

$$||\mathcal{Q}||^2 = {q_s}^2 + {q_v}^2 \tag{3.3}$$

See section A.3 for details.

3.2.4 Quaternion Normalization

The normalized quaternion based on a quaternion $\mathcal{Q}$,

$$\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$$

is formed by scaling the quaternion by the inverse of the quaternion's norm:

$$\mathcal{Q}_{\text{normalized}} = \frac{1}{||\mathcal{Q}||}\mathcal{Q} \tag{3.4}$$

Suppose the norm of the quaternion in question is close to one: $||\mathcal{Q}||^2 = 1 + \epsilon$, where $|\epsilon| \lll 1$. The approximation

$$\frac{1}{||\mathcal{Q}||} \approx \frac{2}{1 + ||\mathcal{Q}||^2} \tag{3.5}$$

is computationally much less expensive than the square root function but is just as accurate as is the square root formulation for small ϵ.

JEOD uses equation 3.5 in lieu of equation 3.4 when $|\epsilon| < 2.107342 * \times 10^{-8}$.

3.2.5 Quaternion Derivative

The time derivative of left transformation quaternion from frame A to frame B, $\mathcal{Q}_{A \to B}$, is

$$\dot{\mathcal{Q}}_{A \to B} = \begin{bmatrix} 0 \\ -\frac{1}{2}\boldsymbol{\omega}_{B:A \to B} \end{bmatrix} \mathcal{Q}_{A \to B} \tag{3.6}$$

See section A.6 for details.

3.3 Detailed Design

The Quaternion model is located in the JEOD models directory utils/quaternions. One class, the Quaternion class, implements the Quaternion model. The Quaternion model API is described in [1].

Chapter 4

User Guide

This chapter describes how to use the Quaternion model from the perspective of a simulation user, a simulation developer, and a model developer.

4.1 Analysis

The Quaternion class is not intended to be used directly at the S_define level. Several JEOD models do however contain Quaternion objects as data elements. Whether these elements can be set in the input or logged to some log file depends on the object.

As the Quaternion data elements comprise the scalar and vector parts of the quaternion, the standard Trick forms for setting / logging a scalar and a three-vector double variable can be used to initialize / log a Quaternion element that embedded inside some larger data structure.

One very important caveat regarding the initialization and analysis of a quaternion: The user must be cognizant that different people use quaternions in different ways represent the orientation of an object or reference frame. In addition to the obvious sense issue (e.g., body to inertial versus inertial to body), some use quaternions to represent rotation while others use quaternions to represent transformations, and some use left quaternions while others use right quaternions. The JEOD Quaternion model uses left unit transformation quaternions.

Because multiple representation schemes exist, a quaternion provided by some external source may need to be converted to the left unit transformation quaternion form used by the JEOD Quaternion model. This is a simple operation: negate the imaginary (vectorial) part of the quaternion. For example, a quaternion received from an external source that uses right unit transformation quaternions will need to be conjugated before it can be used within JEOD.

4.2 Integration

The Quaternion class is not intended to be used directly at the S_define level. Several JEOD models do however contain Quaternion objects as data elements. The simulation developer should understand how quaternions are used in these various models.

The ability to convert a quaternion to an eigen rotation provides a useful capability for the analysis of quaternions used to represent rotations. Suppose two alternate expressions such as the true versus navigated values of a quaternion are available. Analysts will want to know whether these quaternions differ significantly. Designating the two quaternions as Q1 and Q2, if the two were exactly equal, the product Q1 Q2* would be the pure real unit quaternion. An eigen decomposition of this product will yield an eigen angle and eigen unit vector. The eigen angle is very useful in assessing the difference between the quaternions. Embedding the member function eigen_compare in the S_define file can provide this critical analytical capability.

4.3 Extension

The JEOD 2.0 Quaternion model lacks the full functionality described in Appendix A. One obvious extension to the Quaternion class is to provide this full set of capabilities. Other missing capabilities include:

- Support for right transformation quaternions, and
- Support for additional charts on SO(3) such as Euler angles, Rodrigues parameters, and modified Rodrigues parameters.

The primary purpose of the Quaternion class is to be embedded as data element within some other class. The JEOD Orientation, Reference Frame, and Mass models exemplify this use. A model developer who wishes to make such a use of the JEOD Quaternion model is encouraged to look at these other models as a template.

Chapter 5

Verification and Validation

5.1 Inspection

5.2 Verification

5.3 Validation

5.4 Metrics

Table 5.1 presents coarse metrics on the source files that comprise the model.

Table 5.1: Coarse Metrics

	Number of Lines			
File Name	**Blank**	**Comment**	**Code**	**Total**
include/quat.hh	50	116	67	233
include/quat_inline.hh	67	225	243	535
include/quat_messages.hh	18	78	17	113
src/cmake_file_list.cmake	2	0	12	14
src/quat.cc	26	65	62	153
src/quat_from_mat.cc	19	108	51	178
src/quat_messages.cc	17	30	7	54
src/quat_norm.cc	24	56	57	137
src/quat_to_eigenrot.cc	19	46	40	105
src/quat_to_mat.cc	18	77	27	122
Total	260	801	583	1644

Table 5.2 presents the cyclomatic complexity of the methods defined in the model.

Table 5.2: Cyclomatic Complexity

Method	File	Line	ECC
jeod::Quaternion::Quaternion (const double s, const double v[3])	include/quat_inline.hh	78	1
jeod::Quaternion::Quaternion (const double arr[4])	include/quat_inline.hh	91	1
jeod::Quaternion::set_to_zero ()	include/quat_inline.hh	100	1
jeod::Quaternion::make_identity ()	include/quat_inline.hh	109	1
jeod::Quaternion::copy_to (double arr[4])	include/quat_inline.hh	118	1
jeod::Quaternion::copy_from (const double arr[4])	include/quat_inline.hh	131	1
jeod::Quaternion::left_quat_from_eigen_rotation (double eigen_angle, const double eigen_axis[3])	include/quat_inline.hh	144	1
jeod::Quaternion::scale (const double fact)	include/quat_inline.hh	162	1
jeod::Quaternion::scale (const double fact, Quaternion & quat)	include/quat_inline.hh	172	1
jeod::Quaternion::norm_sq ()	include/quat_inline.hh	183	1
jeod::Quaternion::normalize (Quaternion & quat)	include/quat_inline.hh	192	1
jeod::Quaternion::normalize_integ (Quaternion & quat)	include/quat_inline.hh	202	1
jeod::Quaternion::conjugate ()	include/quat_inline.hh	212	1
jeod::Quaternion::conjugate (Quaternion & quat)	include/quat_inline.hh	220	1
jeod::Quaternion::multiply (const Quaternion & quat, Quaternion & prod)	include/quat_inline.hh	230	1
jeod::Quaternion::multiply (const Quaternion & quat)	include/quat_inline.hh	244	1

Continued on next page

Table 5.2: Cyclomatic Complexity (continued)

Method	File	Line	ECC
jeod::Quaternion::conjugate_multiply (const Quaternion & quat, Quaternion & prod)	include/quat_inline.hh	264	1
jeod::Quaternion::conjugate_multiply (const Quaternion & quat)	include/quat_inline.hh	278	1
jeod::Quaternion::multiply_conjugate (const Quaternion & quat, Quaternion & prod)	include/quat_inline.hh	298	1
jeod::Quaternion::multiply_conjugate (const Quaternion & quat)	include/quat_inline.hh	312	1
jeod::Quaternion::multiply_left (const Quaternion & quat, Quaternion & prod)	include/quat_inline.hh	332	1
jeod::Quaternion::multiply_left (const Quaternion & quat)	include/quat_inline.hh	346	1
jeod::Quaternion::multiply_left_conjugate (const Quaternion & quat, Quaternion & prod)	include/quat_inline.hh	366	1
jeod::Quaternion::multiply_left_conjugate (const Quaternion & quat)	include/quat_inline.hh	380	1
jeod::Quaternion::multiply_vector_left (const double vec[3], Quaternion & prod)	include/quat_inline.hh	400	1
jeod::Quaternion::multiply_vector_right (const double vec[3], Quaternion & prod)	include/quat_inline.hh	414	1
jeod::Quaternion::left_quat_transform (const double vec_in[3], double vec_out[3])	include/quat_inline.hh	428	1
jeod::Quaternion::eigen_compare (const Quaternion & quat, double * eigen_angle, double eigen_axis[3])	include/quat_inline.hh	448	1

Continued on next page

Table 5.2: Cyclomatic Complexity (continued)

Method	File	Line	ECC
jeod::Quaternion::compute_left_quat_deriv (const double ang_vel[3], Quaternion & qdot)	include/quat_inline.hh	461	1
jeod::Quaternion::compute_left_quat_second_deriv (const double ang_vel[3], const double ang_acc[3], Quaternion & qddot)	include/quat_inline.hh	473	1
jeod::Quaternion::compute_left_quat_deriv (const double quat[4], const double ang_vel[3], double qdot[4])	include/quat_inline.hh	489	1
jeod::Quaternion::compute_left_quat_second_deriv (const double quat[4], const double ang_vel[3], const double ang_acc[3], double qddot[4])	include/quat_inline.hh	504	1
jeod::Quaternion::Quaternion ()	src/quat.cc	50	2
jeod::Quaternion::Quaternion (const double real_part)	src/quat.cc	61	1
jeod::Quaternion::Quaternion (const double T[3][3])	src/quat.cc	70	1
jeod::Quaternion::compute_slerp (Quaternion & q1, Quaternion & q2, const double T)	src/quat.cc	79	6
jeod::Quaternion::left_quat_from_transformation (const double T[3][3])	src/quat_from_mat.cc	107	5
jeod::Quaternion::normalize ()	src/quat_norm.cc	43	4
jeod::Quaternion::normalize_integ ()	src/quat_norm.cc	79	3
jeod::Quaternion::normalize_integ (double quat[4])	src/quat_norm.cc	103	4

Continued on next page

Table 5.2: Cyclomatic Complexity (continued)

Method	File	Line	ECC
jeod::Quaternion::left_quat_to_eigen_rotation (double * eigen_angle, double eigen_axis[3])	src/quat_to_eigenrot.cc	39	4
jeod::Quaternion::left_quat_to_transformation (double T[3][3])	src/quat_to_mat.cc	74	1

5.5 Requirements Traceability

Bibliography

[1] Generated by doxygen. JEOD Quaternion Model Reference Manual, September 2024.

[2] Goldstein, H. *Classical Mechanics*. Addison-Wesley Publishing Company, Reading, Massachusett, second edition, 1980.

[3] Hammen, D. Dynamic Body Model. Technical Report JSC-61777-dynamics/dyn_body, NASA, Johnson Space Center, Houston, Texas, September 2024.

[4] Jackson, A., Thebeau, C. JSC Engineering Orbital Dynamics. Technical Report JSC-61777-docs, NASA, Johnson Space Center, Houston, Texas, September 2024.

[5] J. B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 2002.

[6] NASA. NASA Software Engineering Requirements. Technical Report NPR-7150.2, NASA, NASA Headquarters, Washington, D.C., September 2004.

[7] Spencer, A. Reference Frame Model. Technical Report JSC-61777-utils/ref_frames, NASA, Johnson Space Center, Houston, Texas, September 2024.

[8] Thebeau, C. Mass Body Model. Technical Report JSC-61777-dynamics/mass, NASA, Johnson Space Center, Houston, Texas, September 2024.

[9] Thompson, B. Orientation Model. Technical Report JSC-61777-utils/orientation, NASA, Johnson Space Center, Houston, Texas, September 2024.

Appendix A

Quaternion Mathematics

The JEOD uses unit quaternions[5] as one of several means for representing rotations and transformations in three space. This section provides an overview on the use of quaternions in the JEOD. The contents of this section are:

- Section A.1 defines the nomenclature used in this chapter.
- Section A.2 establishes the fundamentals of quaternions.
- Section A.3 develops elementary quaternion arithmetic.
- Section A.4 describes the extension of the exponential and logarithm to the quaternions.
- Section A.5 describes how quaternions are used to represent transformations and rotations.
- Section A.6 develops the time derivative of a transformation quaternion.
- Sections A.7 and A.8 describes techniques for propagating vehicle attitude and attitude rate using quaternions.

A.1 Nomenclature

Display style for vectors, matrices, and quaternions

Description	Nomenclature	Example
Scalar	in plain math font	ω
Vector	in bold math font	$\boldsymbol{\omega}$
Matrix	in bold math font	$\boldsymbol{T}$
Quaternion	in caligraphy font, uppercase	$\mathcal{Q}$

Vector Adornments

Description	Nomenclature	Example
Vector from a to b	Arrow-separated subscript	$\boldsymbol{x}_{a \to b}$
Vector from origin to b	Subscript on right	$\boldsymbol{x}_b$
Vector in frame A	Subscript on right	$\boldsymbol{x}_A$
Vector time derivative in frame A	Frame to right of dot	$\overset{\cdot A}{\boldsymbol{x}}$

Matrix Adornments

Description	Nomenclature	Example
Transformation from A to B	Arrow-separated subscript	$\boldsymbol{T}_{A \to B}$
Matrix product	No operator	$\boldsymbol{T}_{B \to C}\,\boldsymbol{T}_{A \to B}$
Matrix transpose	Superscript $\top$	$\boldsymbol{T}^\top$

Quaternion Adornments

Description	Nomenclature	Example
Transformation from A to B	Arrow-separated subscript	$\mathcal{Q}_{A \to B}$
Quaternion product	No operator	$\mathcal{Q}_{B \to C}\,\mathcal{Q}_{A \to B}$
Quaternion conjugate	Superscript $\star$	$\mathcal{Q}^\star$
Quaternion components	Scalar + vector *or* four-vector	$\begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ *or* $\begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix}$

A.2 Quaternion Fundamentals

The quaternions are an extension of the complex numbers first invented by William Rowan Hamilton with three distinct square roots of -1, which are typically denoted as i, j, and k.

A.2.1 Fundamental Formula

The quaternion imaginary units obey Hamilton's Fundamental Formula of Quaternion Algebra,

$$i^2 = j^2 = k^2 = ijk = -1 \tag{A.1}$$

An immediate consequence of this definition is that multiplication of quaternion imaginary units is not commutative:

$$ij = k = -ji \tag{A.2a}$$

$$jk = i = -kj \tag{A.2b}$$

$$ki = j = -ik \tag{A.2c}$$

A.2.2 Quaternion Representation

Just as a complex number can be represented as a sum of real and imaginary parts, a quaternion can also be written as a linear combination of real and imaginary quaternion parts:

$$\mathcal{Q} = q_s + q_i i + q_j j + q_k k \tag{A.3}$$

or more compactly as the four-vector

$$\mathcal{Q} = \begin{bmatrix} q_s \\ q_i \\ q_j \\ q_k \end{bmatrix} \tag{A.4}$$

or even more compactly by representing the imaginary quaternion part as a vector:

$$\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \tag{A.5}$$

where $\boldsymbol{q_v}$ comprises the imaginary quaternion part of $\mathcal{Q}$:

$$\boldsymbol{q_v} = \begin{bmatrix} q_i \\ q_j \\ q_k \end{bmatrix} \tag{A.6}$$

JEOD represents quaternions in the scalar + vector form. This representation scheme is used in this document.

A.2.3 Special Cases

Quaternions of particular interest are:

- The quaternion whose scalar and vector parts are identically zero is the *zero quaternion.*
- A quaternion whose vector part is identically zero is a *pure real quaternion.*
- A quaternion whose scalar part is identically zero is a *pure imaginary quaternion.*
- A quaternion of the form $\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ with ${q_s}^2 + \boldsymbol{q_v} \cdot \boldsymbol{q_v} = 1$ is a *unit quaternion.*
- The quaternion $\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ is the *real unit quaternion.*

A.3 Quaternion Arithmetic

A.3.1 Basic Quaternion Operations

This section develops the basic arithmetic operations on quaternions. In this section, $\mathcal{Q}$, $\mathcal{Q}_1$, and $\mathcal{Q}_2$ are quaternions defined with components

$$\mathcal{Q} = q_s + q_i i + q_j j + q_k k = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$$

$$\mathcal{Q}_1 = q_{1_s} + q_{1_i} i + q_{1_j} j + q_{1_k} k = \begin{bmatrix} q_{1_s} \\ \boldsymbol{q_{1_v}} \end{bmatrix}$$

$$\mathcal{Q}_2 = q_{2_s} + q_{2_i} i + q_{2_j} j + q_{2_k} k = \begin{bmatrix} q_{2_s} \\ \boldsymbol{q_{2_v}} \end{bmatrix}$$

Quaternion addition, scaling a quaternion by a real, and the quaternion norm are defined as isomorphisms to the corresponding operations on a $4-$vector:

Definition A.1 (Quaternion addition)**:**
Given quaternions $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the quaternion sum $\mathcal{Q}_1 + \mathcal{Q}_2$ is defined as

$$\mathcal{Q}_1 + \mathcal{Q}_2 \equiv q_{1_s} + q_{2_s} + (q_{2_i} + q_{2_i}))\, i + (q_{2_j} + q_{2_j}))\, j + (q_{2_k} + q_{2_k}))\, k = \begin{bmatrix} q1_s + q2_s \\ \boldsymbol{q_{1_v}} + \boldsymbol{q_{2_v}} \end{bmatrix} \tag{A.7}$$

Definition A.2 (Quaternion scaling)**:**
Given a quaternion $\mathcal{Q}$ and a real number s, the product of $\mathcal{Q}$ with s is defined as

$$s\mathcal{Q} \equiv sq_s + sq_i i + sq_j j + sq_k k = \begin{bmatrix} sq_s \\ s\boldsymbol{q_v} \end{bmatrix} \tag{A.8}$$

Definition A.3 (Quaternion norm)**:**
Given a quaternion $\mathcal{Q}$, the quaternion norm of $\mathcal{Q}$ is defined as

$$\|\mathcal{Q}\| \equiv \sqrt{q_s^2 + q_i^2 + q_j^2 + q_k^2} = \sqrt{q_s^2 + \boldsymbol{q_v} \cdot \boldsymbol{q_v}} \tag{A.9}$$

The quaternion conjugate is defined analogously to the complex conjugate:

Definition A.4 (Quaternion conjugate)**:**
Given a quaternion $\mathcal{Q}$, the quaternion conjugate of $\mathcal{Q}$, denoted herein as either $\operatorname{conj}\mathcal{Q}$ or $\mathcal{Q}^\star$, is defined as

$$\operatorname{conj}\mathcal{Q} \equiv \mathcal{Q}^\star \equiv q_s - q_i i - q_j j - q_k k = \begin{bmatrix} q_s \\ -\boldsymbol{q_v} \end{bmatrix} \tag{A.10}$$

The definition of the quaternion product follows from Hamilton's Fundamental Formula of Quaternion Algebra, equation (A.1):

Definition A.5 (Quaternion multiplication)**:**
Given quaternions $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the quaternion product $\mathcal{Q}_1\,\mathcal{Q}_2$ is defined as

$$\begin{aligned} \mathcal{Q}_1\,\mathcal{Q}_2 \equiv\; & q_{1_s}q_{2_s} - (q_{1_i}q_{2_i} + q_{1_j}q_{2_j} + q_{1_k}q_{2_k}) \\ & + (q_{1_s}q_{2_i} + q_{2_s}q_{1_i} + (q_{1_j}q_{2_k} - q_{1_k}q_{2_j}))\, i \\ & + (q_{1_s}q_{2_j} + q_{2_s}q_{1_j} + (q_{1_k}q_{2_i} - q_{1_i}q_{2_k}))\, j \\ & + (q_{1_s}q_{2_k} + q_{2_s}q_{1_k} + (q_{1_i}q_{2_j} - q_{1_j}q_{2_i}))\, k \end{aligned} \tag{A.11}$$

or in scalar + vector form,

$$\mathcal{Q}_1\,\mathcal{Q}_2 = \begin{bmatrix} q_{1_s}q_{2_s} - \boldsymbol{q_{1_v}} \cdot \boldsymbol{q_{2_v}} \\ q_{1_s}\boldsymbol{q_{2_v}} + q_{2_s}\boldsymbol{q_{1_v}} + \boldsymbol{q_{1_v}} \times \boldsymbol{q_{2_v}} \end{bmatrix} \tag{A.12}$$

Note that definitions A.2 and A.5 are consistent: The product of a quaternion of the form $\begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ and a real quaternion $\begin{bmatrix} s \\ \boldsymbol{0} \end{bmatrix}$ is

$$\begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \begin{bmatrix} s \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} s \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} = \begin{bmatrix} sq_s \\ s\boldsymbol{q_v} \end{bmatrix} = s\mathcal{Q}$$

Corollary A.1. *The product of a quaternion and its conjugate is the square of the norm of the quaternion.*

Proof. The product of a quaternion $\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ and its conjugate is

$$\begin{aligned} \mathcal{Q}\,\mathcal{Q}^\star &= \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \begin{bmatrix} q_s \\ -\boldsymbol{q_v} \end{bmatrix} \\ &= \begin{bmatrix} q_s^2 + \boldsymbol{q_v} \cdot \boldsymbol{q_v} \\ \boldsymbol{0} \end{bmatrix} \end{aligned}$$

□

A.3.2 Commutativity, Associativity, and Distributivity

Theorem A.2. *Quaternion addition and scaling a quaternion by a real scalar are commutative operations.*

Proof. The quaternion sum and the product of a quaternion and a real are defined as isomorphisms to the corresponding commutative $4-$vector operations. □

Note that the product of two quaternions does not in general commute. This follows directly from the Fundamental Formula of Quaternion Algebra: $ij = k, ji = -k$.

Theorem A.3. *Quaternion multiplication is associative:*

$$(\mathcal{Q}_1\, \mathcal{Q}_2)\, \mathcal{Q}_3 = \mathcal{Q}_1\, (\mathcal{Q}_2\, \mathcal{Q}_3)$$

Proof. This can be easily proven to be the case for three quaternions. Extending the theorem to a product of any number of quaternions and to any grouping of parentheses in the product is accomplished by induction. □

Theorem A.4. *Quaternion multiplication distributes over quaternion addition:*

$$\mathcal{Q}_1\, (\mathcal{Q}_2 + \mathcal{Q}_3) = \mathcal{Q}_1\, \mathcal{Q}_2 + \mathcal{Q}_1 \mathcal{Q}_3 \tag{A.13}$$

This can be easily proven true for any three arbitrary quaternions $\mathcal{Q}_1$, $\mathcal{Q}_2$, and $\mathcal{Q}_3$ by applying the definitions of quaternion addition and multiplication.

A.3.3 Quaternion Identity, Inverse, and Division

Theorem A.5. *The real unit quaternion is the identity element for quaternion multiplication.*

Proof. The left and right products of any quaternion $\begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ and the real unit quaternion $\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ reproduce the quaternion $\mathcal{Q}$:

$$\begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$$

The real unit quaternion is thus both a left- and right- identity element. Since quaternion multiplication is associative, the multiplicative identity is unique. The real unit quaternion is thus the unique identity element for quaternion multiplication. □

Theorem A.6. *The inverse $\mathcal{Q}^{-1}$ of a non-zero quaternion $\mathcal{Q}$ is*

$$\mathcal{Q}^{-1} = \frac{1}{\mathcal{Q}} \equiv \frac{1}{\|\mathcal{Q}\|^2}\mathcal{Q}^\star = \frac{1}{\mathcal{Q}\,\mathcal{Q}^\star}\mathcal{Q}^\star \tag{A.14}$$

Proof.

Let

$$
\begin{aligned}
\mathcal{Q} &= \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix} \\
\mathcal{Q}_2 &= \frac{1}{\mathcal{Q}\,\mathcal{Q}^\star}\mathcal{Q}^\star
\end{aligned}
$$

then

$$
\begin{aligned}
\mathcal{Q}\,\mathcal{Q}_2 &= \frac{1}{\mathcal{Q}\,\mathcal{Q}^\star}\begin{bmatrix} q_s^2 + \boldsymbol{q_v}\cdot\boldsymbol{q_v} \\ \boldsymbol{0} \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ \boldsymbol{0} \end{bmatrix}
\end{aligned}
$$

Thus the stated quaternion is a right multiplicative inverse. Since quaternion multiplication is associative, the stated quaternion is also a left multiplicative inverse and is unique. □

Corollary A.7. *The multiplicative inverse of a unit quaternion is the quaternion conjugate.*

Proof. This follows directly from the definitions of the quaternion multiplicative inverse and the quaternion conjugate. □

Quaternion division is defined in terms of multiplication with the quaternion inverse. Since quaternion multiplication is not commutative, pre-multiplying and post-multiplying a quaternion with the inverse of another quaternion yield different results. This observation leads to the definition of a left and right division operator:

Definition A.6 (Quaternion division)**:**

$$
\mathcal{Q}_1 \setminus \mathcal{Q}_2 \equiv {\mathcal{Q}_2}^{-1}\,\mathcal{Q}_1 \tag{A.15}
$$

$$
\mathcal{Q}_1 / \mathcal{Q}_2 \equiv \mathcal{Q}_1\,{\mathcal{Q}_2}^{-1} \tag{A.16}
$$

A.3.4 Quaternion Decomposition

This section develops an alternative representation scheme for quaternions.

Definition A.7:
A non-zero quaternion represented in terms of its magnitude $s \in \mathbb{R}$, a unit vector $\boldsymbol{\hat{u}} \in \mathbb{R}^3$, and a rotation angle $\theta \in \mathbb{R}$,

$$
\mathcal{Q} = s\begin{bmatrix} \cos\theta \\ \sin\theta\,\boldsymbol{\hat{u}} \end{bmatrix} \tag{A.17}
$$

is a *decomposition* of the quaternion.

Corollary A.8. *The decomposition of a quaternion not unique.*

Proof. if $s\begin{bmatrix} \cos\theta \\ \sin\theta\,\boldsymbol{\hat{u}} \end{bmatrix} = \mathcal{Q}$ then so does $s\begin{bmatrix} \cos(\theta + 2n\pi) \\ \sin(\theta + 2n\pi)\,\boldsymbol{\hat{u}} \end{bmatrix}$ for all integer values of n. □

Corollary A.9. *The decomposition of the real unit quaternion* $\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ *is a trivial rotation* ($\theta = 0$) *about an indeterminate axis.*

Proof. Applying equation (A.17) with $s = 1$ and $\theta = 0$ generates the real unit quaternion regardless of the value of $\hat{\boldsymbol{u}}$. □

Corollary A.10. *Given a quaternion* $\mathcal{Q}$ *with decomposition* $\mathcal{Q} = s\begin{bmatrix} \cos\theta \\ \sin\theta\hat{\boldsymbol{u}} \end{bmatrix}$, *a decomposition of the additive inverse of* $\mathcal{Q}$ *is* $-\mathcal{Q} = s\begin{bmatrix} \cos(\theta+\pi) \\ \sin(\theta+\pi)\hat{\boldsymbol{u}} \end{bmatrix}$,

Proof. Expanding $\begin{bmatrix} \cos(\theta+\pi) \\ \sin(\theta+\pi)\hat{\boldsymbol{u}} \end{bmatrix}$ results in $-\mathcal{Q}$. □

Algorithm A.1 Unit quaternion decomposition

Let $\mathcal{Q} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$ be a unit quaternion (${q_s}^2 + \boldsymbol{q_v} \cdot \boldsymbol{q_v} = 1$) with non-zero vector part. Then

$$\theta = \arctan(\|\boldsymbol{q_v}\|, q_s) \tag{A.18}$$

$$\hat{\boldsymbol{u}} = \frac{\boldsymbol{q_v}}{\|\boldsymbol{q_v}\|} \tag{A.19}$$

is a *decomposition* of the unit quaternion.

Note The two-argument inverse tangent function in equation (A.18) computes the inverse tangent of y/x, with the signs of y and x dictating the quadrant of the result in which the result is placed. The C-library function `atan2` approximates $\arctan(y, x)$. The single-argument inverse cosine and inverse sine functions can be used as an alternative to `atan2`. However, care must be taken to use the function that produces better accuracy and to place the angle in the correct quadrant:

$$\theta = \begin{cases} \pi - \arcsin(\|\boldsymbol{q_v}\|) & \text{if } \quad q_s < -\frac{\sqrt{2}}{2} \\ \arccos(q_s) & \text{if } -\frac{\sqrt{2}}{2} \le q_s \le \frac{\sqrt{2}}{2} \\ \arcsin(\|\boldsymbol{q_v}\|) & \text{if } \quad q_s > \frac{\sqrt{2}}{2} \end{cases} \tag{A.20}$$

Which alternative (equation (A.18) or (A.20)) is faster and/or more accurate is machine-dependent.

Proof.

$$\begin{aligned} \boldsymbol{q_v} &= \|\boldsymbol{q_v}\| \frac{\boldsymbol{q_v}}{\|\boldsymbol{q_v}\|} \\ &= \sqrt{1 - {q_s}^2}\frac{\boldsymbol{q_v}}{\|\boldsymbol{q_v}\|} \quad \text{since } \|\boldsymbol{q_v}\|^2 = 1 - {q_s}^2 \end{aligned}$$

thus

$$\begin{aligned} \cos\theta &= q_s \\ \sin\theta &= \sqrt{1 - {q_s}^2} = \|\boldsymbol{q}_v\| \end{aligned}$$

Applying equation (A.17) with $s = 1$,

$$\begin{aligned} \begin{bmatrix} \cos\theta \\ \sin\theta\hat{\boldsymbol{u}} \end{bmatrix} &= \begin{bmatrix} q_s \\ \|\boldsymbol{q}_v\| \frac{\boldsymbol{q_v}}{\|\boldsymbol{q_v}\|} \end{bmatrix} \\ &= \mathcal{Q} \end{aligned}$$

□

Corollary A.11. *A decomposition exists for all unit quaternions.*

Proof. Corollaries A.9 and A.10 provide decompositions for the two unit quaternions with zero vector parts. Since $\arctan(y, x)$ is defined for all finite values of x and y, algorithm A.1 provides a decomposition for all unit quaternions with non-zero vector parts. □

Corollary A.12. *A decomposition exists for all quaternions.*

Proof. The zero quaternion has an indeterminate decomposition: $s = 0$ and θ and $\hat{\boldsymbol{u}}$ are indeterminate. Represent a non-zero quaternion $\mathcal{Q}$ as $\|\mathcal{Q}\|\frac{\mathcal{Q}}{\|\mathcal{Q}\|}$. The unit quaternion $\frac{\mathcal{Q}}{\|\mathcal{Q}\|}$ has some decomposition $(s = 1, \theta, \hat{\boldsymbol{u}})$ by corollary A.11. The decomposition $(s = \|\mathcal{Q}\|, \theta, \hat{\boldsymbol{u}})$ generates the original quaternion $\mathcal{Q}$. □

Corollary A.13. *The multiplicative inverse of a quaternion $\mathcal{Q}$ is*

$$\mathcal{Q}^{-1} = \frac{1}{s}\begin{bmatrix} \cos\theta \\ -\sin\theta\hat{\boldsymbol{u}} \end{bmatrix} \tag{A.21}$$

where s, θ, and $\hat{\boldsymbol{u}}$ are a decomposition of the quaternion $\mathcal{Q}$.

Proof. Multiplying the decomposition of $\mathcal{Q}$ and the inverse as expressed in equation (A.21) yields the real unit quaternion. □

A.3.5 Quaternion Exponentiation

Raising a non-zero quaternion $\mathcal{Q}$ to a real integer power is defined in terms of iterated multiplication, analogous to the definition for real and complex numbers:

Definition A.8 (Integer Power of a Quaternion)**:**

$$\mathcal{Q}^n \equiv \begin{cases} \prod_{r=1}^{n} \mathcal{Q} & \text{if } n > 0 \\ \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} & \text{if } n = 0 \\ \left(\mathcal{Q}^{-1}\right)^n = \prod_{r=1}^{n} \mathcal{Q}^{-1} & \text{if } n < 0 \end{cases} \tag{A.22}$$

Since quaternion multiplication is associative (theorem A.3), the products in equation (A.22) are well-defined.

Theorem A.14. *The result of raising a quaternion $\mathcal{Q}$ to a rational power r is*

$$\mathcal{Q}^r = s^r \begin{bmatrix} \cos r\theta \\ \sin r\theta \hat{\boldsymbol{u}} \end{bmatrix} \tag{A.23}$$

where s, θ, and $\hat{\boldsymbol{u}}$ are a decomposition of the quaternion $\mathcal{Q}$.

Proof. The theorem is clearly true for $r = 0$, since equations (A.22) and (A.23) yields the real unit quaternion for $r = 0$. Assuming the theorem is true for some integer $r \neq 0$, then

$$\begin{aligned} \mathcal{Q}^{r+1} &= \mathcal{Q}^r \, \mathcal{Q} \\ &= s^r \begin{bmatrix} \cos r\theta \\ \sin r\theta \hat{\boldsymbol{u}} \end{bmatrix} s \begin{bmatrix} \cos \theta \\ \sin \theta \hat{\boldsymbol{u}} \end{bmatrix} \\ &= s^{r+1} \begin{bmatrix} \cos\left((r+1)\,\theta\right) \\ \sin\left((r+1)\,\theta\right) \hat{\boldsymbol{u}} \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathcal{Q}^{r-1} &= \mathcal{Q}^r \, \mathcal{Q}^{-1} \\ &= s^r \begin{bmatrix} \cos r\theta \\ \sin r\theta \hat{\boldsymbol{u}} \end{bmatrix} s^{-1} \begin{bmatrix} \cos \theta \\ -\sin \theta \hat{\boldsymbol{u}} \end{bmatrix} \\ &= s^{r-1} \begin{bmatrix} \cos\left((r-1)\,\theta\right) \\ \sin\left((r-1)\,\theta\right) \hat{\boldsymbol{u}} \end{bmatrix} \end{aligned}$$

By induction, the theorem is true for all integers.

Now consider the n^{th} root of $\mathcal{Q}$ as defined by equation (A.23). Raising this to the integral n^{th} power yields

$$\begin{aligned} \left(s^{1/n} \begin{bmatrix} \cos \frac{\theta}{n} \\ \sin \frac{\theta}{n} \hat{\boldsymbol{u}} \end{bmatrix} \right)^n &= \left(s^{1/n} \right)^n \begin{bmatrix} \cos n\frac{\theta}{n} \\ \sin n\frac{\theta}{n} \hat{\boldsymbol{u}} \end{bmatrix} \\ &= \mathcal{Q} \end{aligned}$$

The theorem has now been proven for all integers and all rationals of the form $1/n$. It is therefore valid for any number of the form p/q, where p and q are integers. □

By continuity, equation (A.23) is extended to cover all the real numbers.

Definition A.9 (Real power of a quaternion)**:**
The result of raising a quaternion $\mathcal{Q}$ to a real power a is

$$\mathcal{Q}^a = s^a \begin{bmatrix} \cos a\theta \\ \sin a\theta \hat{\boldsymbol{u}} \end{bmatrix} \tag{A.24}$$

where s, θ, and $\hat{\boldsymbol{u}}$ are a decomposition of the quaternion $\mathcal{Q}$.

A.4 Quaternion Exponential

This section develops the exponential of a quaternion. The exponential function is the most important function in mathematics. It is defined, for every complex number z, by $\exp z = \sum_{n=0}^{\infty} \frac{z^n}{n!}$. Extending this definition to the quaternions,

Definition A.10 (Quaternion Exponential)**:**
The quaternion exponential function is defined as

$$\exp \mathcal{Q} \equiv \sum_{n=0}^{\infty} \frac{\mathcal{Q}^n}{n!} \tag{A.25}$$

The complex exponential maps the pure imaginary numbers to the unit circle via Euler's Equation, $e^{i\theta} = \cos\theta + i\sin\theta$. The following theorem demonstrates that the quaternion exponential has a similar mapping function when applied to pure imaginary quaternions.

Theorem A.15. *Let $\mathcal{Q}$ be a pure imaginary quaternion of the form* $\begin{bmatrix} 0 \\ \theta\hat{\boldsymbol{u}} \end{bmatrix}$.
Then

$$\exp \mathcal{Q} = \begin{bmatrix} \cos\theta \\ \sin\theta\hat{\boldsymbol{u}} \end{bmatrix} \tag{A.26}$$

Proof.

The quaternion as defined above has decomposition

$$\mathcal{Q} \equiv \begin{bmatrix} 0 \\ \theta\hat{\boldsymbol{u}} \end{bmatrix} = \theta \begin{bmatrix} \cos\frac{\pi}{2} \\ \sin\pi 2\hat{\boldsymbol{u}} \end{bmatrix}$$

By equation (A.23),

$$\begin{aligned}
\mathcal{Q}^{2n} &= \theta^{2n} \begin{bmatrix} \cos((2n)\frac{\pi}{2}) \\ \sin((2n)\frac{\pi}{2})\hat{\boldsymbol{u}} \end{bmatrix} \\
&= (-1)^n \theta^{2n} \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \\
\mathcal{Q}^{2n+1} &= \theta^{2n+1} \begin{bmatrix} \cos((2n+1)\frac{\pi}{2}) \\ \sin((2n+1)\frac{\pi}{2})\hat{\boldsymbol{u}} \end{bmatrix} \\
&= (-1)^n \theta^{2n+1} \begin{bmatrix} 0 \\ \hat{\boldsymbol{u}} \end{bmatrix}
\end{aligned}$$

Partitioning the sum equation (A.10) into even and odd powers of n yields

$$\exp \mathcal{Q} = \sum_{n=0}^{\infty} \frac{\mathcal{Q}^{2n}}{(2n)!} + \sum_{n=0}^{\infty} \frac{\mathcal{Q}^{2n+1}}{(2n+1)!}$$

and thus

$$\exp \begin{bmatrix} 0 \\ \theta \hat{\boldsymbol{u}} \end{bmatrix} = \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \theta^{2n} \right) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} + \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \theta^{2n+1} \right) \begin{bmatrix} 0 \\ \hat{\boldsymbol{u}} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta \\ \sin\theta \hat{\boldsymbol{u}} \end{bmatrix}$$

□

Corollary A.16. *Let $\mathcal{Q}$ be a quaternion of the form* $\begin{bmatrix} q_s \\ q_v \hat{\boldsymbol{u}} \end{bmatrix}$.
Then

$$\exp \mathcal{Q} = \exp q_s \begin{bmatrix} \cos q_v \\ \sin q_v \hat{\boldsymbol{u}} \end{bmatrix} \tag{A.27}$$

where

$\exp q_s$ *is the real exponential function of the real scalar* q_s

Proof. Since the product of a quaternion and a real quaternion commutes,

$$\mathcal{Q}^n = \sum_{r=0}^{n} \frac{n!}{r!(n-r)!} \begin{bmatrix} q_s \\ \mathbf{0} \end{bmatrix}^r \begin{bmatrix} 0 \\ q_v \hat{\boldsymbol{u}} \end{bmatrix}^{n-r}$$

and thus

$$\exp \mathcal{Q} = \sum_{n=0}^{\infty} \sum_{r=0}^{n} \frac{1}{r!(n-r)!} \begin{bmatrix} q_s \\ \mathbf{0} \end{bmatrix}^r \begin{bmatrix} 0 \\ q_v \hat{\boldsymbol{u}} \end{bmatrix}^{n-r}$$

Collecting like powers of $\begin{bmatrix} q_s \\ \mathbf{0} \end{bmatrix}$ and setting $n - r \equiv s$,

$$\exp \mathcal{Q} = \left(\sum_{r=0}^{\infty} \frac{1}{r!} \begin{bmatrix} q_s \\ \mathbf{0} \end{bmatrix}^r \right) \left(\sum_{s=0}^{\infty} \frac{1}{s!} \begin{bmatrix} 0 \\ q_v \hat{\boldsymbol{u}} \end{bmatrix}^s \right)$$

$$= \exp q_s \begin{bmatrix} \cos q_v \\ \sin q_v \hat{\boldsymbol{u}} \end{bmatrix}$$

□

The quaternion logarithm function is defined as the inverse of the quaternion exponential:

Definition A.11 (Quaternion logarithm)**:**

$$\mathcal{Q}_2 = \log \mathcal{Q} \iff \exp \mathcal{Q}_2 = \mathcal{Q}$$

Note that the quaternion logarithm of the zero quaternion is undefined since there is no quaternion $\mathcal{Q}$ such that $\exp \mathcal{Q} = \mathbf{0}$.

Theorem A.17. *The quaternion logarithm of a quaternion $\mathcal{Q}$ is*

$$\log \mathcal{Q} = \begin{bmatrix} \log s \\ \theta \hat{\boldsymbol{u}} \end{bmatrix} \tag{A.28}$$

where s, θ, and $\hat{\boldsymbol{u}}$ are a decomposition of the quaternion $\mathcal{Q}$.

Proof. Exponentiating the quaternion $\begin{bmatrix} \log s \\ \theta \hat{\boldsymbol{u}} \end{bmatrix}$ yields, per equation (A.27), $s \begin{bmatrix} \cos\theta \\ \sin\theta \hat{\boldsymbol{u}} \end{bmatrix} = \mathcal{Q}$. □

Corollary A.18. *The quaternion logarithm of a unit quaternion $\mathcal{Q}$ is*

$$\log \mathcal{Q} = \begin{bmatrix} 0 \\ \theta \hat{\boldsymbol{u}} \end{bmatrix} \tag{A.29}$$

where θ, and $\hat{\boldsymbol{u}}$ are a decomposition of the unit quaternion $\mathcal{Q}$.

Proof. This follows directly from theorem A.17. The scalar s in the decomposition of a unit quaternion is 1, and $\log 1 = 0$. □

A.5 Rotation and Transformation

One reason quaternions are so useful is their ability to represent rotations and transformations. This section develops these capabilities. While quaternions do have other applications beyond their ability to compactly represent rotations and transformations, the JEOD uses quaternions solely for this purpose.

Two possible conventions exist when describing rotations: rotation of an object relative to fixed axes and rotation of the axes relative to a fixed object. In this discussion, the former will be referred to as a *rotation* and the latter, a *transformation*. The two concepts are closely related. For example, the *rotation* matrix that rotates a vector about the $\hat{\boldsymbol{u}}$ axis by an angle of θ is the transpose of the *transformation* matrix that transforms a vector to the frame whose axes are rotated about the $\hat{\boldsymbol{u}}$ axis by an angle of θ relative to the original frame.

A.5.1 Single-Axis Rotations

Theorem A.19 (Single-Axis Rotation)**.** *Rotating a $3-$vector $\boldsymbol{x}$ by an angle θ about an axis directed along the $\hat{\boldsymbol{u}}$ unit vector results in*

$$\boldsymbol{x}' = \cos\theta\, \boldsymbol{x} + (1 - \cos\theta)(\hat{\boldsymbol{u}} \cdot \boldsymbol{x})\hat{\boldsymbol{u}} + \sin\theta\, \hat{\boldsymbol{u}} \times \boldsymbol{x} \tag{A.30}$$

where $\boldsymbol{x}'$ is the result of the rotation.

Proof. The rotation does not affect the component of $\boldsymbol{x}$ along the rotation axis. Rotating a vector $\boldsymbol{x}$ directed solely along the rotation axis $\boldsymbol{x} = x\hat{\boldsymbol{u}}$ has no effect: $\boldsymbol{x}' = \boldsymbol{x}$, which agrees with the theorem since $\hat{\boldsymbol{u}} \times \boldsymbol{x} = 0$ in this case.

Assuming $\boldsymbol{x}$ has some non-zero component normal to the rotation axis, represent $\boldsymbol{x}$ as

$$\boldsymbol{x} = \boldsymbol{x}_u + \boldsymbol{x}_v$$

where

$$\begin{aligned}
\boldsymbol{x}_u &= (\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}} \\
\boldsymbol{x}_v &= \boldsymbol{x} - (\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}}
\end{aligned}$$

Note that $\boldsymbol{x}_v$ is normal to $\hat{\boldsymbol{u}}$ by construction and is non-zero by assumption. Let

$$\begin{aligned}
\hat{\boldsymbol{v}} &= \frac{\boldsymbol{x}_v}{\|\boldsymbol{x}_v\|} \\
\hat{\boldsymbol{w}} &= \hat{\boldsymbol{u}} \times \hat{\boldsymbol{v}}, \text{ completing a RHS.}
\end{aligned}$$

Rotating $\boldsymbol{x}_v$ by an angle θ in the vw plane results in

$$\begin{aligned}
\boldsymbol{x}'_v &= \|\boldsymbol{x}_v\|(\cos\theta\,\hat{\boldsymbol{v}} + \sin\theta\,\hat{\boldsymbol{w}}) \\
&= \cos\theta\,(\boldsymbol{x} - (\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) + \sin\theta\,\hat{\boldsymbol{u}} \times \boldsymbol{x}_v
\end{aligned}$$

The rotation does not affect $\boldsymbol{x}_u$ and thus

$$\begin{aligned}
\boldsymbol{x}' &= \boldsymbol{x}_u + \boldsymbol{x}'_v \\
&= (\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}} + \cos\theta\,(\boldsymbol{x} - (\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) + \sin\theta\,\hat{\boldsymbol{u}} \times \boldsymbol{x}_v \\
&= \cos\theta\,\boldsymbol{x} + (1 - \cos\theta)(\boldsymbol{x} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}} + \sin\theta\,\hat{\boldsymbol{u}} \times \boldsymbol{x}
\end{aligned}$$

□

Theorem A.20 (Single-Axis Transformation)**.** *Given a frame B whose axes are rotated about the $\hat{\boldsymbol{u}}$ axis by an angle of θ relative to frame A, a three-vector $\boldsymbol{x}_A$ expressed in frame A transforms to frame B via*

$$\boldsymbol{x}_B = \cos\theta\,\boldsymbol{x}_A + (1 - \cos\theta)(\hat{\boldsymbol{u}} \cdot \boldsymbol{x}_A)\hat{\boldsymbol{u}} - \sin\theta\,\hat{\boldsymbol{u}} \times \boldsymbol{x}_A \tag{A.31}$$

Proof. This theorem follows immediately from theorem A.19 by recognizing that each axis of frame B is the corresponding axis of frame A rotated according to theorem A.19 and applying the scalar triple product rule. □

A.5.2 Rotation and Transformation Quaternions

Any quaternion can be used to represent a rotation or transformation of a 3−vector via one of the two forms

$$\mathcal{Q} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}^{-1} \qquad \text{or} \qquad \mathcal{Q}^{-1} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}$$

A *unit* quaternion can be used to represent a rotation or transformation of a 3−vector via one of the two forms

$$\mathcal{Q} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}^\star \qquad \text{or} \qquad \mathcal{Q}^\star \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}$$

That such forms do indeed rotate or transform a 3−vector will be developed later in this section.

The four forms differ in the use of the quaternion inverse versus the conjugate and the placement of the quaternion and its inverse or conjugate relative to the vector to be rotated or transformed:

- The inverse is used in the first pair of forms, while the conjugate in the second pair. Since the inverse of a unit quaternion is the conjugate, the second pair of forms is a specialization of the first pair for the unit quaternions.

- The quaternion is placed to the *left* of the vector in the first triple product of each pair of forms. Quaternion used for rotation or transformation based on these forms are thus called *left* rotation or transformation quaternions.

- The quaternion is placed to the *right* of the vector in the second triple product of each pair of forms. Quaternion used for rotation or transformation based on these forms are thus called *right* rotation or transformation quaternions.

There is no loss and much to be gained by eliminating the first pair of forms from consideration—*i.e.*, to restrict rotation/transformation quaternions to the unit quaternions. Since multiplying a quaternion by a real scalar commutes, pre- and post-multiplying either of the first pair of rotation/transformation forms by a real scalar and its multiplicative inverse will not affect the outcome of form. Given a general quaternion that achieves a desired rotation or transformation, the unit quaternion formed by scaling the original quaternion by the inverse of its magnitude thus achieves the same rotation or transformation as does the original quaternion. There is no loss of expressiveness in retricting rotation/transformation quaternions to the unit quaternions. At the same time, there is a considerable gain in making this restriction. The most obvious gain is in the rotation/transformation forms themselves. Finding a quaternion's inverse involves finding its conjugate and then performing extra calculations. The restriction bypasses those extra calculations. More importantly, operations that involve the rotation/transformation quaternion but not its inverse are sensitive to scaling and typically take on a simpler form. For example, the logarithm and the time derivative of a unit quaternion are pure imaginary quaternions. For these reasons, unit quaternions are used almost exclusively to represent rotations and transformations.

There is no outstanding reason, however, to prefer left versus right rotation/transformation quaternions. Making an *a-priori* choice is beneficial in the sense that doing so simplifies the software and reduces the chances of an error in interpretation. However, when one receives a quaternion from an external organization, care must be taken in interpreting that quaternion. One must also take care in the interpretation of the elements of some externally-generated quaternion, as the choice of placing the scalar part first or last in a 4−vector representation is arbitrary.

Left Rotation and Transformation Quaternions

Algorithm A.2
The left rotation unit quaternion

$$\mathcal{Q}_{rot} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix} \tag{A.32}$$

rotates a three-vector $\boldsymbol{x}$ about the $\hat{\boldsymbol{u}}$ axis by an angle of θ:

$$\begin{bmatrix} 0 \\ \boldsymbol{x}' \end{bmatrix} = \mathcal{Q}_{rot} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}^{\star}{}_{rot} \tag{A.33}$$

where $\boldsymbol{x}'$ is the result of the rotation.

Proof. To demonstrate that this quaternion does indeed perform the specified rotation, expanding equation (A.33) results in

$$\begin{aligned} \mathcal{Q}_{rot} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}^{\star}{}_{rot} &= \begin{bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \sin^2\frac{\theta}{2}\hat{\boldsymbol{u}}\cdot\boldsymbol{x}\hat{\boldsymbol{u}} + \cos^2\frac{\theta}{2}\boldsymbol{x} + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\hat{\boldsymbol{u}}\times\boldsymbol{x} - \sin^2\frac{\theta}{2}\left(\hat{\boldsymbol{u}}\times\boldsymbol{x}\right)\times\hat{\boldsymbol{u}} \end{bmatrix} \end{aligned}$$

Using the half-angle formulae

$$\begin{aligned} \sin^2\frac{\theta}{2} &= \frac{1}{2}(1-\cos\theta) \\ \cos^2\frac{\theta}{2} &= \frac{1}{2}(1+\cos\theta) \end{aligned}$$

and the vector triple product identity

$$\left(\hat{\boldsymbol{u}}\times\boldsymbol{x}\right)\times\hat{\boldsymbol{u}} = \boldsymbol{x} - \left(\boldsymbol{x}\cdot\hat{\boldsymbol{u}}\right)\hat{\boldsymbol{u}}$$

the quaternion product equation (A.33) simplifies to

$$\mathcal{Q}_{rot} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}^{\star}{}_{rot} = \begin{bmatrix} 0 \\ \cos\theta\boldsymbol{x} + (1-\cos\theta)(\hat{\boldsymbol{u}}\cdot\boldsymbol{x})\hat{\boldsymbol{u}} + \sin\theta\hat{\boldsymbol{u}}\times\boldsymbol{x} \end{bmatrix}$$

□

Just as rotation and transformation matrices are related via matrix transposition, rotation and transformation quaternions are related via quaternion conjugation:

Theorem A.21. *The left transformation unit quaternion*

$$\mathcal{Q}_{A\to B} = \begin{bmatrix} \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix} \tag{A.34}$$

transforms a three-vector $\boldsymbol{x}_A$ expressed in frame A to frame B whose axes are rotated about the $\hat{\boldsymbol{u}}$ axis by an angle of θ relative to the original frame A via:

$$\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix} = \mathcal{Q}_{A\to B} \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} \mathcal{Q}^{\star}_{A\to B} \tag{A.35}$$

Proof. Expanding and simplifying equation (A.35) results in

$$\mathcal{Q}_{A\to B} \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} \mathcal{Q}^{\star}_{A\to B} = \begin{bmatrix} 0 \\ \cos\theta \boldsymbol{x}_A + (1-\cos\theta)(\hat{\boldsymbol{u}}\cdot\boldsymbol{x}_A)\hat{\boldsymbol{u}} - \sin\theta\hat{\boldsymbol{u}}\times\boldsymbol{x}_A \end{bmatrix}$$

□

Consider a left transformation unit quaternion as defined in equation (A.34).

Theorem A.22. *The quaternion logarithm of a left transformation unit quaternion*

$$\mathcal{Q} = \begin{bmatrix} \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix}$$

is

$$\log\mathcal{Q} = \begin{bmatrix} 0 \\ -\frac{1}{2}\theta\hat{\boldsymbol{u}} \end{bmatrix} \tag{A.36}$$

Proof. The theorem follows immediately from theorem A.17. □

Right Rotation and Transformation Quaternions

One unfortunate aspect of quaternions is that they magnify the confusion regarding *rotation* and *transformation*. The conjugate of a *left* rotation or transformation quaternion can also be used as the basis for rotation or transformation. For example, the *right rotation unit quaternion*

$$\mathcal{Q}_{rot,\text{right}} = \begin{bmatrix} \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2}\hat{\boldsymbol{u}} \end{bmatrix}$$

rotates a three-vector $\boldsymbol{x}$ about the $\hat{\boldsymbol{u}}$ axis by an angle of θ via:

$$\begin{bmatrix} 0 \\ \boldsymbol{x}' \end{bmatrix} = {\mathcal{Q}^{\star}}_{rot,\text{right}} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \mathcal{Q}_{rot,\text{right}}$$

which yields the same value for $\boldsymbol{x}'$ as does equation (A.33).

The Quaternion Model uses left transformation unit quaternions because they chain in the same manner as do transformation matrices and because the dynamics package is concerned with transformations, not rotations.

Chains of Transformations and Rotations

Transformation matrices chain from right to left: given a pair transformations $\boldsymbol{T}_{A\to B}$ and $\boldsymbol{T}_{B\to C}$ from frame A to frame B and from frame B to frame C, the transformation from frame A to frame C is

$$\boldsymbol{T}_{A\to C} = \boldsymbol{T}_{B\to C}\,\boldsymbol{T}_{A\to B} \tag{A.37}$$

Left transformation quaternions similarly chain from right to left:

Theorem A.23.

$$\mathcal{Q}_{A\to C} = \mathcal{Q}_{B\to C}\,\mathcal{Q}_{A\to B} \tag{A.38}$$

Proof. A vector $\boldsymbol{x}_A$ transforms from frame A to frame B and from frame B to C via equation (A.35):

$$\begin{aligned}
\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix} &= \mathcal{Q}_{A\to B} \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} \mathcal{Q}^{\star}_{A\to B} \\
\begin{bmatrix} 0 \\ \boldsymbol{x}_C \end{bmatrix} &= \mathcal{Q}_{B\to C} \begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix} \mathcal{Q}^{\star}_{B\to C} \\
&= \mathcal{Q}_{B\to C} \left(\mathcal{Q}_{A\to B} \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} \mathcal{Q}^{\star}_{A\to B} \right) \mathcal{Q}^{\star}_{B\to C}
\end{aligned}$$

Since quaternion multiplication is associative,

$$\begin{aligned}
\begin{bmatrix} 0 \\ \boldsymbol{x}_C \end{bmatrix} &= (\mathcal{Q}_{B\to C}\,\mathcal{Q}_{A\to B}) \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} (\mathcal{Q}^{\star}_{A\to B}\,\mathcal{Q}^{\star}_{B\to C}) \\
&= (\mathcal{Q}_{B\to C}\,\mathcal{Q}_{A\to B}) \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} (\mathcal{Q}_{B\to C}\,\mathcal{Q}_{A\to B})^{\star}
\end{aligned}$$

The direct transformation from frame A to frame C is

$$\begin{bmatrix} 0 \\ \boldsymbol{x}_C \end{bmatrix} = \mathcal{Q}_{A\to C} \begin{bmatrix} 0 \\ \boldsymbol{x}_A \end{bmatrix} \mathcal{Q}^{\star}_{A\to C}$$

from which

$$\mathcal{Q}_{A\to C} = \mathcal{Q}_{B\to C}\,\mathcal{Q}_{A\to B}$$

$\square$

Rotation matrices chain from left to right: Given a pair of rotation matrices $\boldsymbol{R}_{rot_1}$ and $\boldsymbol{R}_{rot_2}$, the rotation matrix that represents performing rotation rot_2 after performing rotation rot_1 is

$$\boldsymbol{R}_{rot_{1+2}} = \boldsymbol{R}_{rot_1}\,\boldsymbol{R}_{rot_2} \tag{A.39}$$

Left rotation quaternions still chain from right to left:

$$\mathcal{Q}_{rot_{1+2}} = \mathcal{Q}_{rot_2}\,\mathcal{Q}_{rot_1} \tag{A.40}$$

A.5.3 Quaternions and Alternate Representation Schemes

Single-Axis Rotations

Quaternions are closely related to a single axis rotation. Given a rotation about some axis $\hat{\boldsymbol{u}}$ by an angle θ that describes the relative orientation of two reference frames, one merely need apply equation (A.34) to form the corresponding left transformation unit quaternion.

The next theorem establishes a constructive technique for determining the single axis rotation given a left transformation unit quaternion.

Theorem A.24. *Given a left transformation unit quaternion $\mathcal{Q}$ and its decomposition into a unit vector $\hat{\boldsymbol{u}}$ and angle θ' per theorem A.11, The single axis rotation corresponding to $\mathcal{Q}$ is a rotation of an angle $\theta = -2\theta'$ about the $\hat{\boldsymbol{u}}$ axis.*

Proof. Forming the left transformation unit quaternion from the single axis rotation θ about $\hat{\boldsymbol{u}}$ per equation (A.34) yields the original left transformation unit quaternion $\mathcal{Q}$. □

Transformation Matrices

Theorem A.25. *Given a single axis rotation about the $\hat{\boldsymbol{u}}$ axis by an angle of θ from reference frame A to reference frame B, the i, j^{th} element of the transformation matrix $\boldsymbol{T}_{A\to B}$ from frame A to frame B is*

$$\boldsymbol{T}_{A\to B\,ij} = \cos\theta\,\delta_{ij} + (1-\cos\theta)u_i u_j + \sin\theta \sum_k \epsilon_{ijk} u_k \tag{A.41}$$

where

δ_{ij} *is the Kronecker delta*

ϵ_{ijk} *is the permutation symbol:*

$$\epsilon_{ijk} = \begin{cases} 1 & \text{if } i,\ j, \text{ and } k \text{ are an even permutation of } (1,2,3), \\ -1 & \text{if } i,\ j, \text{ and } k \text{ are an odd permutation of } (1,2,3), \\ 0 & \text{otherwise } (i=j,\ i=k, \text{ or } j=k). \end{cases}$$

Proof. The i^{th} row of $\boldsymbol{T}_{A\to B}$ contains the transpose of the unit vector $\hat{\boldsymbol{e}}_i$ rotated about the $\hat{\boldsymbol{u}}$ axis by an angle of θ. (The unit vector $\hat{\boldsymbol{e}}_i$ contains a one in row i and zeros elsewhere: $\hat{\boldsymbol{e}}_{i_j} = \delta_{ij}$). Applying equation (A.30) to $\hat{\boldsymbol{e}}_i$ and representing the cross product of two vectors a and b as $(a\times b)_j = \sum_i \sum_k \epsilon_{ijk} a_k b_i$ results in equation (A.41). □

Theorem A.26. *Given a left transformation unit quaternion from reference frame A to reference frame B $\mathcal{Q}_{A\to B} = \begin{bmatrix} q_s \\ \boldsymbol{q_v} \end{bmatrix}$, the transformation matrix corresponding to $\mathcal{Q}_{A\to B}$ is*

$$\boldsymbol{T}_{A\to B\,ij} = (2{q_s}^2 - 1)\delta_{ij} + 2(q_{v_i} q_{v_j} - \sum_k \epsilon_{ijk} q_s q_{v_k}) \tag{A.42}$$

Proof. By using the half-angle formulae

$$\cos\theta = 2\cos^2\frac{\theta}{2} - 1 = 1 - 2\sin^2\frac{\theta}{2}$$
$$\sin\theta = 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}$$

equation (A.41) becomes

$$\boldsymbol{T}_{A\to Bij} = (2\cos^2\frac{\theta}{2} - 1)\delta_{ij} + 2\sin^2\frac{\theta}{2}u_i u_j + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\sum_k \epsilon_{ijk}u_k$$

Per equation (A.34),

$$q_s = \cos\frac{\theta}{2}$$
$$\boldsymbol{q_v} = -\sin\frac{\theta}{2}\hat{\boldsymbol{u}}$$

by which equation (A.41) further reduces to equation (A.42). □

Theorem A.27. *Given a transformation matrix from reference frame A to reference frame B $\boldsymbol{T}_{A\to B}$ the left transformation unit quaternion $\mathcal{Q}_{A\to B}$ with scalar and vector parts q_s and $\boldsymbol{q_v}$ corresponding to $\boldsymbol{T}_{A\to B}$ is given by four methods labeled q_s and $q_{v_i}, i \in (0,1,2)$.*

Defining

$$tr \equiv \operatorname{tr}(\boldsymbol{T}_{A\to B}) \tag{A.43}$$
$$t_i \equiv \boldsymbol{T}_{A\to Bii} - (\boldsymbol{T}_{A\to Bjj} + \boldsymbol{T}_{A\to Bkk}) \tag{A.44}$$
$$d_k \equiv \boldsymbol{T}_{A\to Bji} - \boldsymbol{T}_{A\to Bij} \quad (\epsilon_{ijk} = 1) \tag{A.45}$$
$$s_{ij} \equiv \boldsymbol{T}_{A\to Bji} + \boldsymbol{T}_{A\to Bij} \quad (i \neq j) \tag{A.46}$$

Method q_s is

$$\left.\begin{aligned} f_1 &= \sqrt{tr+1} \\ f_2 &= \frac{1}{2f_1} \\ q_s &= \frac{1}{2}f_1 \\ q_{v_i} &= d_i f_2 \\ q_{v_j} &= d_j f_2 \\ q_{v_k} &= d_k f_2 \end{aligned}\right\} \tag{A.47}$$

Methods $q_{v_i}, i \in (0,1,2)$ *are*

$$\left.\begin{aligned} f_1 &= \sqrt{t_i+1} \\ f_2 &= \frac{1}{2f_1} \\ q_{v_i} &= \frac{1}{2}f_1 \\ q_{v_j} &= s_{ij}f_2 \\ q_{v_k} &= s_{ik}f_2 \\ q_s &= d_i f_2 \end{aligned}\right\} \tag{A.48}$$

Proof. By equation (A.42), the terms tr, t_i, d_k and s_{ij} are

$$tr = \mathrm{tr}(\boldsymbol{T}_{A\to B}) = 4{q_s}^2 - 1 \tag{A.49}$$

$$t_i = \boldsymbol{T}_{A\to B\,ii} - (\boldsymbol{T}_{A\to B\,jj} + \boldsymbol{T}_{A\to B\,kk}) = 4q_{v_i}^2 - 1 \tag{A.50}$$

$$d_k = \boldsymbol{T}_{A\to B\,ji} - \boldsymbol{T}_{A\to B\,ij}\ (\epsilon_{ijk} = 1) = 4q_s q_{v_k} \tag{A.51}$$

$$s_{ij} = \boldsymbol{T}_{A\to B\,ji} + \boldsymbol{T}_{A\to B\,ij}\ (i \neq j) = 4q_{v_i}q_{v_j} \tag{A.52}$$

from which equations (A.47) and (A.48) are readily derived. □

Euler Angles

Euler proved in 1776 that any arbitrary rotation or transformation can be described by only three parameters. An Euler sequence comprises a sequence of rotations by various angles about various axes in a specified order. For example, a yaw-pitch-roll transformation sequence involves a transformation about the $z-$axis followed by a second transformation about the transformed $y-$axis followed by a final rotation about the doubly-transformed $x-$axis.

Theorem A.28. *Given an Euler sequence comprising:*

- *A rotation through an angle* θ_1 *counterclockwise about the* $\hat{\boldsymbol{u}}_1$ *axis followed by*
- *A rotation through an angle* θ_2 *counterclockwise about the* $\hat{\boldsymbol{u}}_2$ *axis followed by*
- *A rotation through an angle* θ_3 *counterclockwise about the* $\hat{\boldsymbol{u}}_3$ *axis*

that represents the transformation from from reference frame A *to reference frame* B*, the corresponding left transformation unit quaternion* $\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3)$ *is*

$$\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3) = \mathcal{Q}(\theta_3, \hat{\boldsymbol{u}}_3)\,\mathcal{Q}(\theta_2, \hat{\boldsymbol{u}}_2)\,\mathcal{Q}(\theta_1, \hat{\boldsymbol{u}}_1) \tag{A.53}$$

where

$$\mathcal{Q}(\theta_1, \hat{\boldsymbol{u}}_1) \equiv \begin{bmatrix} \cos\frac{\theta_1}{2} \\ -\sin\frac{\theta_1}{2}\hat{\boldsymbol{u}}_1 \end{bmatrix} \tag{A.54}$$

$$\mathcal{Q}(\theta_2, \hat{\boldsymbol{u}}_2) \equiv \begin{bmatrix} \cos\frac{\theta_2}{2} \\ -\sin\frac{\theta_2}{2}\hat{\boldsymbol{u}}_2 \end{bmatrix} \tag{A.55}$$

$$\mathcal{Q}(\theta_3, \hat{\boldsymbol{u}}_3) \equiv \begin{bmatrix} \cos\frac{\theta_3}{2} \\ -\sin\frac{\theta_3}{2}\hat{\boldsymbol{u}}_3 \end{bmatrix} \tag{A.56}$$

Proof. The left transformation quaternions corresponding to the individual rotations, $\mathcal{Q}(\theta_1, \hat{\boldsymbol{u}}_1)$, $\mathcal{Q}(\theta_2, \hat{\boldsymbol{u}}_2)$, and $\mathcal{Q}(\theta_3, \hat{\boldsymbol{u}}_3)$, follow from equation (A.34). The product follows from equation (A.38). $\square$

The Trick package is only concerned with Euler sequences that use some permutation of rotations about the x, y, and z axes. (Trick does not cover the standard astronaumical $\phi - \theta - \psi$ sequence.) With this restriction, equation (A.53) can be reduced to two cases: one if the unit vectors $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ form an even permutation of (x, y, z) and another if the unit vectors $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ form an odd permutation of (x, y, z).

Theorem A.29. *Given an Euler sequence θ_1 about the $\hat{\boldsymbol{u}}_1$ axis, θ_2 about the $\hat{\boldsymbol{u}}_2$ axis, θ_3 about the $\hat{\boldsymbol{u}}_3$ axis, where $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ is an even permutation of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, that represents the transformation from from reference frame A to reference frame B, the corresponding left transformation unit quaternion $\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3)$ is*

$$\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3) = \begin{bmatrix} \cos\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_1}{2} - \sin\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\sin\frac{\theta_1}{2} \\ -\begin{pmatrix} (\cos\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_1}{2} + \sin\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_1}{2})\hat{\boldsymbol{u}}_1 \\ +(\cos\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_1}{2} - \sin\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_1}{2})\hat{\boldsymbol{u}}_2 \\ +(\cos\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\sin\frac{\theta_1}{2} + \sin\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_1}{2})\hat{\boldsymbol{u}}_3 \end{pmatrix} \end{bmatrix} \tag{A.57}$$

Proof. This follows directly by expanding equation (A.53) for the special case of a set of orthogonal unit vectors $\hat{\boldsymbol{u}}_1$, $\hat{\boldsymbol{u}}_2$, $\hat{\boldsymbol{u}}_3$ for which $\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_3$. $\square$

Theorem A.30. *Given an Euler sequence θ_1 about the $\hat{\boldsymbol{u}}_1$ axis, θ_2 about the $\hat{\boldsymbol{u}}_2$ axis, θ_3 about the $\hat{\boldsymbol{u}}_3$ axis, where $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ is an odd permutation of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, that represents the transformation from from reference frame A to reference frame B, the corresponding left transformation unit quaternion $\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3)$ is*

$$\mathcal{Q}_{123}(\theta_1, \hat{\boldsymbol{u}}_1; \theta_2, \hat{\boldsymbol{u}}_2; \theta_3, \hat{\boldsymbol{u}}_3) = \begin{bmatrix} \cos\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_1}{2} + \sin\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\sin\frac{\theta_1}{2} \\ -\begin{pmatrix} (\cos\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_1}{2} - \sin\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_1}{2})\hat{\boldsymbol{u}}_1 \\ +(\cos\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_1}{2} + \sin\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_1}{2})\hat{\boldsymbol{u}}_2 \\ -(\cos\frac{\theta_3}{2}\sin\frac{\theta_2}{2}\sin\frac{\theta_1}{2} - \sin\frac{\theta_3}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_1}{2})\hat{\boldsymbol{u}}_3 \end{pmatrix} \end{bmatrix} \tag{A.58}$$

Proof. This follows directly by expanding equation (A.53) for the special case of a set of orthogonal unit vectors $\hat{\boldsymbol{u}}_1$, $\hat{\boldsymbol{u}}_2$, $\hat{\boldsymbol{u}}_3$ for which $\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{u}}_2 = -\hat{\boldsymbol{u}}_3$. $\square$

With a bit of trigonemtric manipulation, the inverse operation of determining the Euler angles given a quaternion and a rotation sequence follow from theorems A.29 and A.30.

Theorem A.31. *Given an even permutation $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ and a quaternion $\mathcal{Q} = \begin{bmatrix} q_s \\ q_{v_1}\hat{\boldsymbol{u}}_1 + q_{v_2}\hat{\boldsymbol{u}}_2 + q_{v_3}\hat{\boldsymbol{u}}_3 \end{bmatrix}$, the Euler sequence θ_1 about the $\hat{\boldsymbol{u}}_1$ axis, θ_2 about the $\hat{\boldsymbol{u}}_2$ axis, θ_3 about the $\hat{\boldsymbol{u}}_3$ axis is given by*

$$\theta_2 = \arcsin(-2(q_s q_{v_2} - q_{v_1} q_{v_3})) \tag{A.59}$$

$$\theta_1 = \arctan(-2(q_s q_{v_1} + q_{v_2} q_{v_3}), ({q_s}^2 - q_{v_2}^2) - (q_{v_1}^2 - q_{v_3}^2)) \tag{A.60}$$

$$\theta_3 = \arctan(-2(q_s q_{v_3} + q_{v_2} q_{v_1}), ({q_s}^2 - q_{v_2}^2) - (q_{v_3}^2 - q_{v_1}^2)) \tag{A.61}$$

Equations (A.60) and (A.61) are valid only if $|\sin\theta_2| \neq 1$. $|\sin\theta_2| = 1$ represents a singularity, in which case θ_1 and θ_3 are related via

$$\theta_1 + \theta_3 \sin\theta_2 = 2\arctan(-q_{v_3}\sin\theta_2, q_{v_2}\sin\theta_2) = 2\arctan(-q_{v_1}, q_s) \tag{A.62}$$

Proof. Th solution for θ_2 and the non-singular solutions for θ_1 and θ_3 follow by replacing the arguments of the inverse trigonometric functions in equations (A.59) to (A.61) into equation (A.57). The singular solutions follow by replacing $\theta_2 = \pm\frac{\pi}{2}$ into equation (A.57). □

Theorem A.32. *Given an odd permutation $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3)$ of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ and a quaternion $\mathcal{Q} = \begin{bmatrix} q_s \\ q_{v_1}\hat{\boldsymbol{u}}_1 + q_{v_2}\hat{\boldsymbol{u}}_2 + q_{v_3}\hat{\boldsymbol{u}}_3 \end{bmatrix}$, the Euler sequence θ_1 about the $\hat{\boldsymbol{u}}_1$ axis, θ_2 about the $\hat{\boldsymbol{u}}_2$ axis, θ_3 about the $\hat{\boldsymbol{u}}_3$ axis is given by*

$$\theta_2 = \arcsin(-2(q_s q_{v_2} + q_{v_1} q_{v_3})) \tag{A.63}$$

$$\theta_1 = \arctan(-2(q_s q_{v_1} - q_{v_2} q_{v_3}), ({q_s}^2 - q_{v_2}^2) - (q_{v_1}^2 - q_{v_3}^2)) \tag{A.64}$$

$$\theta_3 = \arctan(-2(q_s q_{v_3} - q_{v_2} q_{v_1}), ({q_s}^2 - q_{v_2}^2) - (q_{v_3}^2 - q_{v_1}^2)) \tag{A.65}$$

Equations (A.64) and (A.65) are valid only if $|\sin\theta_2| \neq 1 . |\sin\theta_2| = 1$ represents a singularity, in which case θ_1 and θ_3 are related via

$$\theta_1 - \theta_3 \sin\theta_2 = 2\arctan(q_{v_3}\sin\theta_2, -q_{v_2}\sin\theta_2) = 2\arctan(-q_{v_1}, q_s) \tag{A.66}$$

Proof. Th solution for θ_2 and the non-singular solutions for θ_1 and θ_3 follow by replacing the arguments of the inverse trigonometric functions in equations (A.63) to (A.65) into equation (A.58). The singular solutions follow by replacing $\theta_2 = \pm\frac{\pi}{2}$ into equation (A.58). □

A.5.4 Comparing and Averaging Transformation Quaternions

Suppose $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two vectors that represent the some comparable quantity such as the position of a spacecraft at a specific point in time as computed by two different methods or the position of a spacecraft at two different points in time. Various analyses are frequently based on the difference between the two vectors, $\boldsymbol{\Delta x} \equiv \boldsymbol{x}_2 - \boldsymbol{x}_1$.

Now suppose $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are two left transformation quaternions that that represent some comparable transformation. The additive difference between the two transformation quaternions has no physical meaning and thus is not useful for analysis. The normative mathematics of the transformation quaternions is multiplication rather than addition. A multiplicative rather than additive scheme is needed to determine the "distance" between two quaternions.

Definition A.12 (Quaternion difference)**:**
Given quaternions $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the difference between the two quaternions is defined as

$$\Delta\mathcal{Q} \equiv \ \mathrm{acute}(\mathcal{Q}_2\,\mathcal{Q}_1^\star) \tag{A.67}$$

where *acute* denotes that all components of the product $\mathcal{Q}_2\,\mathcal{Q}_1^\star$ are to be negated when the scalar part of the product is negative.

The magnitude of an error vector is often a more significant indicator than is the direction of the error when analyzing errors in vectors. Similarly, the single axis rotation angle computed from the quaternion difference is often a more significant indicator than the direction of the rotation when analyzing errors in transformation quaternions.

The difference between two vectors is also useful for computing a weighted average: $\bar{\boldsymbol{x}} = w\boldsymbol{\Delta x} + \boldsymbol{x}_1$ where w is some weight factor between 0 and 1. The arithmetic mean results when $w = 1/2$.

Similarly, the weighted mean of two quaternions is

Definition A.13 (Quaternion weighted mean)**:**
Given quaternions $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the weighted mean of the two quaternions is defined as

$$\bar{\mathcal{Q}} = (\Delta\mathcal{Q})^w\,\mathcal{Q}_1 \tag{A.68}$$

where w is a weight factor between 0 and 1 and $\Delta\mathcal{Q}$ is computed via equation (A.67).

Let

$$\Delta\mathcal{Q} = \begin{bmatrix} \Delta q_s \\ \boldsymbol{\Delta q_v} \end{bmatrix}$$

if $|\Delta q_s| \approx 1$ then $(\Delta\mathcal{Q})^w$ can be approximated using the small angle approximations

$$\begin{aligned}
\cos(w\theta) &\approx 1 + w^2(\cos\theta - 1) \\
\sin(w\theta) &\approx w\sin\theta \\
(\Delta\mathcal{Q})^w &\approx \begin{bmatrix} 1 + w^2(\Delta q_s - 1) \\ w\boldsymbol{\Delta q_v} \end{bmatrix}
\end{aligned}$$

A.6 Quaternion Time Derivative

This section develops the time derivative of a left transformation quaternion.

The time derivative of a vector $\boldsymbol{x}$ is observer-dependent. The relation between the time derivative of $\boldsymbol{x}$ as observed in an inertial frame I and the time derivative of $\boldsymbol{x}$ as observed in a frame B rotating at a rate ω with respect to the inertial frame[2] is

$$\overset{\cdot I}{\boldsymbol{x}} = \overset{\cdot B}{\boldsymbol{x}} + \boldsymbol{\omega} \times \boldsymbol{x} \tag{A.69}$$

where

$$\overset{\cdot F}{\boldsymbol{x}} \text{ is the time derivative of } \boldsymbol{x} \text{ as observed in frame } F \tag{A.70}$$

This can be expressed in quaternion form as

$$\mathcal{Q}_{I\to B}\left(\frac{d}{dt}\left(\mathcal{Q}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\mathcal{Q}_{I\to B}\right)\right)\mathcal{Q}^{\star}_{I\to B} = \begin{bmatrix}0\\ \dot{\boldsymbol{x}}_B\end{bmatrix} + \begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\times \boldsymbol{x}_B\end{bmatrix} \tag{A.71}$$

where

$\mathcal{Q}_{I\to B}$ is the left transformation quaternion from frame I to frame B

$\boldsymbol{\omega}_{B:I\to B}$ is the angular velocity of frame B with respect to frame I, expressed in frame B

$\boldsymbol{x}_B$ is an arbitrary vector $\boldsymbol{x}$, expressed in frame B

The derivative in the left-hand side of equation (A.71) expands to

$$\begin{aligned}&\frac{d}{dt}\left(\mathcal{Q}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\mathcal{Q}_{I\to B}\right) = \\ &\quad \dot{\mathcal{Q}}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\mathcal{Q}_{I\to B} + \mathcal{Q}^{\star}_{I\to B}\begin{bmatrix}0\\ \dot{\boldsymbol{x}}_B\end{bmatrix}\mathcal{Q}_{I\to B} - \left(\dot{\mathcal{Q}}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\mathcal{Q}_{I\to B}\right)^{\star}\end{aligned} \tag{A.72}$$

Applying the above to the left-hand side of equation (A.71) and simplifying yields

$$\begin{aligned}&\mathcal{Q}_{I\to B}\left(\frac{d}{dt}\left(\mathcal{Q}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\mathcal{Q}_{I\to B}\right)\right)\mathcal{Q}^{\star}_{I\to B} = \\ &\quad \mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix} - \left(\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^{\star}_{I\to B}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\right)^{\star} + \begin{bmatrix}0\\ \dot{\boldsymbol{x}}_B\end{bmatrix}\end{aligned} \tag{A.73}$$

Using

$$\begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\times \boldsymbol{x}_B\end{bmatrix} = \frac{1}{2}\left(\left(\begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\right) - \left(\begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\right)^{\star}\right) \tag{A.74}$$

The right-hand side of equation (A.71) expands to

$$\begin{aligned}&\frac{d}{dt}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix} + \begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\times \boldsymbol{x}_B\end{bmatrix} = \\ &\quad \frac{d}{dt}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix} + \frac{1}{2}\left(\left(\begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\right) - \left(\begin{bmatrix}0\\ \boldsymbol{\omega}_{B:I\to B}\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{x}_B\end{bmatrix}\right)^{\star}\right)\end{aligned} \tag{A.75}$$

Equating equations (A.73) and (A.75) and eliminating the common term $\begin{bmatrix} 0 \\ \dot{\boldsymbol{x}}_B \end{bmatrix}$ yields

$$\begin{gathered} \mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B}\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix} - \left(\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B}\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix}\right)^\star = \\ \left(\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix}\right) - \left(\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\begin{bmatrix} 0 \\ \boldsymbol{x}_B \end{bmatrix}\right)^\star \end{gathered} \tag{A.76}$$

Since equation (A.76) must be satisfied for *any* vector $\boldsymbol{x}$,

$$\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B} - \left(\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B}\right)^\star = \left(\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\right) - \left(\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\right)^\star \tag{A.77}$$

Note that equation (A.77) is of the form

$$\mathcal{Q}_a - \mathcal{Q}^\star{}_a = \mathcal{Q}_b - \mathcal{Q}^\star{}_b$$

Such a form requires that the vector parts of $\mathcal{Q}_a$ and $\mathcal{Q}_b$ be equal; the scalar parts are unconstrained. Since the scalar parts of $\mathcal{Q}_{I\to B}\dot{\mathcal{Q}}^\star_{I\to B}$ and $\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix}$ are zero, equation (A.77) reduces to

$$\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B} = \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix} \tag{A.78}$$

or

$$\dot{\mathcal{Q}}_{I\to B} = \begin{bmatrix} 0 \\ -\frac{1}{2}\boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\mathcal{Q}_{I\to B} \tag{A.79}$$

The Quaternion Model uses equation (A.79) to form the derivative of the inertial-to-body left transformation quaternion $\mathcal{Q}_{I\to B}$.

Solving equation (A.79) for the body rate yields

$$\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{B:I\to B} \end{bmatrix} = 2\mathcal{Q}_{I\to B}\,\dot{\mathcal{Q}}^\star_{I\to B} \tag{A.80}$$

Differentiating equation (A.79) again yields the second derivative of the inertial-to-body left transformation quaternion $\mathcal{Q}_{I\to B}$:

$$\ddot{\mathcal{Q}}_{I\to B} = \begin{bmatrix} 0 \\ -\frac{1}{2}\dot{\boldsymbol{\omega}}_{B:I\to B} \end{bmatrix}\mathcal{Q}_{I\to B} + \begin{bmatrix} 0 \\ -\frac{1}{2}\boldsymbol{\omega}_{B:I\to B} \end{bmatrix}\dot{\mathcal{Q}}_{I\to B} \tag{A.81}$$

$$= \begin{bmatrix} -\frac{1}{4}\|\boldsymbol{\omega}_{B:I\to B}\|^2 \\ -\frac{1}{2}\dot{\boldsymbol{\omega}}_{B:I\to B} \end{bmatrix}\mathcal{Q}_{I\to B} \tag{A.82}$$

A.7 Attitude Propagation

Suppose the body rate vector $\boldsymbol{\omega}_{B:I\to B}$ is constant. The quaternion time derivative equation (A.79) then takes on the form

$$\dot{x} = -Ax \tag{A.83}$$

When x is a scalar or a vector and A is a scalar or matrix, equations of this form have a solution

$$x(t0 + dt) = \exp(-Adt)x(t0) \tag{A.84}$$

Due to the developments in preceding sections, the same form of solution applies when A is a pure imaginary quaternion and x is a quaternion, and thus the constant body rate solution to equation (A.79) is

$$\begin{aligned} \mathcal{Q}_{I\to B}(t+dt) &= \exp\left(\begin{bmatrix} 0 \\ -\frac{1}{2}\boldsymbol{\omega}_{B:I\to B}dt \end{bmatrix}\right)\mathcal{Q}_{I\to B}(t0) \\ &= \begin{bmatrix} \cos\frac{\omega dt}{2} \\ -\sin\frac{\omega dt}{2}\hat{\boldsymbol{\omega}} \end{bmatrix}\mathcal{Q}_{I\to B}(t0) \end{aligned} \tag{A.85}$$

where

$$\begin{aligned} \omega &\equiv \|\boldsymbol{\omega}_{B:I\to B}\| \\ \hat{\boldsymbol{\omega}} &\equiv \frac{\boldsymbol{\omega}_{B:I\to B}}{\omega} \end{aligned}$$

For a sufficiently small time step dt, the body rate will be approximately constant. Equation (A.85) can thus be used as the basis for quaternion propagation. However, since this equation involves the use of transcendental functions, applying this equation directly over small time steps would be computationally prohibitive.

A simple approach to avoiding transcendental functions is to make the first order small angle assumptions

$$\cos(\frac{\omega dt}{2}) \approx 1 \tag{A.86}$$

$$\sin(\frac{\omega dt}{2}) \approx \frac{\omega dt}{2} \tag{A.87}$$

in which case equation (A.85) becomes

$$\begin{aligned} \mathcal{Q}_{I\to B}(t+dt) &= \begin{bmatrix} 1 \\ -\frac{1}{2}\boldsymbol{\omega}_{B:I\to B}dt \end{bmatrix}\mathcal{Q}_{I\to B}(t0) \\ &= \mathcal{Q}_{I\to B}(t0) + \dot{\mathcal{Q}}_{I\to B}dt \end{aligned} \tag{A.88}$$

A standard numerical integrator can be used to propagate a quaternion via equation (A.88). The first order small angle assumptions thus lead to a simple and very appealing propagator.

A.8 Body Rate Propagation

For a sufficiently small time step dt, the body rate will be approximately constant. However, the body rate for most spacecraft will rarely be constant for any extended period of time. The body rate must be propagated along with the attitude.

The rotational analog of Newton's Second Law is [2]

$$\dot{\boldsymbol{L}}_{I:I\to B} = \boldsymbol{\tau}_{I:ext} \tag{A.89}$$

where

$\dot{\boldsymbol{L}}_{I:I\to B}$ is the body's angular momentum vector and

$\boldsymbol{\tau}_{I:ext}$ is the external torque acting on the body.

The angular momentum is related to the angular velocity via

$$\boldsymbol{L}_{I\to B} = \boldsymbol{I}\,\boldsymbol{\omega}_{I\to B} \tag{A.90}$$

where

$\boldsymbol{I}$ is the body's inertia tensor.

Note that equation (A.89) is valid in an inertial frame only while equation (A.90) is valid in any reference frame (but $\boldsymbol{L}$, $\boldsymbol{I}$, and $\boldsymbol{\omega}$ must all represented in the same reference frame).

Applying equation (A.69) to equations (A.89) and transforming to the body frame yields the body-frame rotational equations of motion

$$\dot{\boldsymbol{\omega}}_{B:I\to B} = {\boldsymbol{I}_B}^{-1}\left(\boldsymbol{\tau}_{B:ext} + (\boldsymbol{I}_B\,\boldsymbol{\omega}_{B:I\to B}) \times \boldsymbol{\omega}_{B:I\to B}\right) \tag{A.91}$$