ARL-TR-10276 ● JAN 2026

Solving the Euler Angle Singularity Problem Using Vector or Quaternion Algebra, without Algorithm Switching

by Mark Bundy

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-10276 ● JAN 2026

Solving the Euler Angle Singularity Problem Using Vector or Quaternion Algebra, without Algorithm Switching

Mark Bundy
DEVCOM Army Research Laboratory

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
January 2026	Technical Report	July 2022	July 2025

4. TITLE AND SUBTITLE
Solving the Euler Angle Singularity Problem Using Vector or Quaternion Algebra, without Algorithm Switching

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER

6. AUTHOR(S)
Mark Bundy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLA-VB
Aberdeen Proving Ground, MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-10276

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report reviews the mathematical origins of, and gives context to, what is traditionally called the “Euler angle singularity problem,” which arises when seeking to track or simulate the motion of a rotating body relative to an earth-fixed frame of reference, based on input from body-fixed angular rate sensors. The singularity problem centers around division by zero in the solution for the earth-referenced angular rotation rates of the body when one of the Euler tracking angles reaches $\pm\pi/2$. The novelty of this report is the demonstration of a methodology for solving the singularity problem for any possible Euler angle 1) without incurring the infinity-causing, division-by-zero condition, and 2) without using quaternion algebra or changing the angle-axes sequencing (a.k.a. algorithm switching). The solution centers around making the iteration time step for solving the nonlinear kinematic equations of motion proportional to, and thus canceling out, the factor that would otherwise produce division by zero in the angular rate equations. Numerical simulations are used to validate this Euler-angle, vector algebra-based solution, versus solutions formulated using 1) quaternion algebra, 2) the vector angular velocity cross product equation (introduced in physics texts), and 3) algorithm switching. Moreover, the report shows how the various formulations, vector, quaternion, and algorithm switching are derived and relatable to each other, explaining what often appears to be plus or minus sign differences in referenced equations, which can be disconcerting to “students” of the subject. A discussion of computational efficiency is also included, with numerical simulations showing that it may be possible, depending on how trigonometric functions and power laws are evaluated (e.g., using memory-based look-up tables), for vector algebra solutions to be faster than quaternion algebra solutions. An overview of the highlights and findings is provided in the summary and conclusions.

15. SUBJECT TERMS
Euler, angle, singularity, quaternion, numerical, solution, efficiency, Mechanical Sciences

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	100

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Mark Bundy	(520) 691-6379

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

Planning, tracking, and control of rotating objects is common in engineering and computer graphic applications. For example, describing the time-dependent attitude of an air platform or a robotic arm relative to an earth-fixed coordinate frame is of engineering interest. However, the traditional mathematical approach for describing such motion often caveats the range of its trusted solution to avoid what appears to be a mathematical singularity, where the angular rates used to track the attitude/orientation of a rotating body would become infinite, and therefore undefined at a particular body orientation. We will present and discuss some of the key historical figures and mathematical approaches they used to describe and solve the kinematic equations of motion using vector- and quaternion-based algebras. The research question to be answered is not whether the mathematics would indicate a singularity at a certain attitude, but whether that singularity-causing attitude is ever reached. We show that if the time-step used to solve the nonlinear system of equations is made proportional to the factor that causes division by zero, the singularity condition is never reached, and thus the apparent problem is avoided or rendered moot.

The mathematics of rotational kinematics was widely investigated and developed for the better part of the 19th century by notable theorists such as Olinde Rodrigues (1795–1851, single axis-angle rotation formula), Leonhard Euler (1707–1783, multiple axis-angle rotation formula/matrix), and William Rowan Hamilton (1805–1865, credited with applying quaternion algebra to describe rotational problems).

First, a brief review will be given of the various methods used to describe the changing attitude (angular orientation) of a rotating vector and body frame of reference relative to a fixed (a.k.a. world, or earth) frame of reference. Doing so will involve a discussion of coordinate frames, vectors, angles and angular rates, and rules for multiplication and symbolic operations on such vectors. These rules and relationships can be compartmentalized into trigonometry, geometry, algebra, and differential and integral calculus. It will suffice here to say that trigonometry deals with the relationship between lines and angles (e.g., parallel, perpendicular, intersection); geometry describes attributes of line segments and shapes (e.g., magnitude and direction of vectors, areas of dimensionally-bounded shapes); algebra involves mathematical operations on the objects of geometry (e.g., dot and cross product on vectors, matrix multiplication); calculus facilitates the description of instantaneous as well as cumulative changes in the magnitude and direction of line segments and vectors. Throughout this report, emphasis will be placed on offering physical interpretations (often through the use of figures) of geometric quantities and algebraic operations (including asking and answering why the physical interpretation is or is not what we should expect). Moreover, the order of sectional topics is based upon whatever question, answer, or insight was gained from the previous section.

Since most introductory engineering and physics courses present the rules of vector algebra (while explaining forces, stresses, moments, accelerations and the like) in terms of vector pointing angles, and the component projections thereof onto other vectors, surfaces, or coordinate frames, the subject of vector rotation analysis will start here from such a traditional perspective.

2. Single Axis–Angle-Based Rotations

We begin the review with a discussion of the dot and cross product operations, which can be used to describe and understand the rotation of a vector about a single axis. Recall, the dot product between two vectors $\vec{a}$ and $\vec{b}$ produces a scalar result, the magnitude of which depends on the product of the magnitude of each vector, $|\vec{a}|$ and $|\vec{b}|$, times the cosine of the angle between them, α. That is,

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos \alpha \tag{1}$$

Since the cosine of a positive angle is the same as the cosine of its negative angle, the value of the cosine is not dependent on the angle's plus (counterclockwise [CCW]) or minus (clockwise [CW]) orientation.

On the other hand, the cross product between vectors $\vec{c}$ and $\vec{d}$ produces a vector $\vec{n}$ that points in the direction perpendicular to both vectors $\vec{c}$ and $\vec{d}$ (according to the right-hand rule), with the magnitude of $\vec{n}$ being equal to the area of the parallelogram formed from the vectors $\vec{c}$ and $\vec{d}$, such that

$$\vec{c} \times \vec{d} = \vec{n} \tag{2}$$

$$|\vec{n}| = |\vec{c}||\vec{d}| \sin \beta \tag{3}$$

where β is the angle between vectors $\vec{c}$ and $\vec{d}$.

Note that the magnitude of the dot and cross product of two vectors can be determined without reference to any coordinate frame, since the magnitude and angle between the two vectors are independent of coordinate frame.

From the above geometrical description of the dot and cross product, it is possible to conceptually formulate how a vector $\vec{a}$, rotated by an angle Ω about an axis-vector $\vec{l}$, ends up pointing in a direction $\vec{a}'$ (Figure 1).

First, let the axis of rotation, $\vec{l}$, be a unit vector. Then, envision decomposing the vector $\vec{a}$ into components that are parallel and perpendicular to $\vec{l}$, namely, $\vec{a} = \vec{a}_{\parallel} + \vec{a}_{\perp}$.

$$\vec{a}_{\parallel} = (\vec{a} \cdot \vec{l})\vec{l} \tag{4}$$

$$\vec{a}_{\perp} = \vec{a} - \vec{a}_{\parallel} = \vec{a} - (\vec{a} \cdot \vec{l})\vec{l} \tag{5}$$

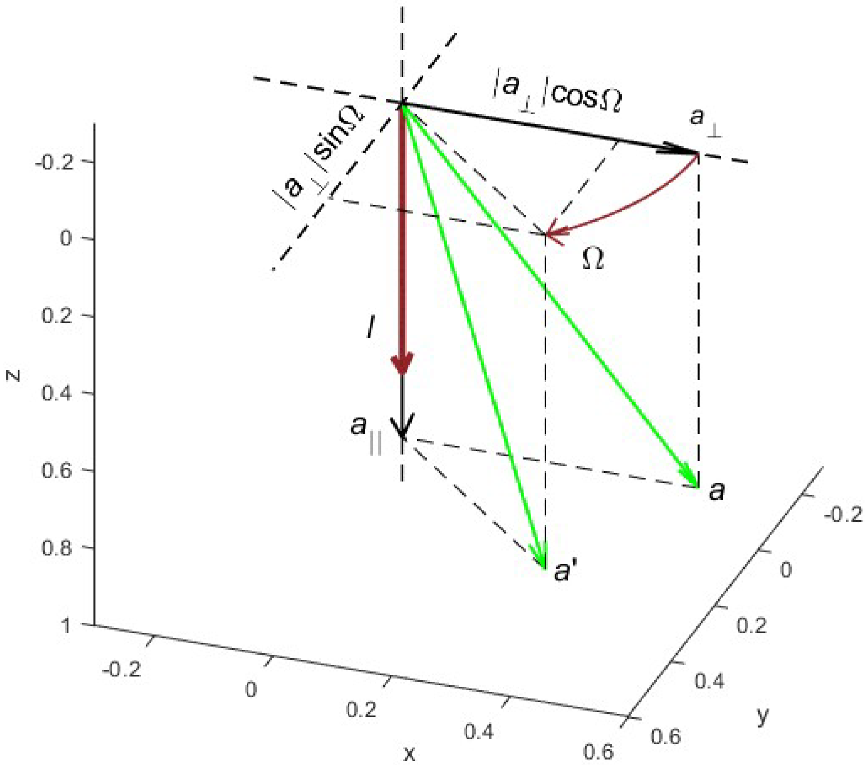


Figure 1. Rodriques representation of rotating vector $\vec{a}$ by Ω about $\vec{l}$ into $\vec{a}'$.

Logically, the parallel component of $\vec{a}$ will not change as a result of rotating $\vec{a}$ about $\vec{l}$ by Ω, but the magnitude of $|\vec{a}_{\perp}|$ will be shortened by $cos\,\Omega$ in the prerotated direction of $\vec{a}_{\perp}$ and the magnitude of $|\vec{a}_{\perp}|$ will be expanded by $sin\,\Omega$ in the direction perpendicular to $\vec{a}$ and $\vec{l}$ according to the right hand rule (i.e., in the direction of $\vec{l} \times \vec{a}$). This description is contained in the Rodrigues formula below, named after the mathematician Olinde Rodrigues:

$$\vec{a}' = \vec{a}_{\parallel} + \vec{a}_{\perp}\, cos\,\Omega + |\vec{a}_{\perp}|\, sin\,\Omega\, (\vec{l} \times \vec{a})/|\vec{l} \times \vec{a}| \tag{6}$$

where $\vec{a}'$ is the attitude/orientation of the original vector $\vec{a}$ after it has been rotated by Ω about $\vec{l}$ (Figure 1). This final attitude is expressed in terms of the original vector components of $\vec{a}$ and $\vec{l}$, which, in that sense, makes it independent of the coordinate system. Since the cross product of $\vec{l}$ with the parallel component of $\vec{a}$ will be zero,

$$\vec{l} \times \vec{a} = \vec{l} \times (\vec{a}_{\parallel} + \vec{a}_{\perp}) = \vec{l} \times \vec{a}_{\perp} \tag{7}$$

and since the angle between the unit vector $\vec{l}$ and $\vec{a}_{\perp}$ is $\pi/2$,

$$\left|\vec{l} \times \vec{a}\right| = \left|\vec{l} \times \vec{a}_{\perp}\right| = \left|\vec{l}\right||\vec{a}_{\perp}| \sin \pi/2 = |\vec{a}_{\perp}| \tag{8}$$

Hence, the Rodriques formula in its most common form, using Equations 4−8, is given by

$$\begin{aligned}
\vec{a}' &= \vec{a}_{\parallel} + \vec{a}_{\perp} \cos \Omega + |\vec{a}_{\perp}| \sin \Omega \, (\vec{l} \times \vec{a})/\left|\vec{l} \times \vec{a}\right| \\
&= (\vec{a} \cdot \vec{l})\vec{l} + \vec{a}_{\perp} \cos \Omega + |\vec{a}_{\perp}| \sin \Omega \, (\vec{l} \times \vec{a})/|\vec{a}_{\perp}| \\
&= (\vec{a} \cdot \vec{l})\vec{l} + (\vec{a} - (\vec{a} \cdot \vec{l})\vec{l}) \cos \Omega + (\vec{l} \times \vec{a}) \sin \Omega \\
&= \cos \Omega \, \vec{a} + (1 - \cos \Omega)(\vec{a} \cdot \vec{l})\vec{l} + \sin \Omega \, (\vec{l} \times \vec{a})
\end{aligned} \tag{9}$$

Note, only two "pieces" of information are needed to determine the rotation of $\vec{a}$ around $\vec{l}$ into $\vec{a}'$: 1) the scalar rotation angle, Ω; and 2) the direction of the 3D unit vector representing the rotation axis, $\vec{l}$. For future reference, these two pieces of information will enable the use of quaternions to describe rotations, as quaternions are inherently 4D variables that contain one scalar and one 3D vector (with three independent and orthogonal components).

3. Multiple Axis-Angle Rotations

The single-axis-based Rodrigues formulation can be viewed as a combination of multiple single-axis rotations.

To facilitate such a discussion, it is useful to define an axis as composed of its vector components along the perpendicular directions of one or more coordinate frames. For this purpose, the symbols *i*, *j*, and *k* will be used in several ways to specify the basis/unit vectors of each coordinate frame. For example, when the symbols *i*, *j*, and *k* are attached to the prefix *ee* (e.g., *eei*, *eej*, *eek*), the first *e* in the sequence conveys it is a unit vector; the second *e* references the earth coordinate frame; and the trailing symbols *i*, *j*, and *k* define specific, orthogonally-independent, unit vector directions. (Later we will define an *eb*, which represents a unit vector in a rotating body.)

Lastly, when we discuss quaternions, the symbols *i, j,* and *k* will stand alone (no prefixes), with each interpreted as a unit vector in one of three orthogonal quaternion "directions." In all cases, whether symbols *i*, *j*, and *k* are preceded by an *ee* (or an *eb*) or whether they stand alone in quaternion algebra, the order of i, j, or k will be such that the relationship between their associated unit vectors follows the right-hand rule and, for simplicity, the unit vector will not appear with the traditional overarrow accent symbol (as shown in Equations 1, 2, etc.).

If the Rodrigues single-axis formula is to hold for rotating a vector around an axis in any 3D direction, then that axis can, in general, be composed of three independent axes of rotations. Moreover, to maintain a continuity of uniqueness, a defined order must be followed in the sequence of rotations about those axes. One choice, commonly used in the literature and by engineering practitioners, is to envision the vector to be rotated as defined by its projections onto a Cartesian coordinate frame that is rotated along with that vector. This defining and rotating coordinate frame (henceforth called the body frame) is originally aligned with the earth-fixed (inertial) Cartesian coordinate frame. The first rotation of the vector, and its defining body frame, is by and angle ψ (called yaw) about the body's vertical axis, denoted *eb3*, which (by definition) begins in alignment with the vertical earth-frame axis, denoted *ee3* (hence initially, $eb3 = ee3$). Such a rotation of the body unit vectors is depicted in Figure 2, and the results of which are tabulated in Table 1, for the projection of the rotated body-frame unit vectors onto the stationary earth-frame unit vectors. Note, $eb3'$(after rotation by ψ) $= eb3$ (before rotation) $= ee3$ because a vector rotated about its own axes does not change.

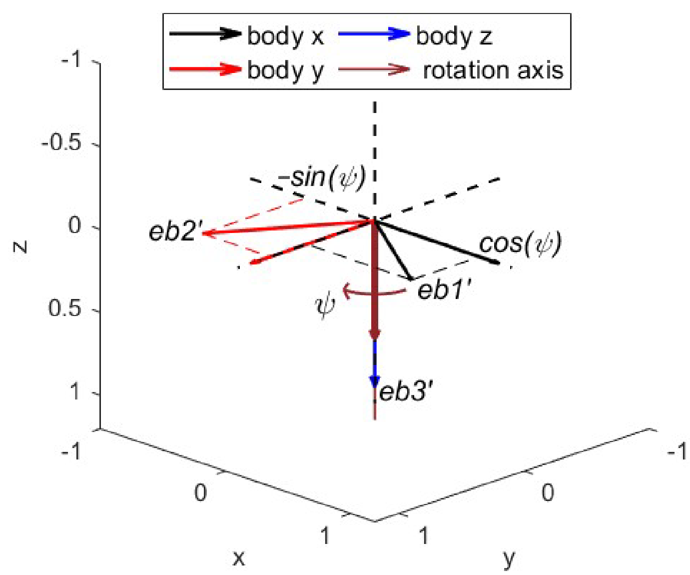


Figure 2. Rotation of body unit vectors by ψ about the body's $eb3' = ee3$.

Table 1. Projections of body axes onto earth axes after a yaw rotation by ψ.

Body unit vectors	$eb1'$			$eb2'$			$eb3' = ee3$		
Earth unit vectors	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$
Projections of body unit vectors onto earth[a]	c_ψ	s_ψ	0	$-s_\psi$	c_ψ	0	0	0	1

[a] $cos(\psi)$ is abbreviated to c_ψ ; $sin(\psi)$ is abbreviated to s_ψ.

Following this first rotation of the body unit vectors by yaw angle ψ, envision and formulate the result of a second rotation of the body's unit vectors by an angle θ (called pitch) about an axis, defined in earth coordinates, but coincident with the newly rotated body's y axis, denoted $eb2'$ in Figure 2 and Table 1. The resulting change in attitude of the body unit vectors after rotation by ψ and θ is depicted in Table 2 and Figure 3 for the projection of the rotated body vectors onto the stationary earth frame. Note $eb2'' = eb2'$ since rotation about its own axis does not change its attitude relative to the earth frame.

Table 2. Projections of body axes onto earth axes after ψ, then θ rotations.

Body unit vectors	$eb1''$			$eb2'' = eb2'$			$eb3''$		
Earth unit vectors	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$
Projections of body unit vectors onto earth after ψ *then* θ rotations	$c_\theta c_\psi$	$c_\theta s_\psi$	$-s_\theta$	$-s_\psi$	c_ψ	0	$s_\theta c_\psi$	$s_\theta s_\psi$	c_θ

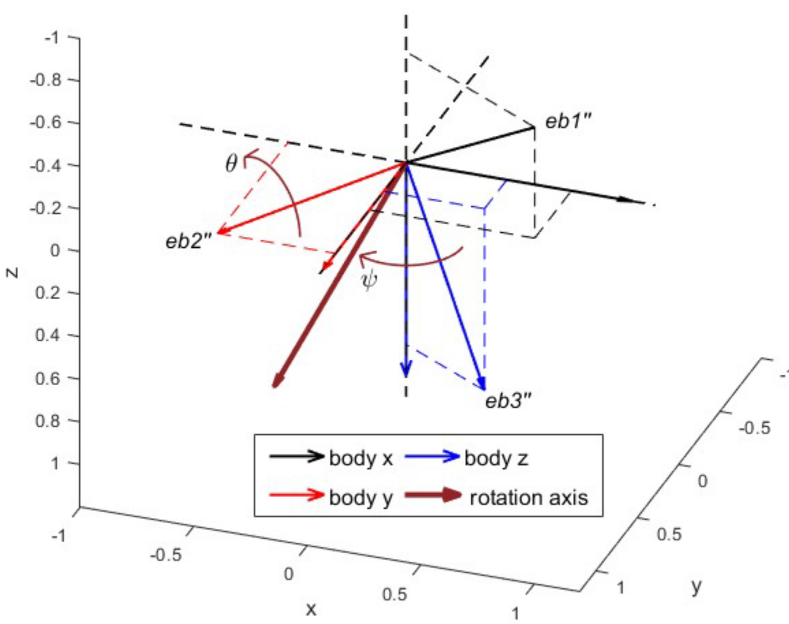


Figure 3. Rotation of body unit vectors by ψ about the body's $eb3' = ee3$ then by θ about $eb2'' = eb2'$.

Finally, we rotate the previous ψ and θ rotations of the body unit vector(s) by an angle ϕ (called roll) about an axis that is coincident with the body x axis, denoted $eb1''$ in Figure 3 and labeled $eb1''' = eb1''$ in Table 3, because the body x axis is unchanged after rotation about itself. Mindful, this third rotation axis is also defined and referenced in the earth coordinate frame.

In Section 5, we will talk about rotation rates around the body axis (as measured, for example, by rotational rate sensors on each body axis, i.e., with gyroscope devices). Those rates will translate into yaw, pitch, and roll angular rates ($\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$), which must satisfy the constraint that the change in yaw, pitch, and roll angles (ψ, θ, and ϕ) will always leave the body axes perpendicular to each other.

Table 3. Projections of body axes onto earth axes after ψ, θ, then ϕ rotations.

Body unit vectors	$eb1''' = eb1''$			$eb2'''$			$eb3'''$		
Earth unit vectors	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$
Projections of body unit vectors onto earth after ψ, θ then ϕ rotations	$c_\theta c_\psi$	$c_\theta s_\psi$	$-s_\theta$	$s_\phi s_\theta c_\psi - s_\psi c_\phi$	$s_\phi s_\theta s_\psi + c_\psi c_\phi$	$c_\theta s_\phi$	$c_\phi s_\theta c_\psi + s_\psi s_\phi$	$s_\psi s_\theta c_\phi - c_\psi s_\phi$	$c_\theta c_\phi$

In summary, rotating the initial body frame unit vectors *eb1*, *eb2*, and *eb3* by the three rotation angles in the specified sequence and about the specified axis results in the final unit vectors of the rotated body frame—denoted, *eb1'''*, *eb2'''*, and *eb3'''*—having their attitudes described by their projected components along the original (and stationary) earth frame unit vectors—denoted *ee1*, *ee2*, and *ee3*—as formulated in Table 3. As projections of unit vectors of the body frame onto unit vectors of the earth frame, the tabular entries can also, by definition of the dot product (Equation 1), be replaced by the cosine of the angle between those respective row-and-column referenced unit vector entries, as depicted in Table 4. (This dot product, or cosine representation, shown in Table 4 is not often used in practice, but we will reference this relationship later when discussing quaternion- vs. vector-algebra dot product relationships.)

Table 4. Projections of body axes onto earth axes, represented in dot product notation.

Body unit vectors	$eb1''' = eb1''$			$eb2'''$			$eb3'''$		
Earth unit vectors	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$	$ee1$	$ee2$	$ee3$
Projection of body unit vectors onto earth after ψ, θ, then ϕ rotations	$eb1''' \cdot ee1$	$eb1''' \cdot ee2$	$eb1''' \cdot ee3$	$eb2''' \cdot ee1$	$eb2''' \cdot ee2$	$eb2''' \cdot ee3$	$eb3''' \cdot ee1$	$eb3''' \cdot ee2$	$eb3''' \cdot ee3$

4. Matrix Representation of Euler Angle Rotations

The sequence of multi-axis rotations described above is attributed (primarily) to Euler (though earlier work by Rodrigues no doubt had some influence). Around that same time period (circa 1850s) another theorist, Arthur Cayley (1821–1895), introduced the mathematical formulism of matrices and matrix multiplication.

There are two ways the tabular data above can be put into matrix forms. One way is to express each entry *eb1'''*, *eb2'''*, and *eb3'''* of Table 3 as functions of the three row entries below it (*ee1*, *ee2*, and *ee3*), such that for example, *eb1'''* = *eb1'''* (*ee1*, *ee2*, *ee3*), where the parentheses following *eb1'''* indicate it is a function of the quantities inside the parentheses. The matrix multiplication that corresponded to this interpretation of Table 3 would look as shown below, and is historically named the Direction Cosine Matrix* (DCM):

$$\begin{pmatrix} eb1''' \\ eb2''' \\ eb3''' \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - s_\psi c_\phi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\theta s_\phi \\ c_\phi s_\theta c_\psi + s_\psi s_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} ee1 \\ ee2 \\ ee3 \end{pmatrix} \qquad (10)$$

The column headings of Table 3 are now row headings of the left column matrix above. Furthermore, going across the rows, the unit vector on the far left in each row (*eb1'''*, etc.) are equated to the multiplication of the row matrix to its right times the vector elements in the column matrix on the far right (*ee1*, etc.). For example, if $\theta = \phi = 0$, then the body's *eb1'''* unit vector would cast a shadow of $cos(\psi)$ on *ee1* and a shadow of $sin(\psi)$ on *ee2*, such that

$$eb1''' = cos(\psi)\ ee1 + sin(\psi)\ ee2 - 0\ ee3 \qquad (11)$$

Or, we could say that *eb1'''* comprises components of *ee1*, *ee2,* and *ee3*.

* The association of the name "cosine" with this matrix stems from the connection of the nine matrix entries with the nine dot product entries in Table 4, and the dot product between unit vectors is just the cosine of the angle between those vectors.

A second way is to group all the *ee1*, *ee2*, and *ee3* components of *eb1‴*, *eb2‴*, *eb3‴* in Table 3 together, so that each row of this alternative matrix representation of Table 3 comprises, for example, *ee1* = *ee1*(*eb1‴*, *eb2‴*, *eb3‴*). Such a table would look as shown below, and is named the Inverse Direction Cosine Matrix (DCM^{-1}):

$$\begin{pmatrix} ee1 \\ ee2 \\ ee3 \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} eb1''' \\ eb2''' \\ eb3''' \end{pmatrix} \quad (12)$$

In this case, the matrix multiplication of each row in the DCM^{-1}, times the column matrix on the far right, is describing how the unit vector elements in the rows of the far-left column project onto the unit vectors of the body axes. For example, if $\theta = \phi = 0$, then the stationary *ee1* unit vector would cast a shadow of *cos(ψ)* on *eb1‴* and a shadow of *sin(ψ)* on −*eb2‴*, such that

$$ee1 = cos(\psi)\ eb1''' - sin(\psi) \text{ on } eb2''' + 0\ eb3''' \quad (13)$$

Or, we could say *ee1* comprises components of *eb1‴*, *eb2‴*, and *eb3‴*. These alternative interpretations of Equations 11 and 13 would be more natural if we were describing the projection of a general earth-fixed vector (rather than earth unit vectors) onto the body frame in the case of Equation 11, or a body-fixed vector onto the earth frame in the case of Equation 13, as will be discussed next.

(Historically, this second conveyance of Table 3 in the form of DCM^{-1} is called the "passive" rotation interpretation, whereas organizing Table 3 in the form of the DCM is called the "active" rotation interpretation.)

Unless otherwise specified, reference to *eb1*, *eb2*, and *eb3* will be synonymous with reference to the final of the three Euler rotations in Table 3 as *eb1‴*, *eb2‴*, and *eb3‴*.

Either matrix interpretation of Table 3 can be made because the dot product of the angle between the unit vectors is the same whether we are asking for the projection of *ebj* onto *eek (j,k = 1,2,3)*, or the projection of *eek* onto *ebj*. This was the take-away from Table 4.

Although the above developments showcase how to relate the basis vectors in the body to the earth basis-vectors through use of the DCM (Equation 10) and vice versa if using the DCM^{-1} (Equation 12), transforming a particular vector with given vector component magnitudes in the earth frame to the equivalent vector components in the body frame, and vice versa, interprets the use of the DCM and DCM^{-1} in a slightly different manner.

In particular, suppose we have a known earth-fixed vector with vector coefficients in the earth frame, *a1 ee1* + *a2 ee2* + *a3 ee3*. Initially, the body frame has these same coefficients along the body axes before rotation (because the body frame starts in alignment with the earth frame), but suppose we want to know what the projection of this earth-fixed vector is on the body frame after the body has been rotated by ψ, θ, and ϕ; i.e., how do we find

$$\vec{b} = b1(a1, a2, a3)\ eb1 + b2(a1, a2, a3)\ eb2 + b3(a1, a2, a3)\ eb3 \tag{14}$$

The body coefficients, *b1*, *b2*, and *b3* can be found as functions of the earth-fixed vector coefficients *a1*, *a2*, and *a3* by matrix-multiplying the DCM times the column matrix of the *a*-vector; i.e.,

$$[b1, b2, b3]^T = \text{DCM}\ [a1, a2, a3]^T$$

$$= \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - s_\psi c_\phi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\theta s_\phi \\ c_\phi s_\theta c_\psi + s_\psi s_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} a1 \\ a2 \\ a3 \end{pmatrix} \tag{15}$$

where the superscript "T" conveys that the row-like, matrix appearance is really representing a column matrix; i.e., $[b1, b2, \text{and } b3]^T$ and $[a1, a2, a3]^T$ are both 3-row by 1-column matrices. For clarity, the long form version would be

$$b1 = a1\{cos(\theta)cos(\psi)\} + a2\{cos(\theta)sin(\psi)\} - a3\{sin(\theta)\} \tag{16}$$

$$b2 = a1\{sin(\phi)sin(\theta)cos(\psi) - sin(\psi)cos(\phi)\}$$
$$+ a2\{cos(\psi)cos(\phi) + sin(\phi)sin(\theta)sin(\psi)\} + a3\{sin(\phi)cos(\theta)\} \tag{17}$$

$$b3 = a1\{sin(\theta)cos(\psi)cos(\phi) + sin(\phi)sin(\psi)\}$$
$$+ a2\{sin(\theta)sin(\psi)cos(\phi) - sin(\phi)cos(\psi)\} + a3\{cos(\theta)cos(\phi)\} \tag{18}$$

If we combine these body vector coefficients with their basis vectors, expressed as functions of the earth basis vectors (Equation 10), we get

$$\vec{b} = b1\ eb1 + b2\ eb2 + b3\ eb3$$
$$= b1\ (cos(\theta)cos(\psi)\ ee1 + cos(\theta)sin(\psi)\ ee2 - sin(\theta)\ ee3\ )$$
$$+ b2\ (\{-sin(\psi)cos(\phi) + sin(\phi)sin(\theta)cos(\psi)\}\ ee1$$
$$+ \{cos(\psi)cos(\phi) + sin(\phi)sin(\theta)sin(\psi)\}\ ee2 + \{sin(\phi)cos(\theta)\}\ ee3\ )$$
$$+ b3\ (\{sin(\theta)cos(\psi)cos(\phi) + sin(\phi)sin(\psi)\}\ ee1$$
$$+ \{sin(\theta)sin(\psi)cos(\phi) - sin(\phi)cos(\psi)\}\ ee2 + \{cos(\theta)cos(\phi)\}\ ee3\ ) \tag{19}$$

By substituting the body vector coefficients of Equations 16–18 into Equation 19, performing the extensive trigonometry (trig) function multiplications, collecting terms, and use trig identities, we will indeed find

$$\vec{b} = \vec{a} \tag{20}$$

For example, for the simple case where $\theta = \phi = 0$, Equations 16–18 would yield

$$b1 = a1\cos(\psi) + a2\sin(\psi) \tag{21}$$

$$b2 = -a1\sin(\psi) + a2\cos(\psi) \tag{22}$$

$$b3 = a3 \tag{23}$$

Substituting Equations 21–23 into Equation 19 would yield

$$\begin{aligned}
\vec{b} &= b1\, eb1 + b2\, eb2 + b3\, eb3 \\
&= (a1\cos(\psi) + a2\sin(\psi))(\cos(\psi)\, ee1 + \sin(\psi)\, ee2) \\
&\quad + (-a1\sin(\psi) + a2\cos(\psi))(-\sin(\psi)\, ee1 + \cos(\psi)\, ee2) + a3\, ee3 \\
&= a1\cos^2(\psi)\, ee1 + \cancel{a2\sin(\psi)\cos(\psi)\, ee1} + \cancel{a1\cos(\psi)\sin(\psi)\, ee2} + a2\sin^2(\psi)\, ee2 \\
&\quad + a1\sin^2(\psi)\, ee1 - \cancel{a2\sin(\psi)\cos(\psi)\, ee1} - \cancel{a1\sin(\psi)\cos(\psi)\, ee2} + a2\cos^2(\psi)\, ee2 \\
&\quad + a3\, ee3 = a1\, ee1 + a2\, ee2 + a3\, ee3 = \vec{a}
\end{aligned} \tag{24}$$

as claimed.

On the other hand, if we know a stationary vector $\vec{b}$ in the moving body frame with component coefficients of $b1$, $b2$, and $b3$, then the projection of $\vec{b}$ onto the earth frame after ψ, θ, and ϕ rotations is given by

$$\vec{b} = a1(b1, b2, b3)\, ee1 + a2(b1, b2, b3)\, ee2 + a3(b1, b2, b3)\, ee3 \tag{25}$$

with the coefficients $a1$, $a2$, and $a3$ obtained from

$$\begin{aligned}
[a1, a2, a3]^{\mathrm{T}} &= \mathrm{DCM}^{-1}\, [b1, b2, b3]^{\mathrm{T}} \\
&= \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix}
\end{aligned} \tag{26}$$

In general, because both the body and earth coordinate frames are composed of perpendicular/orthogonal unit vectors, the magnitude of $|\vec{b}| = |\vec{a}|$ in either matrix transformation (DCM or DCM^{-1}) described above (as proven by taking the square root of the dot product of either vector with itself).

Thus, the DCM of Equation 15 can be used to describe how the vector coefficients, *a1, a2, a3*, of a known earth-fixed (stationary) vector, $\vec{a} = a1\, ee1 + a2\, ee2 + a3\, ee3$, transform into the vector coefficients *b1, b2, b3* of that same fixed earth-vector as seen in the body coordinate frame, i.e., $\vec{a} = b1\, eb1 + b2\, eb2 + b3\, eb3 = \vec{b}$. Likewise, the DCM^{-1} of Equation 26 can be used to determine how the known vector coefficients, *b1, b2, b3,* of a stationary vector in the body frame $\vec{b} = b1\, eb1$

+ *b2 eb2* + *b3 eb3*, project onto, and thus define its vector coefficients, *a1, a2, a3*, as seen in the earth frame, i.e., $\vec{b} = a1\ ee1 + a2\ ee2 + a3\ ee3 = \vec{a}$.

The stepwise process of building up Table 3 from the sequence of ψ, θ, and then ϕ rotations could be created by matrix multiplication of three individual 3 × 3 matrices, each representing one of the three rotations. That is,

$$\mathrm{DCM}(\phi, \theta, \psi) = \mathrm{DCM}(\phi,0,0)\mathrm{DCM}(0, \theta,0)\mathrm{DCM}(0,0, \psi) \tag{27}$$

In the field of aerospace, this particular sequence has been associated with the angles of yaw (for ψ), pitch (for θ), and roll (for ϕ), but there are other rotation sequences in the literature for which the DCM would contain different arrangements of cosines and sines.

Suppose the order of rotations and angle directions were reversed; namely, the first rotation of the body was about *ee1* by $-\phi$, then the body was rotated by $-\theta$ about the *eb2′* axis (that resulted from the previous $-\phi$ rotation), and finally, the body was rotated by $-\psi$ about *eb3″* (resulting from the previous first $-\phi$ then $-\theta$). Following the earlier step-by-step projection analysis, or simply using the prescription given in Equation 27, the final result, for this reverse rotation order

$$\mathrm{DCM}(-\psi, -\theta, -\phi) = \mathrm{DCM}(-\psi,0,0)\mathrm{DCM}(0, -\theta,0)\mathrm{DCM}(0,0, -\phi) \tag{28}$$

can be shown equivalent to

$$\mathrm{DCM}(-\psi, -\theta, -\phi) = \mathrm{DCM}^{-1}(\phi, \theta, \psi) \tag{29}$$

In essence, if we first rotate the body by the angles and sequence corresponding to DCM(ϕ, θ, ψ), then reverse the order and angles of that sequence of operations according to DCM($-\psi$, $-\theta$, $-\phi$), we would be returning the body to its original starting orientation. This is borne out by the matrix multiplication of the two 3 × 3 matrices below, which result in the identity matrix, I (which is a 3 × 3 matrix with ones on the diagonal and zeros on the off-diagonal):

$$\mathrm{DCM}(-\psi, -\theta, -\phi)\ \mathrm{DCM}(\phi, \theta, \psi) \equiv \mathrm{DCM}^{-1}(\phi, \theta, \psi)\ \mathrm{DCM}(\phi, \theta, \psi) = \mathrm{I} \tag{30}$$

Importantly, although the body unit vectors are perpendicular to one another (as are the earth unit vectors), the Euler-angle-based body rotation axes are no longer perpendicular to each other after any nonzero rotation of θ (reference the axes definitions in Section 3). That is, even though the second of the three Euler rotation axes *eb2′* (about which the body is rotated CCW by θ) is perpendicular to the first rotation axis *eb3* = *ee3* (about which the body is rotated CCW by ψ) and is also perpendicular to the third rotation axis *eb1″* (about which the body is rotated CCW by ϕ), this third rotation axis (*eb1″*) is only perpendicular to the first axis *eb3* = *ee3*, if $\theta = 0$. In fact, when the angle θ is 90°, $\pi/2$ radians (rad.) (henceforth, for brevity, if an expression involves π we will suppress/ignore attaching the rad. unit designation after it), it rotates the third axis, *eb1″*, to earth-vertical, i.e., aligned with the first axis, *eb3* = *ee3*. In this special case, when the first and third Euler rotation axis are coincident, specifying a ϕ angle about the then-earth-vertical *eb1″* axis will produce a change in the body's attitude that could also have been produced with a different initial angle ψ about the same vertical axis. In other words, with only the vertical and horizontal plane axes to describe a body's motion, the body's yaw (ψ) and roll (ϕ) angles do not create unique change in the body's attitude. The derivation leading to Table 3, and hence the DCM and DCM^{-1} matrix representation of Table 3, relate the projection of earth to body, and body to earth vectors, for a specific set of Euler angles ψ, θ, and ϕ. If these angles continuously change in time, these changes must be constrained such that the prediction of the DCM and DCM^{-1} will be consistent with the body and earth axes remaining perpendicular to each other. The constraints will appear as conditions on the angular rates, as will be shown next.

5. Euler Angle Rotation Rates and the Appearance of a Rate Singularity Condition

As discussed, the DCM transformation describes how the body axes unit vectors (left column matrix of Equation 10) project onto the earth axes, or how coefficients of a stationary earth vector (right column matrix of Equation 15) project onto the moving body frame coordinate system (left column matrix). Conversely, DCM^{-1} transformations describe how the earth axes unit vectors project onto the moving body axes (left column matrix of Equation 12), or how the coefficients of a vector moving with the body frame (right column matrix of Equation 26) project onto vector coefficients in the stationary earth frame.

If we want to track how projections change with time, then we need to know $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$. For most rotating bodies, the rate of change of these angles is due to an applied torque. This torque, whatever the source and wherever it is applied, causes angular acceleration in one or more body planes that is often measured by an angular rate gyroscope (gyro). We will denote the angular rate change measured by a body gyro about the body's *eb3* axis as *R*, let *Q* represent the angular rate change about the body's *eb2* axis, and *P* signify the angular rate change about the body's *eb1* axis.

As orthogonal vectors of unit length, the body's unit axes can only change with time in their perpendicular directions. For example, $e\dot{b}1$ can only change in the direction of *eb2* and *eb3*.

The rotation rate of $\dot{eb}1$ in the direction of *eb2* would be measured by the rate gyro *R* (Figure 4).

Likewise, the rate of $\dot{eb}1$ in the direction of *–eb3* is measured by the rate gyro *Q*.

In a similar fashion, the time rate of change, $\dot{eb}2$, can be captured from gyro data around the body's *eb1* and *eb3* axes, while $\dot{eb}3$ can be tracked from gyro rate data taken around the *eb1* and *eb2* axes.

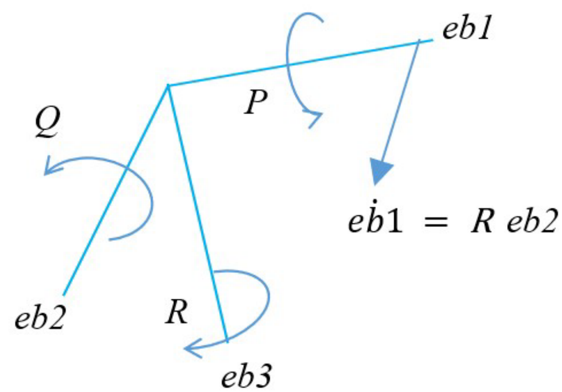


Figure 4. Unit body vectors can only change in directions normal to themselves; e.g., the rotation rate of $\dot{eb}1$ in the direction of *eb2* would be measured by the rate gyro *R*.

We can summarize such formulations as follows:

$$\dot{eb}1 = R\ eb2 - Q\ eb3 \tag{31}$$

$$\dot{eb}2 = -R\ eb1 + P\ eb3 \tag{32}$$

$$\dot{eb}3 = -P\ eb2 + Q\ eb1 \tag{33}$$

These three equations add constraints to the Euler angles such that the body axes always remain perpendicular to each other.

As previously stated, one application of the DCM was to show how the unit body axes (left column matrix) project onto the Earth's axes (Equation 10) and expanded below:

$$eb1 = \{cos(\theta)cos(\psi)\}ee1 + \{cos(\theta)sin(\psi)\}ee2 - \{sin(\theta)\}ee3 \tag{34}$$

$$\begin{aligned} eb2 = \{&sin(\phi)sin(\theta)cos(\psi) - sin(\psi)cos(\phi)\}ee1 \\ &+ \{cos(\psi)cos(\phi) + sin(\phi)sin(\theta)sin(\psi)\}ee2 + \{sin(\phi)cos(\theta)\}ee3 \end{aligned} \tag{35}$$

$$\begin{aligned} eb3 = \{&sin(\theta)cos(\psi)cos(\phi) + sin(\phi)sin(\psi)\}\ ee1 \\ &+ \{sin(\theta)sin(\psi)cos(\phi) - sin(\phi)cos(\psi)\}ee2 + \{cos(\theta)cos(\phi)\}ee3 \end{aligned} \tag{36}$$

Differentiating *eb1* above with respect to time, we obtain

$$e\dot{b}1 = \{ -cos(\psi)sin(\theta)\,[\dot{\theta}] - \cos(\theta)\sin(\psi)\,[\dot{\psi}] \}\ ee1$$
$$+ \{ -sin(\psi)sin(\theta)\,[\dot{\theta}] + cos(\theta)cos(\psi)\,[\dot{\psi}] \}\ ee2$$
$$+ \{ -cos(\theta)\,[\dot{\theta}] \}\ ee3 \qquad (37)$$

which shows how the time rate of change of *eb1* projects onto the earth axes as a function of $\dot{\psi}$ and $\dot{\theta}$.

Using Equation 31 for $e\dot{b}1$ and Equation 35 for *eb2*, we can show

$$R = e\dot{b}1 \cdot eb2 = \cos(\theta)\cos(\phi)\,[\dot{\psi}] - sin(\phi)\,[\dot{\theta}] \qquad (38)$$

Performing comparable substitutions, we can show

$$Q = e\dot{b}1 \cdot (-\,eb3) = cos(\phi)\,[\dot{\theta}] + \cos(\theta)\sin(\phi)\,[\dot{\psi}] \qquad (39)$$

$$P = e\dot{b}2 \cdot eb3 = [\dot{\phi}] - sin(\theta)\,[\dot{\psi}] \qquad (40)$$

In matrix form, these three relationships can be represented as

$$\begin{pmatrix} R \\ Q \\ P \end{pmatrix} = \begin{pmatrix} c_\phi c_\theta & -s_\phi & 0 \\ s_\phi c_\theta & c_\phi & 0 \\ -s_\theta & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} \qquad (41)$$

There are many other rearrangements of this, some of which are

$$\begin{pmatrix} R \\ Q \\ P \end{pmatrix} = \begin{pmatrix} 0 & -s_\phi & c_\phi c_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 1 & 0 & -s_\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \qquad (42)$$

or

$$\begin{pmatrix} P \\ Q \\ R \end{pmatrix} = \begin{pmatrix} -s_\theta & 0 & 1 \\ s_\phi c_\theta & c_\phi & 0 \\ c_\phi c_\theta & -s_\phi & 0 \end{pmatrix} \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} \qquad (43)$$

Equations 38–43 can be found in the literature (e.g., Etkin 1972). The singularity problem comes to the forefront when we solve these three equations for $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$ (which we need to know to update, in time, the Euler angles within the DCM and or DCM^{-1}) as functions of *R*, *P*, and *Q*. Doing so reveals

$$\dot{\phi} = R\ sin(\theta)cos(\phi)/cos(\theta) + Q\ sin(\theta)sin(\phi)/cos(\theta) + P \qquad (44)$$

$$\dot{\theta} = -R\ sin(\phi) + Q\ cos(p) \qquad (45)$$

$$\dot{\psi} = R\ cos(\phi)/cos(\theta) + Q\ sin(\phi)/cos(\theta) \qquad (46)$$

Or, in matrix form:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s_\theta s_\phi / c_\theta & s_\theta c_\phi / c_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix} \tag{47}$$

Equation 47 is found throughout the literature (e.g., Etkin 1972; Ilg 2008; Fresconi et al. 2014; Dykes et al. 2013) and shows that solving for $\dot{\phi}$ and $\dot{\psi}$ is problematic when θ approaches $\pi/2$, whence $\dot{\phi}$ and $\dot{\psi}$ will approach infinity, thus defining a singularity condition (even though the difference, $\dot{\phi} - \dot{\psi}$, will approach the finite rate of *P*). In addition, notation-wise, the angular rates we are symbolizing as *P*, *Q*, and *R* can also be expressed as *wb1*, *wb2*, and *wb3*, i.e., representing angular rates about the body's *eb1*, *eb2,* and *eb3* axes, respectively.

Further investigation in Section 8 will address whether/when a singularity actually occurs. But first we will look into the claim that quaternion algebra can be used to avoid the Euler angle rate singularity problem (e.g., Well and Weaver 1984; ANSI/AIAA 1992; Diebel 2006; Kim 2013, p. 37; Ben-Ari 2014; Ueno et al. 2014, p. 2; Günasti 2016; Chudá 2019; Challis 2020, p. 1).

6. The Basics of Quaternion Algebra

As mentioned in the introduction, algebras define mathematical operations, "rules," between symbolic representations of variables in one or more dimensions. The traditional field of vector algebra in 3D space is based upon the concepts and definitions of trigonometry, parallel and perpendicular vectors, dot and cross product operations, and the rules of matrices, and the like. Quaternion algebra, developed by Hamilton, uses some of the same formalisms as traditional vector algebra, but adds some new rules and definitions as well.

A quaternion is a 4D object, having a pure scalar dimension (which can be thought of as a coordinate-less point, but with no specified basis symbol), denoted here as simply

$$q_o \tag{48}$$

along with a 3D vector, denoted here as $\vec{q}$, with basis vectors *i*, *j*, and *k* and basis vector coefficients/magnitudes q_i, q_j, q_k such that

$$\vec{q} = q_i\, i + q_j\, j + q_k\, k \tag{49}$$

The complete quaternion with its scalar and vector components is

$$q = q_o + \vec{q} = q_o + q_i\, i + q_j\, j + q_k\, k \tag{50}$$

As previously mentioned, the *i*, *j*, and *k* basis vectors of a quaternion have some aspects in common with the basis vectors of a Cartesian coordinate system, namely the following: they are referenced to an origin, are perpendicular to each other, have unit magnitude, and relate to each other in a cross-product-like sense:

$$i \times j \text{ (simplified to } ij) = k \tag{51}$$

$$ki = j \tag{52}$$

$$jk = i \tag{53}$$

$$ji = -k \tag{54}$$

as well as

$$i \times i = j \times j = k \times k = 0 \tag{55}$$

and

$$i \cdot j = 0 \quad (\text{ for } i \neq j) \tag{56}$$

However, unlike Cartesian basis vectors, quaternion basis vectors behave like imaginary numbers in that they square to –1:

$$i \cdot i = j \cdot j = k \cdot k = -1 \tag{57}$$

Because of this squaring to -1 property, the dot product rule for traditional vector algebra, which is computed from the magnitude of each vector times the cosine of the angle, α, between vectors (Equation 1), is not the same as the quaternion dot product, which in effect is computed from the cosine of π + α such that

$$\begin{aligned}\vec{q} \cdot \vec{q}' &= (q_i\, i + q_j\, j + q_k\, k) \cdot (q_i'\, i + q_j'\, j + q_k'\, k) \\ &= (q_i\, i \cdot q_i'\, i) + (q_j\, j \cdot q_j' j) + (q_k\, k \cdot q_k'\, k) \\ &= (q_i)(q_i')(i \cdot i) + (q_j)(q_j')(j \cdot j) + (q_k)(q_k')(k \cdot k) \\ &= -(q_i)(q_i') - (q_j)(q_j') - (q_k)(q_k')\end{aligned} \tag{58}$$

In traditional vector algebra, the mathematical definition for multiplication of scalars versus vectors are separate, there is one scalar product (a simple multiplication of two scalar terms), and there are two vector products: the dot and cross product. The dot product involves three multiplications of vector coefficients and base vectors, and the cross product is composed of six coefficient and base vector multiplications. Since a quaternion, Equation 50, is a four-component complex number, quaternion multiplication will involve 16 complex number multiplications, which results in

$$\begin{aligned}q\, q' &= (q_o + q_i\, i + q_j\, j + q_k\, k)(q_o' + q_i'\, i + q_j'\, j + q_k'\, k) \\ &= (q_o q_o' - q_i q_i' - q_j q_j' - q_k q_k') + q_o(q_i'\, i + q_j'\, j + q_k'\, k) + q_o'(q_i\, i + q_j\, j + q_k\, k) \\ &\quad + (q_i q_j' - q_j q_i')\, k + (q_k q_i' - q_i q_k')\, j + (q_j q_k' - q_k q_j')\, i\end{aligned} \tag{59}$$

(In some texts, quaternion multiplication is symbolized by the $\otimes$ symbol inserted between the quaternions being multiplied, but it is often common to represent such multiplication with no symbol between the quaternions, which is the form that will be used here.)

As seen, some of these multiplication results are scalars, and some are quaternion vectors, so that the result remains a quaternion.

Upon inspection, and a judicious regrouping of terms, the quaternion multiplication above can be put into the form:

$$q\,q' = \underbrace{q_o q_o' + \vec{q}\cdot\vec{q}\,'}_{\text{scalar}} + \underbrace{q_o(q_i'\,i + q_j'\,j + q_k'\,k) + q_o'(q_i\,i + q_j\,j + q_k\,k) + \vec{q}\times\vec{q}\,'}_{\text{vector}} \tag{60}$$

This alternative, shorter-looking form of quaternion multiplication is often referred to as the quaternion product (though it produces the same answer as the longer form, Equation 59). To avoid possible confusion, most reference material (e.g., Kuipers 2000; Jia 2022, p. 2) has a minus sign ahead of the $\vec{q}\cdot\vec{q}\,'$ term in the definition of the quaternion product. The explanation for such a minus sign is that in those literature formulations, the dot product is treated as though the vectors being dotted are Euclidian (where basis vectors square to +1); in that context, it is necessary to insert a minus sign to compensate for the fact that quaternion basis vectors square to -1. We choose the form above, Equation 60, since quaternion dot-product multiplication is complex number multiplication, and as such, the minus sign would be the defined outcome (Equation 57).

If the scalar terms are zero, the quaternion product of quaternion vectors alone would be

$$\vec{q}\,\vec{q}\,' = \vec{q}\cdot\vec{q}\,' + \vec{q}\times\vec{q}\,' \tag{61}$$

In addition, since the cross product of either Cartesian or quaternion vectors is anti-commutative,

$$\vec{q}\times\vec{q}\,' = -\vec{q}\,'\times\vec{q} \tag{62}$$

it means that

$$\vec{q}\,\vec{q}\,' \neq \vec{q}\,'\vec{q} \tag{63}$$

and therefore, the full quaternion product is also not commutative:

$$q\,q' \neq q'q \tag{64}$$

Quaternions have a definable magnitude, contained in both its scalar (point) and vector dimensioned terms. Since a quaternion, q, is a complex number, its magnitude is computed by multiplying itself by its complex conjugate, q^*. Thus, the magnitude of a quaternion, as determined by either multiplication methods above (Equation 59 or 60) is given by

$$|q| = \sqrt{q\, q^*} = \sqrt{q_o^2 + \sum_{n=i,j,k} q_n^2} \tag{65}$$

As a philosophical aside, it might be noted (and perhaps has, somewhere in the literature) that the magnitude of a 4D quaternion is composed of a scalar and three vector magnitudes, not unlike an analogy with the description of total energy when it is composed of a scalar potential energy and the three vector-component magnitudes of kinetic energy.

Returning to quaternion algebra, and lastly in this brief review, quaternions have an inverse, denoted q^{-1}, defined as

$$q^{-1} = q^*/|q|^2 \tag{66}$$

such that

$$q\, q^{-1} = q^{-1} q = 1 \tag{67}$$

With the above quaternion algebra concepts in mind, it is now possible to begin discussion of how quaternions can be used to describe 3D Euclidean-space vector rotations.

7. Describing 3D Rotations Using Quaternions

7.1 Single Axis (Rodrigues Case Study)

As the Rodrigues formula showed (Equation 9), all that is needed to describe a vector rotation is the 3D axis of rotation, $\vec{l}$, and the scalar angle of rotation Ω about that axis, i.e., four independent quantities. Since a quaternion contains both scalar and 3D vector components (albeit a vector with imaginary unit bases), it is reasonable to expect that there would be a way to use the 4D quaternion elements to duplicate the scalar and vector formulation of Rodrigues (rewritten below):

$$\vec{a}' = \cos\Omega\, \vec{a} + (1 - \cos\Omega)(\vec{a} \cdot \vec{l})\vec{l} + \sin\Omega\, (\vec{l} \times \vec{a}) \tag{9}$$

If we are going to look for a quaternion equivalent expression of the Rodrigues vector rotation formula, we will need to adopt a mapping of Cartesian bases vectors to quaternion bases vectors and vice versa. For that purpose, let *ee1* ↦ *i*, *ee2* ↦ *j*, and *ee3* ↦ *k*. Hence, a Cartesian vector $\vec{a}$ would map into a quaternion vector as follows:

$$\vec{a} = a1\, ee1 + a2\, ee2 + a3\, ee3 \;\mapsto\; \vec{a} = a_i\, i + a_j\, j + a_k\, k \tag{68}$$

where $a1 = a_i$, etc. In 3D Euclidean vector algebra, vector multiplication takes place in the form of either a dot or a cross product. As mentioned above, in quaternion algebra we have a quaternion product (Equation 60) that combines both a dot and cross product operation, which, for $q = q_o + \vec{q}$ and using $\vec{a}$ as shown in Equation 68 (with no scalar term), we arrive at

$$q\,\vec{a} \;=\; q_o\vec{a} + \vec{q}\cdot\vec{a} + \vec{q}\times\vec{a} \tag{69}$$

Equation 69 shows that this quaternion multiplication produces two vector terms: one that is in the direction of the original $\vec{a}$ (the first of three terms on the right, above) and one that is perpendicular to both $\vec{q}$ and $\vec{a}$ (the third term on the right), as well as one scalar term (the second term on the right). But since $q\,\vec{a}$ contains a scalar term, it cannot represent a pure vector, unless $\vec{q}\cdot\vec{a} = 0$. Then, $q\,\vec{a}$ could describe the rotation of $\vec{a}$ in a plane perpendicular to $\vec{q}$, provided the scalar part of q was equated to the cosine of such a rotation angle and the magnitude of the vector part of q was equated to the since of that rotation angle. A more certain outcome from the quaternion product can be obtained as follows.

The Rodrigues formula contains three vector terms: one that remained in the direction of the vector being rotated, $\vec{a}$, and depended on the cosine of the rotation angle, Ω (the first term in Equation 9); another vector term that is perpendicular to both the rotation axis and the vector being rotated, $\vec{l}\times\vec{a}$, and depended on the sine of the rotation angle (the third term in Equation 9); and the final vector term that is in the direction of the axis of rotation, $\vec{l}$ (the second term in Equation 9).

Comparing the Rodrigues formula with the quaternion multiplication formula for $q\,\vec{a}$, we can identify some similarities. In particular, if q_o is associated with the cosine of the rotation angle, and $\vec{q}$ is associated with the direction of the rotation axis, $\vec{l}$, but scaled down by the sine of the rotation angle, then the first and third terms of Equation 69 would appear similar to the first and third term of the Rodrigues formula. However, the second vector term of the Rodrigues formula would not match the scalar (second) term of the quaternion multiplication result.

The second term of Rodrigues equation is in the direction of the axis of rotation, $\vec{l}$, and represents the component of $\vec{a}$ that is parallel to the axis of rotation (and is thus unchanged by rotation about the axis). Similarly, the scalar term of the quaternion multiplication would represent the component of $\vec{a}$ that was parallel to $\vec{q}$ (which seems to bear at least a scaled association with the unit-length rotational vector $\vec{l}$), but without including the direction of $\vec{q}$. Thus, the match is close, but not exact. Could an exact match be created?

It can be shown that if we take the quaternion product above, and post-multiply it with the inverse quaternion q^{-1}, we will achieve a result that gets closer to looking like the Rodrigues formula in that it contains only vector terms that are either in the direction of $\vec{a}$, $\vec{q}$, or $\vec{q} \times \vec{a}$, with no scalars present. In particular,

$$\vec{a}' = (q\,\vec{a})q^{-1} = (q_o^2)\,\vec{a} + 2\,q_o\,(\vec{q} \times \vec{a}) - 2\,(\vec{q} \cdot \vec{a})\,\vec{q} + (\vec{q} \cdot \vec{q})\,\vec{a} \quad (70)$$

To derive the above expression from the quaternion product rule (Equation 60), we made use of the trig identity for the quaternion vector triple product, namely

$$(\vec{q} \times \vec{a}) \times \vec{q} = (\vec{q} \cdot \vec{a})\,\vec{q} - (\vec{q} \cdot \vec{q})\,\vec{a} \quad (71)$$

This is different from the vector triple product for Cartesian vectors because we continue to view the dot product of quaternion vectors as multiplying their basis vectors as well as their basis vector coefficients. Hence, the dot products on the right in the above quaternion equation are the negatives of this same equation if it was written for 3D Cartesian vectors. Again, these types of quaternion versus Cartesian dot product rule differences are due to the squaring of quaternion bases to –1 and must be kept in mind when relating/mapping formulas in vector algebra to their quaternion algebra counterparts—and vice versa.

If we revise slightly our previous attempt to associate q_o and $\vec{q}$ with the Rodrigues variables in Equation 9, and now equate/map

$$q_o \mapsto cos(\Omega\,/2) \quad (72)$$

$$\vec{q} \mapsto sin(\Omega\,/2)\vec{l} \quad (73)$$

Then, substituting these expressions, along with

$$(\vec{q} \cdot \vec{q})\,\vec{a} = sin^2(\Omega\,/2)\,(\vec{l} \cdot \vec{l})\,\vec{a} = sin^2(\Omega\,/2)\,(-1)\,\vec{a} \quad (74)$$

into Equation 70, yields the following:

$$\begin{aligned}\vec{a}' &= (cos^2(\Omega\,/2) - sin^2(\Omega\,/2))\,\vec{a} + 2cos(\Omega\,/2)sin(\Omega\,/2)\,(\vec{l} \times \vec{a}) - 2sin^2(\Omega\,/2)\,(\vec{a} \cdot \vec{l})\vec{l} \\ &= cos(\Omega)\,\vec{a} + sin(\Omega)\,(\vec{l} \times \vec{a}) + (cos(\Omega) - 1)\,(\vec{a} \cdot \vec{l})\vec{l} \quad (75)\end{aligned}$$

where we have used standard half-angle trig identities, including

$$\begin{aligned}-2sin^2(\Omega\,/2) &= -sin^2(\Omega\,/2) - sin^2(\Omega\,/2) \\ &= cos^2(\Omega\,/2) - 1 - sin^2(\Omega\,/2) = (cos(\Omega) - 1) \quad (76)\end{aligned}$$

Comparing the above equation for $\vec{a}' = (q\,\vec{a})q^{-1}$ (Equation 75, using Equations 70–74 and 76) with the Rodrigues formula, we see that they are now similar, except for the sign difference in the $(\vec{a} \cdot \vec{l})\vec{l}$ term. In fact, when we recognize, as was emphasized above, that mapping quaternion formulas into their equivalent 3D vector algebra formulas requires a change in the sign of any quaternion dot products, such as the $(\vec{a} \cdot \vec{l})$ term, the two formulations will become identical.

Hence, if we 1) mapped the quaternion vectors, $\vec{a}$ and $\vec{l}$, in Equation 75 into their 3D (real vector space) counterparts (reference the mapping rule of Equation 68) and then 2) change the sign of the quaternion dot product term so that it is appropriate for operating on the real $\vec{a}$ and $\vec{l}$ counterpart mappings, we would see that the two formulas are identical:

$$\vec{a}' = (q\,\vec{a})q^{-1} = cos(\Omega)\,\vec{a} + sin(\Omega)\,(\vec{l} \times \vec{a}) + (cos(\Omega) - 1)\,(\vec{a} \cdot \vec{l})\vec{l} \qquad (75)$$

$$\mapsto \qquad \vec{a}' = cos\,\Omega\,\vec{a} + (1 - cos\,\Omega)(\vec{a} \cdot \vec{l})\vec{l} + sin\,\Omega\,(\vec{l} \times \vec{a}) \qquad (9)$$

Note, in the Rodrigues formula the magnitude of the axis of rotation $\vec{l}$ was 1. In Equation 75, we are associating the direction of the quaternion vector $\vec{q}$ with the direction of $\vec{l}$, but the magnitude of $\vec{q}$ is less than 1, as it is equated to $\vec{l}$ scaled by $sin(\Omega/2)$ (Equation 73). Nevertheless, the magnitude of the quaternion itself, $q = q_o + \vec{q}$ is equal to 1, as proven below using Equations 72–73.

$$|q| = \sqrt{q\,q^*} = \sqrt{q_o^2 + \sum_{n=i,j,k} q_n^2} = \sqrt{cos^2(\Omega/2) + sin^2(\Omega/2)} = 1 \qquad (77)$$

As an aside, the unit quaternion that we have identified as useful for rotating quaternion vectors (Equation 75) can also be put into exponential form:

$$q = q_o + \vec{q} = \cos(\Omega/2) + \sin(\Omega/2)\,\vec{l} = e^{\vec{l}\,\Omega/2} \qquad (78)$$

This exponential form is shorter, notation-wise, and is often used in the literature, but we will continue to use (below) the two-term trig function form.

In summary, the unit quaternion contains information about both an axis (albeit imaginary-number based) and angle of rotation, whereas these are two separate but physically identifiable aspects of rotation in 3D vector space. The above development (arriving at Equation 75) shows how such a unit quaternion can be used to rotate vectors around a single axis in the same way the Rodrigues formula did, but in this case, with a different (quaternion) algebra. We might also emphasize that the final vector $\vec{a}'$ is referenced to the direction of the vector being rotated, $\vec{a}$, and of course the axis, $\vec{q}$ or $\vec{l}$, and the angle of rotation, Ω.

Section 7.2 describes how quaternion algebra can be used to track a 3D body of rotation and any vectors that may be referenced to that body, which could be a vector such as $\vec{a}$, in this single-axis rotational case. Moreover, we will show how quaternion algebra can be used to track the motion of a body that is rotating about multiple axes. In essence, we show how to find the equivalent quaternion-algebra expression of the direction cosine matrices, which described the rotation of a vector $\vec{a}$ about the three Euler axes by the three Euler angles, ψ, θ, and ϕ.

7.2 Multiple Axis Rotations (Intrinsic Euler Angle Case Study)

As was demonstrated above for the Rodrigues case study, the quaternion formulation for rotating a vector about a quaternion axis can be done independent of any particular coordinate frame, i.e., describing the resulting vector $\vec{a}'$ in terms of the vector that was rotated, $\vec{a}$; the axis, $\vec{q}$; and angle, Ω, of rotation. That said, it can be more useful from an application standpoint to decompose $\vec{a}$ and $\vec{q}$ into vector components along a quaternion reference frame, writing

$$\vec{a} = a_i\, i + a_j\, j + a_k\, k \tag{79}$$

$$\vec{q} = q_i\, i + q_j\, j + q_k\, k \tag{80}$$

then substituting Equations 79–80 into Equation 70 and collecting i, j, and k terms will yield:

$$\begin{aligned}
\vec{a}' = {} & (q\,\vec{a})q^{-1} = (q_o^2)\,\vec{a} + 2\,q_o\,(\vec{q} \times \vec{a}) - 2\,(\vec{q} \cdot \vec{a})\,\vec{q} + (\vec{q} \cdot \vec{q})\,\vec{a} \\
= {} & \left\{\left(q_0^2 + q_i^2 - q_j^2 - q_k^2\right)a_i + 2\left(q_j q_i - q_o q_k\right)a_j + 2\left(q_o q_j + q_k q_i\right)a_k\right\} i \\
& + \left\{2\left(q_o q_k + q_j q_i\right)a_i + \left(q_o^2 + q_j^2 - q_i^2 - q_k^2\right)a_j + 2\left(q_k q_j - q_o q_i\right)a_k\right\} j \\
& + \left\{2\left(q_k q_i - q_o q_j\right)a_i + 2\left(q_o q_i + q_j q_k\right)a_j + \left(q_o^2 + q_k^2 - q_i^2 - q_j^2\right)a_k\right\} k
\end{aligned} \tag{81}$$

This can be put in matrix form as well:

$$\begin{pmatrix} a_i' \\ a_j' \\ a_k' \end{pmatrix} = \begin{pmatrix} \left(q_0^2 + q_i^2 - q_j^2 - q_k^2\right) & 2\left(q_j q_i - q_o q_k\right) & 2\left(q_o q_j + q_k q_i\right) \\ 2\left(q_o q_k + q_j q_i\right) & \left(q_o^2 + q_j^2 - q_i^2 - q_k^2\right) & 2\left(q_k q_j - q_o q_i\right) \\ 2\left(q_k q_i - q_o q_j\right) & 2\left(q_o q_i + q_j q_k\right) & \left(q_o^2 + q_k^2 - q_i^2 - q_j^2\right) \end{pmatrix} \begin{pmatrix} a_i \\ a_j \\ a_k \end{pmatrix} \tag{82}$$

Although there is nothing added by decomposing $\vec{a}$ and $\vec{q}$ into their components along the quaternion bases axes, expressing any rotation in this fashion emphasizes that quaternion-based rotations (of both the vector being rotated and the axis of rotation) can always be composed from three orthogonal axis, *i*, *j*, and *k*. Equation 82 is also found throughout the literature (e.g., Henderson 1977, p. 7; Rogers 2003; Sonderegger 2022, p. 51). Some references develop/use the inverse of Equation 82, $(q^{-1}\vec{a}')q$, in which case the matrix elements in Equation 82 are mirrored about the diagonal.

However, as of yet, we have not specified what the angles of rotation about each of those three separate axes are. But in principle, having a matrix equation that determines the coefficients of a rotated vector $\vec{a}'$ on the left of Equation 82 in terms of the coefficients of a vector prior to its rotation $\vec{a}$ is analogous to what the DCM^{-1} matrix of Equation 26 provided. That is, if we associate the vector $\vec{a}$, nested in $q\,\vec{a}\,q^{-1}$, with the fixed vector $\vec{b}$ that is attached to a rotating body, then the matrix multiplication on the right of Equation 82 gives the projection of that rotated body vector, designated as $\vec{a}'$ after rotation, back onto the original coordinate frame.

That is, if we can find the quaternion $q = q_o + q_i\, i + q_j\, j + q_k\, k$ that represent the motion of the three Euler axes and ascribe the Euler rotation angles about those quaternion-representations of the Euler axes, then our resultant matrix above for transforming the coefficients of $\vec{a}$ prior to rotation (or, in the body that is being rotated) into the coefficients of $\vec{a}'$ after rotation should look like the DCM^{-1} matrix. Let us see if that is the case.

We seek to determine whether the above coefficient values for $\vec{a}' = (q\, \vec{a})q^{-1}$ in matrix Equation 82 are equal to the same coefficient values that would be obtained for the Euler DCM^{-1} matrix (Equation 26) operating on the coefficients of a body-attached vector $\vec{a}$; i.e.,

$$\vec{a}' = q(\phi, \theta, \psi)\vec{a}q(\phi, \theta, \psi)^{-1} \mapsto [\vec{a}']^{\mathrm{T}} = ? = \mathrm{DCM}^{-1}(\phi, \theta, \psi)\, [\vec{a}]^{\mathrm{T}} \tag{83}$$

(Note, in Equation 26 $\vec{a}$ represented the projection of a body vector $\vec{b}$ onto the fixed earth frame; whereas now, in Equation 83 and henceforth, we are changing our notation such that $\vec{a}$ will represent a body vector, and $\vec{a}'$ will represent its projection onto the earth frame.) To answer this question posed in Equation 83, we need to find how $q = q_o + q_i\, i + q_j\, j + q_k\, k$ relates to the trig functions of the Euler angles. We can describe the quaternion axes in terms of the Euler rotation axes as follows. Before doing this, we can explain why the word intrinsic was used in this section title; the body rotation is described about axes that are intrinsically associated with the body, these are the Euler axes. We will pose the same problem in the next section, but describe the angles about axes that remain fixed in the earth frame; this will be called an extrinsic angular analysis of the body-rotation problem.

The first Euler rotation axis is solely in the direction of the *ee3 axis*; hence, that axis would map into quaternion (imaginary) axis *k*. We know the first Euler rotation, by ψ about *ee3*, rotates the body's axes into *eb1'*, *eb2'*, and *eb3'* as shown in Table 1. Thus, the quaternion, q_ψ, used to achieve the first rotation, is constructed by multiplying the axis of rotation, *k*, by the sine of one half the (Euler) angle of rotation ($\psi/2$ in this case) and adding it to the cosine of $\psi/2$. Doing so,

$$q_\psi = cos(\psi/2) + sin(\psi/2)\, k \tag{84}$$

If we nest q_ψ and its inverse, q_ψ^{-1} around the vector to be rotated, which we take to be $\vec{a} = i$ (the starting *eb1* axis), then performing the quaternion multiplication $q_\psi(i)q_\psi^{-1}$ would yield the following quaternion expression for $\vec{a} = i$ after rotation by ψ around the earth's vertical axis (and the quaternion *k* axis):

$$\vec{a}' = q_\psi(i = eb1)q_\psi^{-1} = \{cos(\psi/2) + sin(\psi/2)\, k\}\ (i)\ \{cos(\psi/2) - sin(\psi/2)\, k\}$$

$$= \{cos^2(\psi/2) - sin^2(\psi/2)\}\ i\ + 2\ sin(\psi/2)\ j$$

$$= cos(\psi)\ i + sin(\psi)\ j = eb1' \qquad (85)$$

This would correspond to the CCW rotation of the vector i by angle ψ in the i–j plane, which maps to column 1 of Table 1, i.e., $eb1'$.

Likewise, if we wanted to find the quaternion-equivalent of rotating $\vec{a} = j$ (the starting $eb2$ axis), by ψ around the Earth's vertical axis, we would find

$$eb2' = q_\psi(j = eb2)q_\psi^{-1} = -\ sin(\psi)\ i + cos(\psi)\ j \qquad (86)$$

which maps to column 2 of Table 1.

Since we know that $eb2'$ is the Euler axis for the second Euler rotation by the angle θ, the quaternion for the second Euler rotation by θ would be

$$q_\theta = cos(\theta/2) + \{sin(\theta/2)\}\{eb2' = -\ sin(\psi)\ i + cos(\psi)\ j\}$$

$$= cos(\theta/2)\ - sin(\theta/2)\ sin(\psi)\ i + sin(\theta/2)cos(\psi)\ j \qquad (87)$$

such that the quaternion axis for the third Euler rotation axis, $eb1''$, which results from rotating $eb1'$ around $eb2'$ by θ, is given by

$$eb1'' = q_\theta(eb1' = cos(\psi)\ i + sin(\psi)\ j\ )q_\theta^{-1}$$

$$= \{cos(\theta/2)+sin(\theta/2)[cos(\psi)j-sin(\psi)i]\}\{cos(\psi)i\}\{cos(\theta/2)-sin(\theta/2)[cos(\psi)j-sin(\psi)i]\}$$

$$+ \{cos(\theta/2)+sin(\theta/2)[cos(\psi)j-sin(\psi)i]\}\{sin(\psi)j\}\{cos(\theta/2)-sin(\theta/2)[cos(\psi)j-sin(\psi)i]\}$$

$$= cos(\theta)cos(\psi)\ i + cos(\theta)sin(\psi)\ j - sin(\theta)\ k \qquad (88)$$

which maps to column 1 of Table 2. In a similar fashion, the quaternion equivalent of the third Euler rotation by ϕ about $eb1''$ would be obtained from

$$q_\phi = cos(\phi/2) + \{sin(\phi/2)\}\{\ eb1'' = cos(\theta)cos(\psi)\ i + cos(\theta)sin(\psi)\ j - sin(\theta)\ k\ \}$$

$$= cos(\phi/2) + sin(\phi/2)cos(\theta)cos(\psi)\ i + sin(\phi/2)cos(\theta)sin(\psi)\ j - sin(\phi/2)sin(\theta)\ k \qquad (89)$$

Thus, we have developed quaternion expressions for each of the three Euler axes. Collectively then, if we wanted to know what the quaternion equivalent of the three Euler rotations of a general vector, $\vec{a} = a_i\, i + a_j\, j + a_k\, k,$ we would conduct three consecutive quaternion rotations, analogous to the three consecutive Euler axes rotations, as prescribed below:

$$\vec{a}' = q(\phi,\theta,\psi)(\vec{a})q(\phi,\theta,\psi)^{-1}$$

$$= q_\phi\ (\ q_\theta\ [\ q_\psi\ \{\ a_i\, i + a_j\, j + a_k\, k\}\ q_\psi^{-1}\ ]\ q_\theta^{-1}\ )\ q_\phi^{-1} \qquad (90)$$

But our objective was to find a general $q = q_o + \vec{q} = q_o + q_i\, i + q_j\, j + q_k\, k$ for nesting around the vector being rotated, $\vec{a}$, in the expression $(q\,\vec{a})q^{-1}$. This is accomplished by computing the scalar and vector components of the quaternion product of $q_\phi\; q_\theta\; q_\psi$. That is, collecting the scalar and quaternion vector products that result from this product multiplications above will show the following:

$$q_o = cos(\phi/2)cos(\theta/2)cos(\psi/2) + sin(\phi/2)sin(\psi/2)sin(\theta/2) \tag{91}$$

$$q_i = sin(\phi/2)cos(\psi/2)cos(\theta/2) - cos(\phi/2)sin(\psi/2)sin(\theta/2) \tag{92}$$

$$q_j = cos(\phi/2)sin(\theta/2)cos(\psi/2) + sin(\phi/2)cos(\theta/2)sin(\psi/2) \tag{93}$$

$$q_k = cos(\phi/2)cos(\theta/2)sin(\psi/2) - sin(\phi/2)cos(\psi/2)sin(\theta/2) \tag{94}$$

Indeed, if these expressions are substituted in the quaternion matrix of Equation 82, it will produce the DCM^{-1}(ϕ, θ, ψ) matrix of Equation 26, such that Equation 82 becomes:

$$\begin{pmatrix} a_i' \\ a_j' \\ a_k' \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} a_i \\ a_j \\ a_k \end{pmatrix} \tag{95}$$

Equations 91–94 for q_o, q_i, q_j, q_k, as functions of the Euler angles are found throughout the literature (e.g., Kuipers 1999; Chudá 2019; Sonderegger 2022, p. 50).

Let's take a simple test case example: supposing $\phi = \theta = 0$, then Equation 95 produces

$$\begin{pmatrix} (cos(\psi)) & (-sin(\psi)) & (0) \\ (sin(\psi)) & (cos(\psi)) & (0) \\ (0) & (0) & (1) \end{pmatrix} \tag{96}$$

which is the same as DCM^{-1}(0,0, ψ) for this example.

Thus, we have proven that the q-matrix of Equation 82 is equivalent to the DCM^{-1}, but is there any computational efficiency in using Euler angles to evaluate q-matrix coefficients, versus using Euler angles to evaluate the DCM^{-1}? There are 14 2-tuples (examples given in Equation 97) and 1 singlet, with 4 add/subtracts, for a total of 14 multiplications and 4 add/subtracts (18 math operations) in the DCM^{-1}.

$$\begin{gathered} 1)\ c_\theta s_\psi,\ 2)\ c_\theta c_\psi,\ 3)\ c_\theta s_\phi,\ 4)\ c_\theta c_\phi,\ 5)\ s_\psi s_\phi,\ 6)\ s_\psi c_\phi, \\ 7)\ c_\psi s_\phi,\ 8)\ c_\psi c_\phi,\ 9)\ s_\phi s_\theta,\ 10)\ (s_\phi s_\theta)c_\psi,\ 11)\ (s_\phi s_\theta)s_\psi, \\ 12)\ c_\phi s_\theta,\ 13)\ (c_\phi s_\theta)s_\psi,\ 14)\ (c_\phi s_\theta)c_\psi \end{gathered} \tag{97}$$

The quaternion matrix of Equation 82 involves 10 distinct 2-tuples along with 12 add/subtracts (not counting sign change as a math operation) and 6 multiplications (by the number 2). Also, to evaluate each of the quaternion q coefficients using Equations 91–94 would require 12 more 2-tuples and 4 adds/subtracts for a total of 22 2-tuple multiplication, 16 add/subtracts, and 6 simple multiplication by the scalar 2, for a total of 44 math operations to fully evaluate the quaternion matrix using the trig functions. Clearly, the DCM^{-1} is a more computationally efficient approach (18 vs. 44 math operations) for use in evaluation $\vec{a}'$. Now, it is not the case that every math operation is computationally equivalent, but the degree to which the efficiency varies depends on the desired accuracy of the solution. Normally, a trig function evaluation using a Taylor series expansion for example would be far less efficient than a multiplication, but depending on the level of accuracy needed, and if memory is not a factor, a lookup table can be used to evaluate any trig function (ref. e.g., Singla et al., p. 4). For example, if we wish to round angles to the nearest 0.001°, and we don't use interpolation, we need a lookup table of 360,000 entries, but since trig functions repeat themselves every 90°, we only need to retain ¼ of those entries, or 90,000 values. If we round to fewer decimal places the necessary memory would be less, or if we use interpolation between fewer stored trig entries, the necessary memory would also be less, but there would be at least two math operations (for linear, or more for higher order interpolation schemes) needed to compute an interpolated trig function value.

For example, interpolation-wise, if we stored the trig entries, for say s_θ, for every degree from 0 to 90, and we computed the 89 slope values between each of those entries (requiring a total of 179 stored entries) then using linear interpolation to compute s_θ for any angle between 0 to 90, would entail two math operations using the traditional linear interpolation formula (where there is one multiplication-operation, of the slope times the interpolated angle value, plus one add-operation of that multiplication product to the nearest-lower stored trig value). Thus, using linear interpolation, we might interpolate say ten trig function values between each stored trig function entry, reducing the stored memory by a factor of five (i.e., instead of storing those ten trig function values we only need to store two, one trig value and one slope between it and the nearest trig value). However, for simplicity in this analysis, we will assume that all math operations are computationally equivalent, i.e., we will assume that we can use a look-up table for all functional values.

We would still have to multiple either the 3 × 3 DCM^{-1} of Equation 95 or the quaternion q matrix of Equation 82 by the 3 × 1 $\vec{a}$ matrix (another 9 multiplications and 6 adds; i.e., 15 more operations) to completely express all components of the

$\vec{a}'$ in terms of the $\vec{a}$ components and trig functions (hence, 18 [or 44] + 15 = 33 [or 59] operations).

In addition, if we want to update $\vec{a}'$ at a time Δt later, we will need to know the angular rates at the current body orientation (i.e., the current angles). From Equation 47, this would require nine math operations (multiplication, division, adds or subtracts) within/among the trig functions and the gyro rates. For example,

$$1)\ c_\phi R,\ \ 2)\ s_\phi Q,\ \ 3)\ c_\phi R + s_\phi Q,\ \ 4)\ (c_\phi R + s_\phi Q)/c_\theta,\ \ 5)\ s_\theta(c_\phi R + s_\phi Q)/c_\theta,$$

$$6)\ s_\theta(c_\phi R + s_\phi Q)/c_\theta + P,\ \ 7)\ -s_\phi R,\ \ 8)\ c_\phi Q,\ \ 9)\ -s_\phi R + c_\phi Q \qquad (98)$$

We would also have to specify the time interval Δt, and multiply it by the angular rates (three more operations), and, finally, we would have to add this rate-caused change in angle to the current value of the angle (three additional operations) for a total of 15 operations to update the Euler angles. Adding these 15 angle-update operations to the 33 operations needed to update $\vec{a}'$ using the DCM^{-1} formulation of Equation 26 or 95, or adding them to the 59 operations needed to update $\vec{a}'$ using the q matrix operator of Equation 82, yields a total of 48 operations for the former and 74 operations for the latter to update $\vec{a}'$ in time. Figure 5 summarizes these sequences of math operations.

Although updating $\vec{a}'$ by using Euler angles in Equations 91–94 to evaluate the q-matrix is computationally more involved than directly using Euler angles to evaluate the DCM^{-1}, we next show how Equations 91–94 can be used to find a more efficient solution for updating $\vec{a}'$; moreover, one that avoids updating Euler angles entirely. This will come about by finding the quaternion-assisted solution for determining the body's angular velocity.

start

$$\begin{pmatrix} a_i' \\ a_j' \\ a_k' \end{pmatrix} = \underbrace{\begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} a_i \\ a_j \\ a_k \end{pmatrix}}_{\text{33 math operations}}$$

$$\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\sin\theta\cos\phi}{\cos\theta} & \frac{\sin\theta\sin\phi}{\cos\theta} & 1 \\ -\sin\phi & \cos\phi & 0 \\ \frac{\cos\phi}{\cos\theta} & \frac{\sin\phi}{\cos\theta} & 0 \end{bmatrix} \begin{bmatrix} R \\ Q \\ P \end{bmatrix}}_{\text{9 math operations}} \quad \underbrace{\begin{matrix} \phi(t+\Delta t) = \dot\phi\,\Delta t + \phi(t) \\ \theta\,(t+\Delta t) = \dot\theta\,\Delta t + \theta(t) \\ \psi\,(t+\Delta t) = \dot\psi\,\Delta t + \psi(t) \end{matrix}}_{\text{6 math operations}}$$

iterate on Δt

DCM^{-1} 48 math operations

$$\underbrace{\begin{aligned} q_o &= cos(\phi/2)cos(\theta/2)cos(\psi/2) + sin(\phi/2)sin(\psi/2)sin(\theta/2) \\ q_i &= sin(\phi/2)cos(\psi/2)cos(\theta/2) - cos(\phi/2)sin(\psi/2)sin(\theta/2) \\ q_j &= cos(\phi/2)sin(\theta/2)cos(\psi/2) + sin(\phi/2)cos(\theta/2)sin(\psi/2) \\ q_k &= cos(\phi/2)cos(\theta/2)sin(\psi/2) - sin(\phi/2)cos(\psi/2)sin(\theta/2) \end{aligned}}_{\text{16 math operations}}$$

start

$$\begin{bmatrix} a_i' \\ a_j' \\ a_k' \end{bmatrix} = \underbrace{\begin{bmatrix} (q_o^2 + q_i^2 - q_j^2 - q_k^2) & 2(q_j q_i - q_o q_k) & 2(q_o q_j + q_k q_i) \\ 2(q_o q_k + q_j q_i) & (q_o^2 + q_j^2 - q_i^2 - q_k^2) & 2(q_k q_j - q_o q_i) \\ 2(q_k q_i - q_o q_j) & 2(q_o q_i + q_j q_k) & (q_o^2 + q_k^2 - q_i^2 - q_j^2) \end{bmatrix} \begin{bmatrix} a_i \\ a_j \\ a_k \end{bmatrix}}_{\text{43 math operations}}$$

iterate on Δt

q matrix 74 math operations

Figure 5. Comparison of math operation needed to update $\vec{a}'$ using Euler angle updates to the DCM^{-1} vs. q matrix.

In review, the Euler angle rotation rate $\dot\psi$ about the vertical *eb3* = *ee3* axes maps into a quaternion rate, designated here as *wee3*, about the quaternion equivalent of the *ee3* axes, namely the *k* axis (which would be the same as the body's eb3 axis if the body only rotated about the ee3 axes).

$$\dot\psi\ (ee3)\ \mapsto\ \overrightarrow{wee3} = \dot\psi\ (k) \tag{99}$$

The angular velocity $\dot\theta$ about the *eb2'* axis (after its rotation by ψ), would map into a quaternion angular velocity about that specific orientation of the body's eb2 axes

$$\dot\theta\ (eb2')\ \mapsto\ \overrightarrow{web2'}\ = \dot\theta\ (\ cos(\psi)\ j - sin(\psi)\ i) \tag{100}$$

and the quaternion-equivalent of the angular velocity $\dot\phi$ about the third Euler axis *eb1''* (after its rotation by ψ then θ), would map into a quaternion angular velocity about that axis

$$\dot\phi\ (\ eb1'')\ \mapsto\ \overrightarrow{web1''}\ = \dot\phi\ (\ cos(\theta)cos(\psi)\ i + cos(\theta)sin(\psi)\ j - sin(\theta)\ k\ ) \tag{101}$$

There is a distinction here between $\overrightarrow{web2'}$ and $\overrightarrow{wb2}$ ($= Q\ \ eb2$), as well as between $\overrightarrow{web1''}$ and $\overrightarrow{wb1}$ ($= P\ eb1$), the later are angular velocities about the eb2 and eb1 axes, regardless of what the previous rotations were that brought the body to its current orientation; whereas the former are specifically the angular velocities about the body's eb2 and eb1 axes after rotation (first) by ψ about the earth's vertical axis and (second) by θ about the body's $eb2'$ axis (Tables 1 and 2).

The angular velocity about each of the three (orthogonal) quaternion axes, *i*, *j*, and *k*—which we will denote as $\overrightarrow{wi}$, $\overrightarrow{wj}$, and $\overrightarrow{wk}$, respectively—can be obtained from the three quaternion expressions for angular velocities about the three (nonorthogonal) quaternion-representations of the Euler axes ($\overrightarrow{wee3}$, $\overrightarrow{web2'}$, and $\overrightarrow{web1''}$) by collecting their *i*, *j*, and *k* components (or, think of projecting each of these quaternion angular velocities vectors onto the quaternion representation of the earth axes, *i*, *j*, and *k*). Thus,

$$\begin{aligned}\overrightarrow{wi} + \overrightarrow{wj} + \overrightarrow{wk} &= \overrightarrow{wee3} + \overrightarrow{web2'} + \overrightarrow{web1''} \\ &= \dot{\psi}\,(k) + \dot{\theta}\,(\,cos(\psi)\;j - sin(\psi)\;i) \\ &\quad + \dot{\phi}\,(\,cos(\theta)cos(\psi)\,i + cos(\theta)sin(\psi)\,j - sin(\theta)\,k\,)\end{aligned} \tag{102}$$

Collecting vector coefficient terms:

$$\overrightarrow{wi} = [\,\dot{\phi}\;cos(\theta)cos(\psi) - \dot{\theta}\;sin(\psi)\,]\;i \tag{103}$$

$$\overrightarrow{wj} = [\,\dot{\theta}\;cos(\psi) + \dot{\phi}\;cos(\theta)sin(\psi)\,]\;j \tag{104}$$

$$\overrightarrow{wk} = [\,\dot{\psi}\; - \;\dot{\phi}\;sin(\theta)]\;k \tag{105}$$

Or, in matrix form:

$$\begin{pmatrix} wi \\ wj \\ wk \end{pmatrix} = \begin{pmatrix} cos(\theta)cos(\psi) & -sin(\psi) & 0 \\ cos(\theta)sin(\psi) & cos(\psi) & 0 \\ -sin(\theta) & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{106}$$

where *wi*, *wj* and *wk* on the left of Equation 106 represent the magnitudes of their vector velocity counterparts in Equations 103–105.

Recall Equation 47 (reproduced below) relates the Euler angular rates to the body gyro rates, *P*, *Q*, and *R* (which are equivalent to *wb1*, *wb2*, and *wb3*):

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s_\theta s_\phi / c_\theta & s_\theta c_\phi / c_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix} \tag{47}$$

Ignoring the aforementioned singularity condition if $\theta = \pi/2$, and substituting the relationships described in Equation 47 into Equation 106, we arrive at

$$\begin{pmatrix} wi \\ wj \\ wk \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

$$= \mathrm{DCM}^{-1}\,[P,Q,R]^{\mathrm{T}} \tag{107}$$

Likewise, $[P,Q,R]^{\mathrm{T}} = \mathrm{DCM}\,[wi,\ wj,\ wk]^{\mathrm{T}}$. We can perform the matrix multiplication in Equation 107, then add the three vector angular velocities on the left and express them as vector collections of the coefficients of P, Q and R; whence we will recognize from Equation 10 that the total angular velocity in the i, j, and k frame is equivalent to the total angular velocity in the body frame. That is,

$$\vec{w} = \overrightarrow{wi} + \overrightarrow{wj} + \overrightarrow{wk} = P\,(c_\theta c_\psi\, i + c_\theta s_\psi\, j - s_\theta\, k)$$

$$+ Q\,(\{s_\phi s_\theta c_\psi - s_\psi c_\phi\}\, i + \{s_\phi s_\theta s_\psi + c_\psi c_\phi\}\, j + \{c_\theta s_\phi\}\, k)$$

$$+ R\,(\{c_\phi s_\theta c_\psi + s_\psi s_\phi\}\, i + \{c_\phi s_\theta s_\psi - c_\psi s_\phi\}\, j + \{c_\theta c_\phi\}\, k)$$

$$= P\, eb1 + Q\, eb2 + R\, eb3 = \overrightarrow{wb} \tag{108}$$

which is what we should expect since the *eb1* body axis is associated with gyro rate *P*, *Q* is aligned with *eb2,* and *R* is directed along *eb3*.

Suppose $\phi = \theta = 0$, and $\psi = \pi/2$, then Equation 107 or 108 would yield

$$\begin{pmatrix} wi \\ wj \\ wk \end{pmatrix} = \begin{pmatrix} -Q \\ P \\ R \end{pmatrix} \tag{109}$$

This simple test case result is what we would expect if the body's *eb1* and *eb2* axes are rotated by $\pi/2$ about the k axis, along with their gyro rates P and Q, respectively, into alignment with the j and $-i$ axes, respectively; while *eb3* axis and gyro rate R is unchanged by rotation about k, hence, $wk = R$.

Differentiating the expressions for q_o, q_i, q_j and q_k, given in Equations 91–94 would yield $\dot{q}_o, \dot{q}_i, \dot{q}_j$ and $\dot{q}_k$ in terms of $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$, and trig functions of the half angles $\psi/2$, $\theta/2$, and $\phi/2$. Since Equation 106 relates $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$ to *wi*, *wj*, and *wk* as well as the trig functions of the full angles ψ and θ, we will need to convert trig functions of full angles to trig functions of half angles and judiciously rearrange and combine the products; doing so (as enumerated in the Appendix A), we can show the following:

$$\dot{q}_o = -(1/2)(wi q_i + wj q_j + wk q_k) \tag{110}$$

$$\dot{q}_i = -(1/2)(wk q_j - wj q_k - wi q_o) \tag{111}$$

$$\dot{q}_j = -(1/2)(-wk q_i + wi q_k - wj q_o) \tag{112}$$

$$\dot{q}_k = -(1/2)(wj q_i - wi q_j - wk q_o) \tag{113}$$

Or we can put these four equations into one of two matrix forms:

$$\begin{bmatrix} \dot{q}_o \\ \dot{q}_i \\ \dot{q}_j \\ \dot{q}_k \end{bmatrix} = (1/2) \begin{bmatrix} 0 & -wi & -wj & -wk \\ wi & 0 & -wk & wj \\ wj & wk & 0 & -wi \\ wk & -wj & wi & 0 \end{bmatrix} \begin{bmatrix} q_o \\ q_i \\ q_j \\ q_k \end{bmatrix} = (1/2) \begin{bmatrix} q_o & -q_i & -q_j & -q_k \\ q_i & q_o & q_k & -q_j \\ q_j & -q_k & q_o & q_i \\ q_k & q_j & -q_i & q_o \end{bmatrix} \begin{bmatrix} 0 \\ wi \\ wj \\ wk \end{bmatrix} \quad (114)$$

Alternatively, this could be written as a quaternion product, in the following form:

$$\frac{dq}{dt} = (1/2)[\,(\vec{w})q_o + (\vec{w} \cdot \vec{q}) + (\vec{w} \times \vec{q})\ ] = (1/2)\vec{w}q \qquad (115)$$

where $\vec{w} \cdot \vec{q}$ is a quaternion-vector dot product, not a Euclidean-vector dot product. Equations 110–115 can found throughout the literature (e.g., Schwab 2002; Sabatini 2005; Graf 2007, p. 7; Narayan 2017; Baker 2023a).

It is important to recognize that the components of angular velocity (*wi*, *wj*, and *wk*) in Equation 116 are referenced to the earth-fixed coordinate frame; i.e., they are the angular velocities around the *ee1*, *ee2*, and *ee3* axes, respectively.

The equivalent expression to Equation 115 as a function of angular velocity about the body axes can be derived as follows. First, multiply Equation 115 by the identity in the form qq^{-1}=1 (Equation 67), such that we have

$$\begin{aligned} \frac{dq}{dt} &= qq^{-1}(1/2)\vec{w}q = (1/2)q(q^{-1}\vec{w}q) = (1/2)q\left(\overrightarrow{wb}\right) \\ &= (1/2)[\,q_o\left(\overrightarrow{wb}\right) + (\vec{q} \cdot \overrightarrow{wb}) + (\vec{q} \times \overrightarrow{wb})\,] \end{aligned} \qquad (116)$$

Reference to Equation 116 can be found in the literature (e.g., Schwab 2002; Graf 2007). From Equation 108,

$$\vec{w} = (wi)\,i + (wj)\,j + (wk)k = \overrightarrow{wb} = (wbi)\,i + (wbj)\,j + (wbk)\,k \quad (117)$$

To preserve the context of Equation 116 we will use the body syntax of the angular velocity and put the equation in matrix form as follows:

$$\begin{bmatrix} \dot{q}_o \\ \dot{q}_i \\ \dot{q}_j \\ \dot{q}_k \end{bmatrix} = (1/2) \begin{bmatrix} 0 & -wbi & -wbj & -wbk \\ wbi & 0 & wbk & -wbj \\ wbj & -wbk & 0 & wbi \\ wbk & wbj & -wbi & 0 \end{bmatrix} \begin{bmatrix} q_o \\ q_i \\ q_j \\ q_k \end{bmatrix} = (1/2) \begin{bmatrix} q_o & -q_i & -q_j & -q_k \\ q_i & q_o & -q_k & q_j \\ q_j & q_k & q_o & -q_i \\ q_k & -q_j & q_i & q_o \end{bmatrix} \begin{bmatrix} 0 \\ wbi \\ wbj \\ wbk \end{bmatrix} \quad (118)$$

Equation 118 can be found in the literature (e.g., Narayan 2017, p. 6).

Also from Equation 108, $\overrightarrow{wb} = P\ eb1 + Q\ eb2 + R\ eb3$; hence

$$\overrightarrow{wb} = (wbi)\; i + (wbj)\; j\; + (wbk)\; k = \; (P\; eb1 + Q\; eb2 + R\; eb3)_i\; i$$
$$+ \; (P\; eb1 + Q\; eb2 + R\; eb3)_j\; j\; + (P\; eb1 + Q\; eb2 + R\; eb3)_k\; k$$
$$= \; (P\; \{c_\theta c_\psi\} + Q\; \{s_\phi s_\theta c_\psi - s_\psi c_\phi\} + R\; \{c_\phi s_\theta c_\psi + s_\psi s_\phi\})\; i$$
$$+(P\; \{c_\theta s_\psi\} + Q\; \{s_\phi s_\theta s_\psi + c_\psi c_\phi\} + R\; \{c_\phi s_\theta s_\psi - c_\psi s_\phi\})\; j$$
$$+ \; (P\; (\{-s_\theta\} + Q\{\{c_\theta s_\phi\}\}\; j + \{c_\theta c_\phi\})\; k \tag{119}$$

At time zero, the body axes are aligned with the earth axes, $\psi = \theta = \phi = 0$, and Equation 119 reduces to simply

$$\overrightarrow{wb} = P\; i + Q\; j\; + R\; k \tag{120}$$

Hence, initially and only initially, Equation 118 becomes the following:

$$\begin{bmatrix} \dot{q}_o \\ \dot{q}_i \\ \dot{q}_j \\ \dot{q}_k \end{bmatrix} = (1/2) \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_o \\ q_i \\ q_j \\ q_k \end{bmatrix} = (1/2) \begin{bmatrix} q_o & -q_i & -q_j & -q_k \\ q_i & q_o & -q_k & q_j \\ q_j & q_k & q_o & -q_i \\ q_k & -q_j & q_i & q_o \end{bmatrix} \begin{bmatrix} 0 \\ P \\ Q \\ R \end{bmatrix} \tag{121}$$

Reference to Equation 121 can be found in the literature (e.g., Phillips et al. 2001, p. 727; Rogers 2003; Kim 2013, p. 94; Ueno et al. 2014, p. 4; Kartiwa et al. 2023).

The significance of how simplified the equations governing the update of the quaternion coefficients becomes at time zero, reducing from, Equation 118 to Equation 121, should not be underappreciated, because there are no trig functions involved in the update from time zero forward. That is, the quaternion coefficients $q_o, q_i, q_j,$ and q_k can be iteratively evaluated from time zero forward without complications from the Euler angle rate singularity problem if $\theta = \pi/2$ (Equation 47). That is, $q_o, q_i, q_j,$ and q_k can be evaluated without reliance on the Euler angle trig functions present in Equations 91–94, or without needing the trig functions to evaluate $\vec{w}$ in Equation 107 for use in Equation 117.

Computationally, each of the q-coefficient time derivatives on the left column matrix of Equation 121 would require 3 multiplications (1 for each quaternion coefficient times an angular rate, e.g., $-q_i P$), plus 2 add/subtracts to combine those 3 products, then 1 more multiplication of that sum by ½, for a total of 6 operations per row of the matrix multiplication, times 4 rows, or 24 operations to evaluate the left of Equation 121. Then we would have to multiple each time derivative in the left column matrix by a Δt, and add these quaternion-rate updates to the current quaternion value, which brings the total to eight math operations to update each quaternion coefficient (q_o, q_i, q_j, and q_k), as shown in Equation 122.

$$\begin{aligned} q_o\,(t+\Delta t) &= \dot{q}_o \Delta t + q_o(t) \\ q_i\,(t+\Delta t) &= \dot{q}_i \Delta t + q_i(t) \\ q_j\,(t+\Delta t) &= \dot{q}_j \Delta t + q_j(t) \\ q_k\,(t+\Delta t) &= \dot{q}_k \Delta t + q_k(t) \end{aligned} \tag{122}$$

Thus, the 24 operations needed to evaluate Equation 121, plus the 8 operations needed to evaluate Equation 122, means 32 math operations are needed to evaluate the coefficients for use in Equation 82. We previously mentioned (in the explanation of Figure 5) that there are 28 math operations to evaluate the q-matrix using the quaternion coefficients, and 15 math operations needed to multiply the 3 × 3 q-matrix times the 3 × 1 a-matrix (43 more math operations). Hence, by using Equations 121 and 122 to solve for the q-coefficients and then substituting them in Equation 82, we would be able to update $\vec{a}'$ with a total of 75 (= 32 + 43) math operations. A schematic of the process is shown in Figure 6. This is roughly the same (75 vs. 74) if we used Equations 91–94 to evaluate the q-coefficients in Equation 82, but we do not have to be concerned with avoiding the singularity causing pitch angle if $\theta = \pi/2$.

DCM^{-1} — 48 math operations

start

$$\begin{pmatrix} a'_i \\ a'_j \\ a'_k \end{pmatrix} = \underbrace{\begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} a_i \\ a_j \\ a_k \end{pmatrix}}_{\text{33 math operations}}$$

iterate on Δt

$$\underbrace{\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \begin{bmatrix} \frac{\sin\theta\cos\phi}{\boldsymbol{\cos\theta}} & \frac{\sin\theta\sin\phi}{\boldsymbol{\cos\theta}} & 1 \\ -\sin\phi & \cos\phi & 0 \\ \frac{\cos\phi}{\boldsymbol{\cos\theta}} & \frac{\sin\phi}{\boldsymbol{\cos\theta}} & 0 \end{bmatrix} \begin{bmatrix} R \\ Q \\ P \end{bmatrix}}_{\text{9 math operations}} \qquad \underbrace{\begin{aligned} \phi(t+\Delta t) &= \dot\phi\,\Delta t + \phi(t) \\ \theta(t+\Delta t) &= \dot\theta\,\Delta t + \theta(t) \\ \psi(t+\Delta t) &= \dot\psi\,\Delta t + \psi(t) \end{aligned}}_{\text{6 math operations}}$$

---

q matrix — 75 math operations

iterate on Δt

start

$$\underbrace{\begin{bmatrix} \dot q_o \\ \dot q_i \\ \dot q_j \\ \dot q_k \end{bmatrix} = (1/2) \begin{bmatrix} q_o & -q_i & -q_j & -q_k \\ q_i & q_o & -q_k & q_j \\ q_j & q_k & q_o & -q_i \\ q_k & -q_j & q_i & q_o \end{bmatrix} \begin{bmatrix} 0 \\ P \\ Q \\ R \end{bmatrix} \qquad \begin{aligned} q_o(t+\Delta t) &= \dot q_o\Delta t + q_o(t) \\ q_i(t+\Delta t) &= \dot q_i\Delta t + q_i(t) \\ q_j(t+\Delta t) &= \dot q_j\Delta t + q_j(t) \\ q_k(t+\Delta t) &= \dot q_k\Delta t + q_k(t) \end{aligned}}_{\text{32 math operations}}$$

$$\begin{bmatrix} a'_i \\ a'_j \\ a'_k \end{bmatrix} = \underbrace{\begin{bmatrix} (q_o^2 + q_i^2 - q_j^2 - q_k^2) & 2(q_j q_i - q_o q_k) & 2(q_o q_j + q_k q_i) \\ 2(q_o q_k + q_j q_i) & (q_o^2 + q_j^2 - q_i^2 - q_k^2) & 2(q_k q_j - q_o q_i) \\ 2(q_k q_i - q_o q_j) & 2(q_o q_i + q_j q_k) & (q_o^2 + q_k^2 - q_i^2 - q_j^2) \end{bmatrix} \begin{bmatrix} a_i \\ a_j \\ a_k \end{bmatrix}}_{\text{43 math operations}}$$

Figure 6. Comparison of math operation needed to update $\vec{a}'$ using the DCM^{-1} vs. time-integrating the *q*-matrix coefficients, starting from *t* = 0 then using $\boldsymbol{q(t+\Delta t) = \dot q \Delta t + q(t)}$.

To this point, we have shown that using the DCM^{-1}solution is more efficient than using quaternions, provided that we are not near $\theta = \pi/2$; however, if that angle comes into play, then even though the quaternion formulation of Figure 6 has more math operations (75 vs. 48), it will not have a singularity problem at $\theta = \pi/2$. That said, we will show in the next section that there is a method for solving the DCM^{-1} without encountering the singularity condition. However, it will involve adding several more math operations to the current 48.

Before doing that, we are at a place in the mathematical development where it is straightforward to derive yet another alternative solution to updating $\vec{a}'$, though it will take some detailed intermediate derivations. This alternative solution starts with the idea that we can update $\vec{a}'$ over time if we know the time rate of change of $\vec{a}'$, which we can find as follows:

$$\frac{d\vec{a}'}{dt} = \frac{d(q\vec{a})q^{-1}}{dt} = \dot q(\vec{a}q^{-1}) + q\frac{d\vec{a}}{dt}q^{-1} + (q\vec{a})(\dot{q^{-1}}) \tag{123}$$

We will assume that $\vec{a}$ is stationary in the moving body frame (so that $\frac{d\vec{a}}{dt} = 0$), in which case all we need to know is the time derivatives, $\dot{q}$ and $\dot{q^{-1}}$. Continuing then, Equation 123 simplifies to

$$\frac{d\vec{a}\prime}{dt} = \dot{q}(\vec{a}q^{-1}) + (q\vec{a})(\dot{q^{-1}}) \tag{124}$$

Since $qq^{-1} = q^{-1}q = 1$ (Equation 67), we can insert this form of unity ("1") into Equation 124 as follows:

$$\begin{aligned} \frac{d\vec{a}\prime}{dt} &= \dot{q}(q^{-1}q)(\vec{a}q^{-1}) + (q\vec{a})(q^{-1}q)(\dot{q^{-1}}) \\ &= \dot{q}(q^{-1})(q\vec{a}q^{-1}) + (q\vec{a}q^{-1})(q)(\dot{q^{-1}}) \\ &= \dot{q}(q^{-1})(\vec{a}') + (\vec{a}')(q)(\dot{q^{-1}}) \end{aligned} \tag{125}$$

Also, since $qq^{-1} = 1$,

$$\frac{dqq^{-1}}{dt} = \frac{d1}{dt} = 0 = (\dot{q})q^{-1} + q(\dot{q^{-1}}) \tag{126}$$

so that

$$(\dot{q})q^{-1} = -q(\dot{q^{-1}}) \tag{127}$$

and hence,

$$\frac{d\vec{a}\prime}{dt} = -q(\dot{q^{-1}})(\vec{a}') + (\vec{a}')(q)(\dot{q^{-1}}) \tag{128}$$

From the definition of quaternion multiplication, the $q(\dot{q^{-1}})$ terms in Equation 128 can be written as follows:

$$q(\dot{q^{-1}}) = q \cdot (\dot{q^{-1}}) + q \times (\dot{q^{-1}}) \tag{129}$$

If we evaluate just the dot product term in Equation 129, we see

$$q \cdot (\dot{q^{-1}}) = q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + q_0(\dot{q_o^{-1}}) + \vec{q} \cdot (\dot{\vec{q}^{-1}}) \tag{130}$$

Since the inverse of a scalar is that scalar, the 3rd term in Equation 130 can be written as

$$q_0(\dot{q_o^{-1}}) = q_0(\dot{q_0}) = (1/2)\frac{dq_o^2}{dt} \tag{131}$$

The 4th term in Equation 130 can be written as

$$\begin{aligned} \vec{q} \cdot (\dot{\vec{q}^{-1}}) &= (q_i\, i + q_j\, j + q_k\, k) \cdot (\frac{d(-q_i\, i - q_j\, j - q_k\, k)}{dt}) \\ &= (q_i\dot{q_\iota} + q_j\dot{q_\jmath} + q_k\dot{q_k}) = (1/2)(\frac{dq_i^2}{dt} + \frac{dq_j^2}{dt} + \frac{dq_k^2}{dt}) \end{aligned} \tag{132}$$

Thus, the sum of the last two terms in Equation 130 (as expressed in Equations 131–132) equate to the following:

$$q_0(\dot{q_o^{-1}}) + \vec{q} \cdot (\dot{\vec{q}^{-1}}) = (1/2)(\frac{dq_o^2}{dt} + \frac{dq_i^2}{dt} + \frac{dq_j^2}{dt} + \frac{dq_k^2}{dt})$$
$$= (1/2)\frac{d|q|^2}{dt} = (1/2)\frac{d1}{dt} = 0 \qquad (133)$$

(where we are using the constraint that q is of unit magnitude [Equation 77]).

The quaternion dot product on the right of Equation 129, expanded as shown in Equation 130, and reduced by the equality shown in Equation 133, becomes

$$q \cdot (\dot{q^{-1}}) = q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) \qquad (134)$$

(Equation 134 is an example that shows the dot product of quaternions is not always a scalar, as it was when used in 3D vector algebra).

If we now turn attention to the quaternion cross product term, $q \times (\dot{q^{-1}})$, we can expand the quaternions on the right of Equation 129 into their scalar and vector terms, as follows:

$$q \times (\dot{q^{-1}}) = (q_0 + \vec{q}) \times (\dot{q_o^{-1}} + \dot{\vec{q}^{-1}}) \qquad (135)$$

Since the cross product of any scalar with another scalar or any scalar with a vector is zero (substantiated for example by comparing the terms in the full complex variable expansion of the quaternion product given by Equation 59 with consolidated alternative quaternion product formulation given in Equation 60), Equation 135 becomes

$$q \times (\dot{q^{-1}}) = \vec{q} \times \dot{\vec{q}^{-1}} \qquad (136)$$

Hence, substituting Equations 134 and 135 into Equation 129, we find

$$q(\dot{q^{-1}}) = q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}} \qquad (137)$$

Making use of Equation 137 in Equation 128,

$$\frac{d\vec{a}\prime}{dt} = -\, q(\dot{q^{-1}})(\vec{a}') + (\vec{a}')(q)(\dot{q^{-1}})$$
$$= -\,[q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}](\vec{a}') + (\vec{a}')[q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}]$$
$$= -\,[q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}] \cdot \vec{a}' + \vec{a}' \cdot [q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}]$$
$$-\,[q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}] \times \vec{a}' + \vec{a}' \times [q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \vec{q} \times \dot{\vec{q}^{-1}}]$$
$$(138)$$

Factoring out scalar terms, and since the dot product between two quaternion vectors is commutive (Equation 58),

$$\vec{q}(\dot{q_o^{-1}}) \cdot (\vec{a}') \;=\; (\dot{q_o^{-1}})\vec{q} \cdot (\vec{a}') \;=\; (\dot{q_o^{-1}})(\vec{a}') \cdot \vec{q} \;=\; (\vec{a}') \cdot \vec{q}(\dot{q_o^{-1}}) \qquad (139)$$

and

$$q_o(\dot{\vec{q}^{-1}}) \cdot (\vec{a}') \;=\; (\vec{a}') \cdot q_o(\dot{\vec{q}^{-1}}) \qquad (140)$$

Moreover, since $\vec{q} \times \dot{\vec{q}^{-1}}$ is a vector, we can be assured that

$$[\, \vec{q} \times \dot{\vec{q}^{-1}} \,] \cdot (\vec{a}') \;=\; (\vec{a}') \cdot [\, \vec{q} \times \dot{\vec{q}^{-1}}] \qquad (141)$$

Based on the equalities shown in Equations 139–141, all six quaternion-vector dot product terms in Equation 138 cancel each other out, so we are left with

$$\frac{d\vec{a}'}{dt} \;=\; -\,[\, q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}} \,] \times (\vec{a}')$$

$$+ (\vec{a}') \times [q_o(\dot{\vec{q}^{-1}}) \;+\; \vec{q}(\dot{q_o^{-1}}) \;+\; \vec{q} \times \dot{\vec{q}^{-1}}] \qquad (142)$$

Switching the order of the negative cross product in Equation 142 makes it a positive cross product, i.e.,

$$-\,[q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}}] \times (\vec{a}')$$

$$=\; (\vec{a}') \times [q_o(\dot{\vec{q}^{-1}}) \;+\; \vec{q}(\dot{q_o^{-1}}) \;+ \vec{q} \times \dot{\vec{q}^{-1}} \qquad (143)$$

so that Equation 142 simplifies further to become

$$\frac{d\vec{a}'}{dt} \;=\; 2(\vec{a}') \times [q_o(\dot{\vec{q}^{-1}}) \;+\; \vec{q}(\dot{q_o^{-1}}) \;+\; \vec{q} \times \dot{\vec{q}^{-1}}] \qquad (144)$$

We can create a more concise form of this expression for $\frac{d\vec{a}'}{dt}$, if we add back in the zero term of Equation 133, namely, $\{q_0(\dot{q_o^{-1}}) + \vec{q} \cdot (\dot{\vec{q}^{-1}})\} \;=\; 0$, then

$$\frac{d\vec{a}'}{dt} \;=\; 2(\vec{a}') \times [\,\{q_0(\dot{q_o^{-1}}) + \vec{q} \cdot (\dot{\vec{q}^{-1}})\} + q_o(\dot{\vec{q}^{-1}}) + \vec{q}(\dot{q_o^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}}] \qquad (145)$$

rearranging and regrouping, then recombing

$$\begin{aligned}
\frac{d\vec{a}'}{dt} &= 2(\vec{a}') \times [\,\{q_0(\dot{q_o^{-1}}) + q_o(\dot{\vec{q}^{-1}}) + \vec{q} \cdot (\dot{\vec{q}^{-1}})\} + \vec{q}(\dot{q_o^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}}] \\
&= 2(\vec{a}') \times [\, q_0(\dot{q^{-1}}) + \vec{q} \cdot (\dot{q^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}}] \\
&= 2(\vec{a}') \times [\, (q_0 + \vec{q}) \cdot (\dot{q^{-1}}) + \; \vec{q} \times \dot{\vec{q}^{-1}}] \\
&= 2(\vec{a}') \times [q(\dot{q^{-1}})] \qquad (146)
\end{aligned}$$

Or, from Equation 127, which showed that $(\dot{q})q^{-1} = -\,q(\dot{q^{-1}})$, we could also write Equation 146 as

$$\frac{d\vec{a}'}{dt} \;=\; -2(\vec{a}') \times [(\dot{q})q^{-1}] = 2[(\dot{q})q^{-1}] \times (\vec{a}') \qquad (147)$$

These last forms for $\frac{d\vec{a}'}{dt}$ can also be found in the literature (e.g., Baker 2023b).

Recall from Equation 116 that $\dot{q} = (1/2)\vec{w}q$, hence

$$(\dot{q})q^{-1} = (1/2)\vec{w} \tag{148}$$

Therefore, yet another expression for

$$\frac{d\vec{a}'}{dt} = -2(\vec{a}') \times [(\dot{q})q^{-1}] = 2[(\dot{q})q^{-1}] \times (\vec{a}') = \vec{w} \times \vec{a}' \tag{149}$$

With Equation 149, we have arrived at the promised alternative (in the paragraph preceding Equation 123) to using the DCM^{-1} or the q matrix for updating $\vec{a}'$. In this case the update is based on knowing the current $\vec{a}'$ and a knowledge of the current angular velocity $\vec{w}$. This expression can also be found in the literature for $\frac{d\vec{a}'}{dt}$ (e.g., Graf 2007; Narayan 2017, p. 5; Jia 2022, p. 14) and is in fact the classical vector algebra expression for the time derivative of a vector in terms of angular velocity of the vector about an axis of rotation, where both the vector and the axis of rotation are expressed in the fixed earth coordinate frame (e.g., Battin 1999). We have not specified anything about the quaternion, in terms of axis or angle, and yet we have arrived at an expression in Equation 149 that does not involve quaternions.

Of course, the update of $\vec{a}'$ will depend on iterative time integration, since both $\vec{w}$ (Equation 107) and $\vec{a}'$ (Equation 95) are based on nonlinear Euler-angle-dependent DCM^{-1} matrix equations.

We could also put Equation 149 in matrix form:

$$\frac{d\vec{a}'}{dt} = \begin{pmatrix} \dot{a}'_i \\ \dot{a}'_j \\ \dot{a}'_k \end{pmatrix} = \begin{pmatrix} 0 & -wk & wj \\ wk & 0 & -wi \\ -wj & wi & 0 \end{pmatrix} \begin{pmatrix} a'_i \\ a'_j \\ a'_k \end{pmatrix} \tag{150}$$

From a computational cost perspective, how many math operations are required to update $\vec{a}'$ using Equation 150?

Assuming we start with the current values of ψ, θ, ϕ, a'_i, a'_j, and a'_k, then Equation 47 reveals it will require nine math operations to get the current angular rates from the gyro rates. From the current angular rates, Equation 106 shows it will take 10 math operations to get the coefficients of the current angular velocity, *wi*, *wj*, and *wk*. With the current angular velocity and current vector coordinates, Equation 150 indicates it will take nine operations to get the current value of $\frac{d\vec{a}'}{dt}$. Updating $\vec{a}'(t + \Delta t)$ will then be done according to the following:

$$\vec{a}'(t + \Delta t) = \frac{d\vec{a}'}{dt}\Delta t + \vec{a}'(t) = \left(\vec{w} \times \vec{a}'(t)\right)\Delta t + \vec{a}'(t) \tag{151}$$

which will take another six operations (two for each vector component). In total, the sequence to update $\vec{a}'(t)$ to $\vec{a}'(t + \Delta t)$ will take 34 (= 9 + 10 + 9 + 6) math operations. But to set the stage for the next update, we must update the Euler angles as well, using their current angular rates, time interval, and current values, which takes six more operations. Hence, as summarized in Figure 7, the complete updating operation using Equation 151 will require 40 math operations. Clearly, 40 operations is less than the 48 needed using the DCM^{-1}, and the 75 needed with the q matrix (Figure 6); however, at this point in the analysis we are still left with the Euler angle singularity problem if $\theta = \pi/2$, because updating $\vec{w}$ using Equation 106 or 107 requires updating the Euler angle trig functions, which relies on the same singularity conditions (Equation 47) that were needed to update the DCM or DCM^{-1} matrix. In Section 8, we will show how the singularity problem can be dealt with.

40 math operations

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\sin\theta\cos\phi}{\boldsymbol{\cos\theta}} & \frac{\sin\theta\sin\phi}{\boldsymbol{\cos\theta}} & 1 \\ -\sin\phi & \cos\phi & 0 \\ \frac{\cos\phi}{\boldsymbol{\cos\theta}} & \frac{\sin\phi}{\boldsymbol{\cos\theta}} & 0 \end{bmatrix} \begin{bmatrix} R \\ Q \\ P \end{bmatrix}}_{9 \text{ math operations}}$$

$$\begin{pmatrix} wi \\ wj \\ wk \end{pmatrix} = \underbrace{\begin{pmatrix} cos(\theta)cos(\psi) & -sin(\psi) & 0 \\ cos(\theta)sin(\psi) & cos(\psi) & 0 \\ -sin(\theta) & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}}_{10 \text{ math operations}}$$

$$\frac{d\vec{a}'}{dt} = \vec{w} \times \vec{a}' = \underbrace{\begin{pmatrix} 0 & -wk & wj \\ wk & 0 & -wi \\ -wj & wi & 0 \end{pmatrix} \begin{pmatrix} a_i' \\ a_j' \\ a_k' \end{pmatrix}}_{9 \text{ math operations}}$$

$$\vec{a}'(t + \Delta t) = \underbrace{\frac{d\vec{a}'}{dt}\Delta t + \vec{a}'(t)}_{6 \text{ math operations}}$$

$$\underbrace{\begin{aligned} \phi(t + \Delta t) &= \dot{\phi}\,\Delta t + \phi(t) \\ \theta\,(t + \Delta t) &= \dot{\theta}\,\Delta t + \theta(t) \\ \psi\,(t + \Delta t) &= \dot{\psi}\,\Delta t + \psi(t) \end{aligned}}_{6 \text{ math operations}}$$

iterate on Δt

Figure 7. Math operations needed to update $\vec{a}'$ using $\frac{d\vec{a}'}{dt} = \vec{w} \times \vec{a}'$.

For completeness, a slightly different way to derive the relationship of Equation 148 ($\dot{q}q^{-1} = (1/2)\vec{w}$) is shown in Appendix B, while two other formulations for $\frac{d\vec{a}'}{dt}$ are shown in Appendix C, neither of which is computationally more efficient.

In summary, this section accomplished finding the quaternion-equivalent solution to the Euler-axes-based formulation (Equation 26) and evaluated how the two solutions compared; namely, the question raised in Equation 83 is proven true:

$$\vec{a}' = q(\phi,\theta,\psi)\vec{a}q(\phi,\theta,\psi)^{-1} \mapsto [\vec{a}']^{\mathrm{T}} = \mathrm{DCM}^{-1}(\phi, \theta, \psi)\,[\vec{a}]^{\mathrm{T}} \qquad (83)$$

The proof was obtained by deriving expressions for the quaternion coefficients q_o, q_i, q_j, and q_k as functions of the Euler half angle trig functions, expressed in Equations 91–94 (and in agreement with the literature), which, when substituted into the quaternion matrix (Equation 82) was found to produce the same DCM^{-1} matrix, thus validating Equation 83.

Nonetheless, we showed that using the quaternion matrix to solve for $\vec{a}'$ rather than using the DCM^{-1} matrix (Equation 26) is more computationally expensive. Even though that is the case, we pursued examining whether the quaternion approach/algebra had some advantage in being able to avoid the singularity condition present using vector algebra. To that end, we derived expressions for the angular velocities, $\overrightarrow{wi}$, $\overrightarrow{wj}$, and $\overrightarrow{wk}$ in the earth-fixed frame as functions of the body gyro rates (Equations 103–107). We acknowledged that to track $\vec{a}'$ in time would require updating the quaternion coefficients and derived expressions for the time derivatives of the quaternion coefficients $\dot{q}_o, \dot{q}_\iota, \dot{q}_J$ and $\dot{q}_k$ as function of the quaternion angular velocities (Equations 110–115, which were also consistent with the literature). We then derived the equivalent formulas as functions of the body angular velocities (Equations 116–121) and showed that, in this case, we can track the evolution of the quaternion coefficients starting from time zero without encountering and singularity problem (Equation 122). Lastly, we continued to pursue solution schemes and arrived at the traditional vector algebra formula for the time derivative of $\vec{a}'$ as a function of the current $\vec{a}'$ and the current angular velocity $\vec{w}$ (Equation 149 or 150). We found that this time-derivative approach to updating $\vec{a}'$ (Equation 151) is more efficient at face value than using the DCM^{-1}matrix of Equation 26, even though the angular singularity condition at $\theta = \pi/2$ persists. Finally, we stated, and will show later, that using Equation 151 will require roughly the same number of computations as needed to update $\vec{a}'$ using the DCM^{-1}, as we will have to renormalize $\vec{a}'$after each iteration.

The next section will examine yet another approach (a fourth, in addition to the DCM^{-1}, q-matrix, or $\vec{w} \times \vec{a}'$ strategies) for tracking a body's motion relative to an earth-fixed reference frame, which comes under the category of algorithm switching (e.g., Singla et al. 2005, p. 8; Kang and Park 2011). That is, we will pose the rotating body solution based on a different sequence of Euler angles; in particular, the set of angles about earth-fixed rotation axes, otherwise known as extrinsic rotation axes.

7.3 Multiple Axis Rotations (about Fixed/Extrinsic Quaternion Axes)

The question might be asked, can we avoid the Euler angle singularity problem if we formulate the quaternion q, distinguishing it as q', by defining it directly from rotations about earth-fixed axes, rather than the body axes of Section 7.2? This would merit calling it an extrinsic-axes-based vector analysis of body rotation.

Let the first quaternion rotation be about the k axis as before but denote the angle of rotation as ψ' for distinction. Hence, the quaternion representation of rotation about this axis will be

$$q'_{\psi'} = cos(\psi'/2) + sin(\psi'/2)\,k \tag{152}$$

Different from the intrinsic axes analysis of body rotation problem, we will define the second quaternion rotation about the j axis (not about the quaternion equivalent of the $eb2'$ axis, which had both i and j components [Equation 100]). We will designate the angle of rotation about j as θ', such that

$$q'_{\theta'} = cos(\theta'/2) + sin(\theta'/2)\,j \tag{153}$$

Lastly, the final rotation will be about the i axis by ϕ', such that

$$q'_{\phi'} = cos(\phi'/2) + sin(\phi'/2)\,i \tag{154}$$

If we quaternion multiply these three rotations and collect the scalar and vector components we obtain

$$q' = (q'_{\phi'})(q'_{\theta'})(q'_{\psi'}) = q'_o + \vec{q}' = q'_o + q'_i\,i + q'_j\,j + q'_k\,k \tag{155}$$

where

$$q'_o = \cos(\phi'/2)\cos(\theta'/2)\cos(\psi'/2) - \sin(\phi'/2)\sin(\psi'/2)\sin(\theta'/2) \tag{156}$$

$$q'_i = \sin(\phi'/2)\cos(\psi'/2)\cos(\theta'/2) + \cos(\phi'/2)\sin(\psi'/2)\sin(\theta'/2) \tag{157}$$

$$q'_j = \cos(\phi'/2)\sin(\theta'/2)\cos(\psi'/2) - \sin(\phi'/2)\cos(\theta'/2)\sin(\psi'/2) \tag{158}$$

$$q'_k = \cos(\phi'/2)\cos(\theta'/2)\sin(\psi'/2) + \sin(\phi'/2)\cos(\psi'/2)\sin(\theta'/2) \tag{159}$$

Although the angles and axes are different than the previous quaternion (intrinsic Euler axes) representations of the body rotations, there is a similarity in the trig function groupings (albeit a difference in the signs joining the three-tuple trig functions) compared with Equations 91–94. This extrinsic form of the q matrix coefficients can also be found in the literature (Henderson 1977, p. 10).

Nesting this q' and its inverse around a vector $\vec{a}$ will yield an expression for a rotated body vector $\vec{a}'$:

$$\vec{a}' = (q'\ \vec{a})\ q'^{-1} \tag{160}$$

This can be put in matrix form as well, producing the same formulation as Equation 82, even though the q matrix coefficients are slightly different (Equations 156–159 vs. Equations 91–94):

$$\begin{bmatrix} a_i' \\ a_j' \\ a_k' \end{bmatrix} = \begin{bmatrix} (q_o'^2 + q_i'^2 - q_j'^2 - q_k'^2) & 2(q_j'q_i' - q_o'q_k') & 2(q_o'q_j' + q_k'q_i') \\ 2(q_o'q_k' + q_j'q_i') & (q_o'^2 + q_j'^2 - q_i'^2 - q_k'^2) & 2(q_k'q_j' - q_o'q_i') \\ 2(q_k'q_i' - q_o'q_j') & 2(q_o'q_i' + q_j'q_k') & (q_o'^2 + q_k'^2 - q_i'^2 - q_j'^2) \end{bmatrix} \begin{bmatrix} a_i \\ a_j \\ a_k \end{bmatrix} \tag{161}$$

If we substitute Equations 156–159 into Equation 161, consolidate and collect terms using trig identities, we arrive at

$$\vec{a}' = (q'\ \vec{a})\ q'^{-1} =$$

$$\begin{pmatrix} a_i' \\ a_j' \\ a_k' \end{pmatrix} = \begin{pmatrix} c_{\theta'}c_{\psi'} & -c_{\theta'}s_{\psi'} & s_{\theta'} \\ s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'} & c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'} & -c_{\theta'}s_{\phi'} \\ s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'} & c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'} & c_{\theta'}c_{\phi'} \end{pmatrix} \begin{pmatrix} a_i \\ a_j \\ a_k \end{pmatrix} \tag{162}$$

At first glance, this matrix looks similar to the DCM(ϕ, θ, ψ) of Equation 10, but the definition of the axis/angles are different, as are some of the sign multipliers. Nonetheless, since the vector $\vec{a}$ still represents a fixed vector in a rotating body, the matrix of Equation 162 could/should/can be designated as DCM$'^{-1}$, representing the inverse-direction cosine matrix for describing body rotations about the extrinsic earth-fixed axes. Since the number of trig function combinations are the same as the DCM or DCM^{-1}, the minimum number of computations (multiplications, adds, or subtracts) needed to update $\vec{a}'$ using the DCM$'^{-1}$ of Equation 162 remains at 33.

What are some of the physical takeaways from this DCM$'^{-1}$ matrix? First, we see from the far right column that if our starting vector $\vec{a}$ has an a_k component, then the projection of this a_k component onto any of the earth/quaternion axes will not be affected by any ψ' rotation. This was not the case for rotations about the Euler axes. In addition, we see from the first row of the matrix that a change in the projection of the (rotated $\vec{a}$ into) $\vec{a}'$ vector onto the i axis will not be influenced by any ϕ' rotation about the i axis. This was also not the case for rotations about the Euler axes.

As a sanity check, note that if $\phi' = \theta' = 0$, in the DCM$'^{-1}$ (ϕ', θ', and ψ') matrix above, then

$$\vec{a}_i' = (cos(\psi')\, a_i \; - \sin(\psi')\, a_j)\, i \tag{163}$$

$$\vec{a}_j' = (\sin(\psi')\, a_i \; + cos(\psi')\, a_j)\, j \tag{164}$$

$$\vec{a}_k' = a_k\, k \tag{165}$$

which is the expected outcome after a stationary vector $\vec{a}$ in the body's reference frame is rotated by ψ' around the ee3 axis, and then that rotated body vector is projected onto the fixed earth axes.

Figure 8 shows how the body axes relate to the quaternion axes through the angles ϕ', θ', and ψ'. We could stepwise-analyze the projections of each body unit vector onto the i, j, and k axes after ψ' then θ' and finally ϕ', or we could use the matrix Equation 162 by evaluating how a unit vector $\vec{a} = i = eb1$ (i.e., set $a_i = 1, a_j = 0$, and $a_k = 0$) is transformed by the rotation. Doing so, we arrive at *eb1* after rotation (as projected onto the earth axes):

$$eb1 = c_{\theta'} c_{\psi'}\, i + [\, s_{\phi'} s_{\theta'} c_{\psi'} + s_{\psi'} c_{\phi'} \,] j + [\, s_{\psi'} s_{\phi'} - c_{\phi'} s_{\theta'} c_{\psi'} \,] k \tag{166}$$

Similarly, setting $\vec{a} = j = eb2$ in Equation 162 reveals the following after rotation:

$$eb2 = -c_{\theta'} s_{\psi'}\, i + [\, c_{\psi'} c_{\phi'} - s_{\phi'} s_{\theta'} s_{\psi'} \,] j + [\, c_{\phi'} s_{\theta'} s_{\psi'} + c_{\psi'} s_{\phi'} \,] k \tag{167}$$

and likewise setting $\vec{a} = k = eb3$ in Equation 162 produces the following:

$$eb3 = s_{\theta'}\, i \; - c_{\theta'} s_{\phi'}\, j + c_{\theta'} c_{\phi'}\, k \tag{168}$$

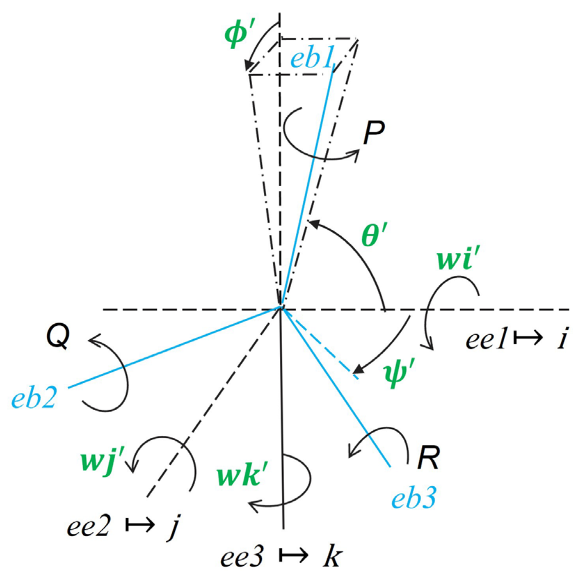


Figure 8. Geometry of angular rates relative to body and earth axes.

Writing Equations 166–168 as a matrix gives us the following:

$$\begin{pmatrix} eb1 \\ eb2 \\ eb3 \end{pmatrix} = \begin{pmatrix} c_{\theta'}c_{\psi'} & s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'} & s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'} \\ -c_{\theta'}s_{\psi'} & c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'} & c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'} \\ s_{\theta'} & -c_{\theta'}s_{\phi'} & c_{\theta'}c_{\phi'} \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} \quad (169)$$

This would correspond to the "active" rotation interpretation (as defined, after Equation 13, for DCM(ϕ, θ, ψ)), and is in fact the inverse of $\mathrm{DCM}'^{-1}(\phi', \theta', \psi')$, thus we define the above 3 × 3 matrix as $\mathrm{DCM}'(\phi', \theta', \psi')$.

Equations 31–33 are reproduced below. They set the requirement that the three body axes remain perpendicular to each other by specifying the time rate of change of each body unit vector could only change in the directions perpendicular to itself. Those conditions remain:

$$e\dot{b}1 = R\,eb2 - Q\,eb3 \quad (31)$$

$$e\dot{b}2 = -R\,eb1 + P\,eb3 \quad (32)$$

$$e\dot{b}3 = -P\,eb2 + Q\,eb1 \quad (33)$$

Time-differentiating Equations 166–168 yields expressions for the left of Equations 31–33, then substituting Equations 166–168 into the right of Equations 31–33, grouping and rearranging terms produces:

$$\begin{pmatrix} \dot{\phi}' \\ \dot{\theta}' \\ \dot{\psi}' \end{pmatrix} = \begin{pmatrix} 0 & -s_{\psi'}/c_{\theta'} & c_{\psi'}/c_{\theta'} \\ 0 & c_{\psi'} & s_{\psi'} \\ 1 & s_{\theta'}s_{\psi'}/c_{\theta'} & -s_{\theta'}c_{\psi'}/c_{\theta'} \end{pmatrix} \begin{pmatrix} R \\ Q \\ P \end{pmatrix} \quad (170)$$

If we compare the above matrix for $\dot{\phi}'$, $\dot{\theta}'$, and $\dot{\psi}'$ (or wi', wj', and wk') with that of Equation 47 for $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$, we see there would still be a singularity (division by zero) at $\theta' = \pi/2$.

Hence, even though we are defining quaternion rotations about always-orthogonal (nontraditional Euler angle, i.e., extrinsic) axes, we still arrive at a condition where rotation rates would go to infinity if the θ' angle ever reaches $\pi/2$.

Nonetheless, we can still proceed to find the corresponding expression for angular velocity given that we know

$$\vec{w}' = \overrightarrow{wi}' + \overrightarrow{wj}' + \overrightarrow{wk}' = wi'i + wj'j + wk'\mathrm{k} = P\,eb1 + Q\,eb2 + R\,eb3 \quad (171)$$

we can use Equations 166–168 to substitute for *eb1*, *eb2*, and *eb3* in Equation 171,

$$\vec{w}' = P\,eb1 + Q\,eb2 + R\,eb3$$
$$= P\,(c_{\theta'}c_{\psi'}\,i + [\,s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'}\,]\,j + [\,s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'}\,]\,k)$$
$$+ Q\,(-c_{\theta'}s_{\psi'}\,i + [\,c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'}\,]\,j + [\,c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'}\,]\,k)$$

$$+ R\,(s_{\theta'}\,i\;-c_{\theta'}s_{\phi'}\,j + c_{\theta'}c_{\phi'}\,k) \tag{172}$$

Collecting the coefficients of i, j, and k in Equation 172, we get

$$\vec{w}' = \{\,P\,c_{\theta'}c_{\psi'} - Q\,c_{\theta'}s_{\psi'}\;+\;R\,s_{\theta'}\,\}\,i$$

$$+\{\,P\,[\,s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'}\,] + Q\,[\,c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'}\,] - R\,c_{\theta'}s_{\phi'}\,\}\,j$$

$$+\{\,P\,[\,s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'}\,]\;+Q\,[\,c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'}\,] + R\,c_{\theta'}c_{\phi'}\,\}\,k \tag{173}$$

Comparing Equation 171 with 173, we can make the following identification for the components of $\vec{w}'$:

$$wi' =\;P\,c_{\theta'}c_{\psi'} - Q\,c_{\theta'}s_{\psi'} + R\,s_{\theta'} \tag{174}$$

$$wj' =\;P\,[\,s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'}\,] + Q\,[\,c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'}\,] - R\,c_{\theta'}s_{\phi'} \tag{175}$$

$$wk' = P\,[\,s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'}\,]\;+Q\,[\,c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'}\,] + R\,c_{\theta'}c_{\phi'} \tag{176}$$

Or in matrix form:

$$\begin{pmatrix} wi' \\ wj' \\ wk' \end{pmatrix} = \begin{pmatrix} c_{\theta'}c_{\psi'} & -c_{\theta'}s_{\psi'} & s_{\theta'} \\ s_{\phi'}s_{\theta'}c_{\psi'} + s_{\psi'}c_{\phi'} & c_{\psi'}c_{\phi'} - s_{\phi'}s_{\theta'}s_{\psi'} & -c_{\theta'}s_{\phi'} \\ s_{\psi'}s_{\phi'} - c_{\phi'}s_{\theta'}c_{\psi'} & c_{\phi'}s_{\theta'}s_{\psi'} + c_{\psi'}s_{\phi'} & c_{\theta'}c_{\phi'} \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

$$= \mathrm{DCM}'^{-1}\,[P,Q,R]^{\mathrm{T}} \tag{177}$$

In summary to this point, we found singularity conditions still exist, whether using quaternion-based rotational axes representing the intrinsic Euler axes and angles ψ, θ, and ϕ, or quaternion-based rotational axes and angles, ψ', θ', and ϕ' about the fixed earth frame. That said, we will show in the next section how (algorithm) switching from intrinsic to extrinsic formulations, or vice versa, could be used (and has been used by others in the literature) to avoid the singularity condition when θ or θ' approaches $\pi/2$. However, a question yet to be asked is: will the middle/pitch angle θ or θ' ever reach the singularity-causing value of $\pi/2$?

8. Avoiding Euler Angle Singularities when Solving the 3D Body Rotation Problem, Regardless of the Algorithm Sequence

A philosophical argument can be made that the pitch angle θ or θ' cannot reach $\pi/2$ because then $\dot{\psi}$ or $\dot{\psi}'$ and $\dot{\phi}$ or $\dot{\phi}'$ would become infinite, or non-time-differentiable, and it is known that smooth functions of time, such as ψ or ψ', θ or θ', and ϕ or ϕ' (functions that do not have sharp changes in their independent variables [corners, breaks, jumps]) are time-continuous and differentiable. Physically, this means that we do not expect ψ or ψ', θ or θ', nor ϕ or ϕ' to suddenly jump or change within an infinitely small period of time. But how can we reconcile this expectation with the appearance of the singularity condition in Equations 47 and 170?

To review, with the exception of using the quaternion derivative method of Equations 82, 121, 122, and Figure 6 to track the rotation of a body vector $\vec{a}$ in time, the other three methods—1) the moving Euler axes in the DCM^{-1} of Equation 26 or 95; 2) updating $\vec{a}'$ based on its time-rate of change using Equation 151; and 3) using the earth-fixed axes in the DCM$'^{-1}$ of Equation 162—all need to update the Euler angles that track the body's frame of reference to track the rotation of a body vector $\vec{a}$ in time. These Euler angles have a time rate of change that depends (nonlinearly) on the body's gyro rates about each of the body axes, as given by Equation 47 (if using the Euler axes to track the rotation), or as given in Equation 168 (if using the earth-fixed axes). Using either reference frame for tracking, the relationship between the angular rates in those frames and the (theoretically measurable) gyro rates involves division by the cosine of either θ or θ', which would thus make those angular rates approach infinity as θ or θ' approach $\pi/2$. So, to reiterate the question posed at the end of the last section: will these angles ever reach $\pi/2$?

To answer such a question, let us look at how the angles are updated. Since both the Euler (intrinsic) angles, ψ, θ, and ϕ, and the fixed axes (extrinsic) angles, ψ', θ', and ϕ', will be updated the same way, we will choose to show, for illustrative purposes, the intrinsic Euler angle update process. The values of ϕ, θ, and ψ can be updated by iteratively summing the angular rates in time steps of Δt, as shown below:

$$\phi(t) = \phi(t{=}0) + \sum \dot{\phi}\,\Delta t = \sum \dot{\phi}\,\Delta t \tag{178}$$

$$\theta(t) = \theta\,(t{=}0) + \sum \dot{\theta}\,\Delta t = \sum \dot{\theta}\,\Delta t \tag{179}$$

$$\psi(t) = \psi\,(t{=}0) + \sum \dot{\psi}\,\Delta t = \sum \dot{\psi}\,\Delta t \tag{180}$$

where all time-zero angular values are zero by definition (i.e., the body axes start in alignment with the earth axes).

We are free to make Δt anything we choose; however, the magnitude of Δt will affect both the accuracy and the computational efficiency of the solution, as the angular rates in Equations 178–180 (or the vector velocity of Equation 151) are computed and held constant over Δt based on the matrix evaluation of Equation 47 at each iteration in the summation. Since the values of R, Q, and P, affect the magnitude of the angular rates, it would be sensible (from an accuracy and efficiency perspective) to normalize the time step by making it inversely proportional to the square root of the sum of the squares of the gyro rates. Thus, if any gyro rate is very high, we could limit the loss in accuracy in our iterative solution by using a proportionately smaller time step; while if all gyro rates were very small, we could increase solution efficiency with reduced loss of accuracy by taking a larger time step. However, in numerical simulations with gyro rates less than unity, it was found that dividing by the square root of the sum of their squares could lead to rather large angular changes per iteration. Hence, a better normalization for small angular rates was to add $+1$ to the sum of the gyro rates squared. (For gyro rates on the order of unity or higher, retaining the addition of $+1$ to the sum of their squares will not have a significant effect step size per iteration.) In addition to the magnitude of the gyro rates themselves, which for the immediate discussion are assumed constant, division by the cosine of the pitch angle θ in Equation 47 will dominate the angular rate equations when/if the pitch angle approaches $\pi/2$. Thus, we should decrease the iteration time step by the $|cos(\theta)|$ to offset/negate/cancel this increase in angular rates as θ approaches $\pi/2$. Using this rational, we arrive at the following sensible time step:

$$\Delta t = (|cos(\theta)||)/\sqrt{1 + R^2 + Q^2 + P^2} \tag{181}$$

Thus, for the Euler-angle update Equations 178–180, using Equations 47 and 181, we would be solving the following:

$$\phi(t) = \sum \dot{\phi}\ \Delta t$$

$$= \sum\ (sin(\theta)\left\{R\frac{cos(\phi)}{\cos(\theta)} + Q\frac{sin(\phi)}{\cos(\theta)}\right\} +\ \dot{P})|cos(\theta)|/\sqrt{1 + R^2 + Q^2 + P^2}$$

$$= \sum\ (sin(\theta)sign(cos(\theta))\{Rcos(\phi) + Q\dot{s}in(\phi)\} + P|\cos(\theta)|)/\sqrt{1 + R^2 + Q^2 + P^2} \tag{182}$$

$$\theta(t) = \sum \dot{\theta}\ \Delta t = \sum\ (-Rsin(\phi) + Qcos(\phi))|\dot{c}os(\theta)|/\sqrt{1 + R^2 + Q^2 + P^2} \tag{183}$$

$$\psi(t) = \sum \dot{\psi}\ \Delta t = \sum\ sign(cos(\theta))(Rcos(\phi) + \dot{Q}sin(\phi))/\sqrt{1 + R^2 + Q^2 + P^2} \tag{184}$$

where $sign(cos(\theta)) = cos(\theta)/|cos(\theta)|$ is a function that returns either $+$ or $-$ depending on the sign of the $cos(\theta)$.

Plainly, there is no division by the $cos(\theta)$ in the angular solutions of Equations 182–184, so the singularity condition is avoided by the choice of Equation 181 for Δt.

Hence, the above iterative solutions will not "blow up" due to any division by zero terms, but will θ have a maximum value, such that $\dot{\theta} = 0$? As long as R and P are not both zero (i.e., the nontrivial case, where the body is simply rotating continuously in the pitch plane), the answer is yes, $\dot{\theta}$ will $= 0$ at some orientation, as reasoned below.

From Equation 47,

$$\dot{\theta} = -Rsin(\phi) + Qcos(\phi) \tag{185}$$

which is = 0 when

$$tan(\phi) = Q/R, \text{ or when } \phi = tan^{-1}(Q/R) \tag{186}$$

For special cases where $R = 0$ and Q is nonzero, then $\dot{\theta} = 0$ when $\phi = \pm\pi/2$ (or $\pm\pi$ multiples thereof), and since the only non-trivial case is when *R* and *P* are not both 0, ϕ must move off its initial value of 0 for any nonzero *P*. Or, if $Q = 0$ and *R* is nonzero, then $\dot{\theta} = 0$ in Equation 186, when $\phi = 0$ (or $\pm\pi$ multiples thereof).

If both Q and $R = 0$, then $\dot{\theta} = 0$ for all time in Equation 185, which means θ stays at its initial value of 0. The other extreme would be if both *P* and $R = 0$ with $Q \neq 0$, then as we indicated, the problem would be physically describing the trivial case of a body that rotates continuously in the pitch plane only.

We illustrated the above conditions for the Euler axes case: if we were using the fixed-frame quaternion rotational axes, ψ', θ', or ϕ' (about the earth *z*, *y*, *x* axes, respectively) the equivalent iterative nonsingular solution would be the following:

$$\Delta t = (|cos(\theta')||)/\sqrt{1 + R^2 + Q^2 + P^2} \tag{187}$$

$$\phi'(t) = \sum \dot{\phi}'\Delta t = \sum\ sign(cos(\theta'))(Pcos(\psi') - Q\dot{s}in(\psi'))/\sqrt{1 + R^2 + Q^2 + P^2} \tag{188}$$

$$\theta'(t) = \sum \dot{\theta}'\Delta t = \sum\ (Psin\left(\psi'\right) + Q\dot{c}os\left(\psi'\right))\left|cos(\theta')\right|/\sqrt{1 + R^2 + Q^2 + P^2} \tag{189}$$

$$\psi'(t) = \sum \dot{\psi}'\Delta t$$

$$= \sum\ (sin(\theta')sign(cos(\theta'))\{Qsin(\phi') - P\dot{c}os(\phi')\} + R|\cos(\theta')|)/\sqrt{1 + R^2 + Q^2 + P^2} \tag{190}$$

From Equation 189, $\dot{\theta}' = 0$ when

$$\dot{\theta}' = Psin\left(\psi'\right) + Qcos\left(\psi'\right) = 0 \tag{191}$$

or, when

$$tan(\psi') = -\,Q/P, \text{ or when } \psi' = -\,tan^{-1}(Q/P) \tag{192}$$

If $P = 0$, then $\dot{\theta}' = 0$ when $\psi' = \pm\ \pi/2$ (or $\pm\pi$ multiples thereof); if $Q = 0$, then $\dot{\theta}' = 0$ when $\psi' = 0$ (or $\pm\pi$ multiples thereof).

However, we cannot know a priori what the maximum value of θ or θ' will be without solving the above set of nonlinear equations for any particular values of *R*, *Q*, and *P*. That said, there are some instructive observations that can be made by showing example solutions that span a range of *R*:*Q*:*P* ratios.

Since showing all four solution schemes (i.e., the intrinsic DCM^{-1} solution of Equation 26 or 95; the nonsingular q-matrix solution of Equations 121–122 in Equation 82; the $\vec{w} \times \vec{a}'$ solution of Equation 151; and the extrinsic DCM'^{-1} solution of Equation 162) would produce cluttered comparison plots, we choose to show pairwise comparisons between the four different solution methodologies.

The first comparison plot will showcase the intrinsic DCM^{-1} solution versus the extrinsic DCM'^{-1} solution. It was hypothesized in Equations 182–184 and Equations 188–190 that body rotations can be tracked, for any gyro sensor rates, without incurring the singularity condition, by using the time increment defined by Equation 181 for updating ψ, θ, or ϕ, and by using the time increment defined by Equation 187 for updating ψ', θ', or ϕ'. Moreover, it is expected that both algorithms will yield the same prediction for the 3D body rotation.

Let us assume, for example, that reasonable gyro rates vary from ± 1 °/s down to ± 0.01 °/s (~± 0.0174 to ± 0.000174 rad/s) over a 100-fold range, though only the ratio of the gyro magnitudes are of relevance to the simulations that follow.

Suppose we start with all gyro rates being relatively small, having a value of 0.01 °/s. Figure 9 shows the angular change for these gyro rates using the iteration formulations above. For abbreviation purposes, the Euler-axis-based quaternion formulation (ψ, θ, and ϕ solutions; recall this solution was shown to be the same using 4D quaternions or traditional 3D vector algebra) are associated with the "Euler-Quat" designation in the Figure 9 legend, and the "fixed Quat" legend-labels refer to rotations about the fixed-frame/earth axes (i.e., the ψ', θ', or ϕ' solutions). Furthermore, terminology-wise, we will use the labels yaw, pitch, and roll to designate both ψ and ψ', θ and θ', and ϕ and ϕ', respectively.

Three observations can be made. First, as postulated, there is a maximum value for θ and θ', which for the first cycle is −89.9864° and 89.9975°, respectively (but occurring at different iteration steps, exhibiting the practicality/usefulness of algorithm switching in solutions schemes (such as Singla et al., p. 8; Kang and Park) that are not using a pitch angle-dependent time step [as being done here, to avoid the singularity problem]). Second, these values are close to, but less than, $\pm\pi/2$. Third, there are 13 completed cycles within this 1,000,000 iteration calculation, with little change in the angular pattern. For example, the maximum θ and θ' at the 13 cycle are −89.7808° and 89.8071°, respectively (the slight difference between first cycle values and subsequent cycles will be discussed later in this Section).

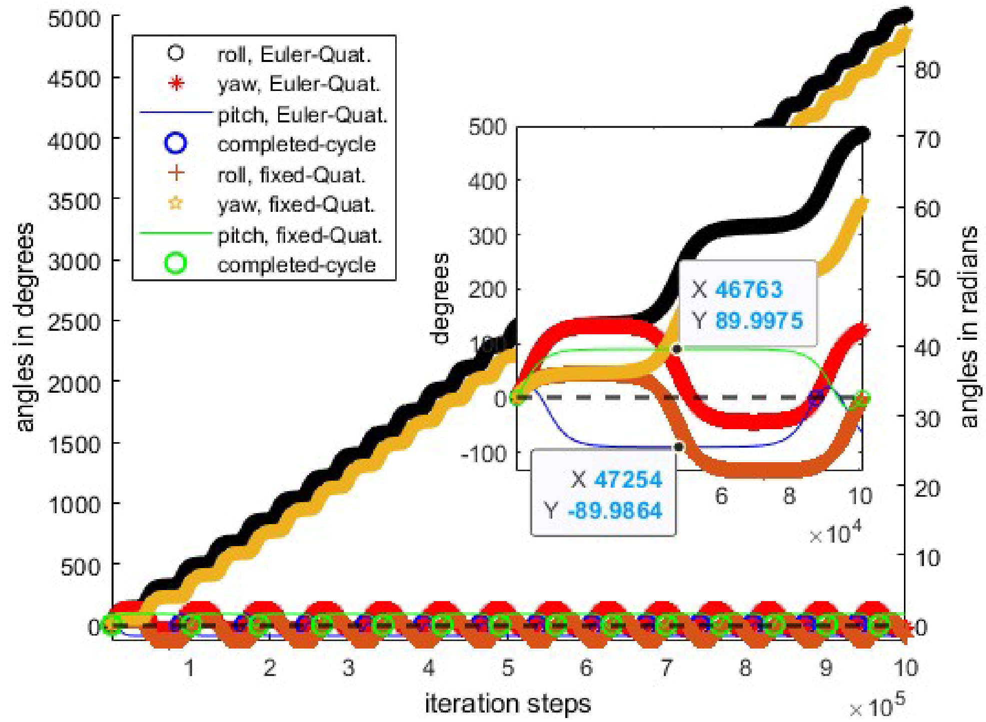


Figure 9. **Iterative angular solutions of the nonlinear, Euler-axes-based and fixed-axes-based kinematic equations of motion for relatively slow, but equal, body gyro rates of 0.01 °/s, using the variable time steps prescription of Equations 181 and 187, respectively.**

It might be questioned: are these two different solution methodologies tracking the same body rotation? That is, recall from the earlier discussions (e.g., Equation 10) that the Euler-Quat solution tracks the motion of the body axes (Equations 34–36) or a moving body vector (e.g., Equation 26 or 95) using the 3D vector or 4D quaternion formulation, respectively, for Euler rotation angles (ψ, θ, and ϕ) about the (intrinsic) Euler axes (*eb3*=*ee3*, *eb2′*, and *eb1″*). While the fixed-Quat solution tracks the body axes motion (Equations 166–168) using quaternions that represent rotation angles (ψ', θ', and ϕ') about the (extrinsic) fixed-earth-frame axes (*ee3*, *ee2*, *ee1*, which map directly to the quaternion axes *k*, *j*, *i*, respectively). Figure 10 is a 3D plot of both solutions (Equations 10 and 169) for the body axes trajectories (in the legend *eb1*$\stackrel{\text{def}}{=}$body-x, *eb2*$\stackrel{\text{def}}{=}$body-y, and *eb3*$\stackrel{\text{def}}{=}$body-z) for the first 70,000 iterations (roughly ¾ of the solution shown in the inset) of Figure 9 (note, the *x*, *y*, and *z* plot axes correspond to the quaternion *i*, *j*, and *k* axes, or equivalently, the earth's *ee1*, *ee2*, and *ee3* axes).

Three observations from Equation 10 can be made. First, although the maximum in the Euler-Quat solution is near $-\pi/2$, while the maximum in the fixed-Quat solution is near a positive $+\pi/2$, the two different approaches are tracking the same body motion, as expected (here each solution is plotted every 150 iteration steps, though the two solutions for the trajectories of the body axes are offset from each other by 25 iteration steps, in order to avoid visual overlap). Second, the density of plots increases as each θ or θ' solution approaches $\pm\pi/2$, since the iteration step size (which is also relatable to the time) decreases at a rate proportional to $\left|\cos\left(\theta \text{ or } \theta'\right)\right|$, which happens at different iterations for each solution (again, associated with the rationale for using algorithm switching). Third, the axis of rotation stays constant (pointing in the same direction as it starts) for both solutions, in this example (for all-equal gyro rates) the rotation axis direction is 0.01 $(i + j + k)$ °/s (but scaled up by 2500× in length and 6× in line-width, to be seen in the plot).

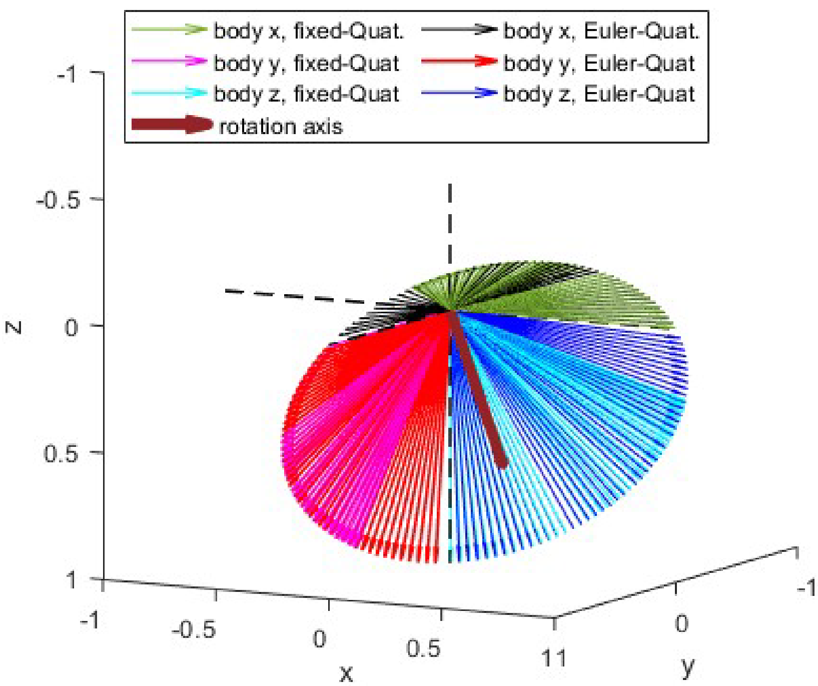


Figure 10. 3D plot of the unit-magnitude body axes (*x*, *y*, and *z*, or *eb1*, *eb2* and *eb3*) trajectories for the first 70,000 iterations (roughly ¾ of the solution shown in the inset) of Figure 9.

Analytically, the axis of rotation can be found using either: 1) the Euler-Quat equation for $\vec{w}$ (Equation 107); 2) Equation 108 for $\overrightarrow{wb}$; or 3) the fixed-Quat equation for $\vec{w}'$ (Equation 177); or simply recognizing that

$$\vec{w} = \overrightarrow{wb} = \vec{w}' = Peb1 + Qeb2 + Reb3 \tag{193}$$

Since the body axes are all orthogonal, taking the dot product of Equation 193 with itself would show that

$$|\vec{w}| = |\vec{w}'| = \sqrt{(Peb1 + Qeb2 + Reb3) \cdot (Peb1 + Qeb2 + Reb3)}$$
$$= \sqrt{R^2 + Q^2 + P^2} \tag{194}$$

Numerically, the axis of rotation for each iteration in Figure 10 is computed by evaluating the expression given in Equation 193, using the solved-for and plotted Euler-Quat and fixed-Quat solutions for *eb1*, *eb2*, and *eb3* (Equations 34–36 and 166–168, respectively).

Figure 11 is analogous to Figure 9, but is at the other extreme of our chosen range of simulation examples, where all gyro rates are 100× faster at 1 °/s. In comparison (Figure 11 to Figure 9), angles and cycles are reached with fewer iterations, as expected. For example, there are 29 pitch-angle cycles completed in 10,000 iterations shown in Figure 11, versus 13 cycles completed in 1,000,000 iterations in Figure 9. Nonetheless, all maximum values of θ or θ' remain close to, but less than, $\pm\pi/2$, using the time step size computed for each iteration as given by Equations 181 and 187.

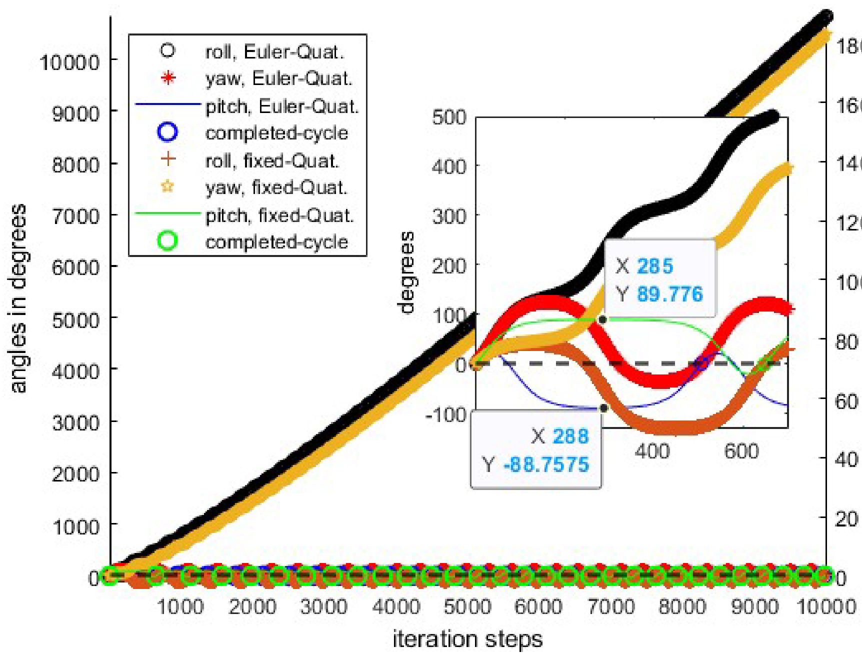


Figure 11. Iterative angular solutions of the nonlinear, Euler-axes-based and fixed-axes-based equations of kinematic motion for relatively fast, but equal, body gyro rates of 1 °/s using the variable time steps prescription of Equations 181 and 187, respectively.

What about a case where one gyro rotation rate is significantly greater than the others? For example, suppose we take the case where the gyro rate for roll and yaw are both slow at 0.01 °/s, but the gyro rate for pitch is much faster at 1 °/s. Figure 12 plots the results. It can be seen there is nothing strikingly different in comparison with the previous two angular output results (Figures 9 and 11) other than 13 pitch cycles completed in 20,000 iterations, versus the same number of cycles taking 1,000,000 iterations when all the gyro rates are as slow as 0.01 °/s.

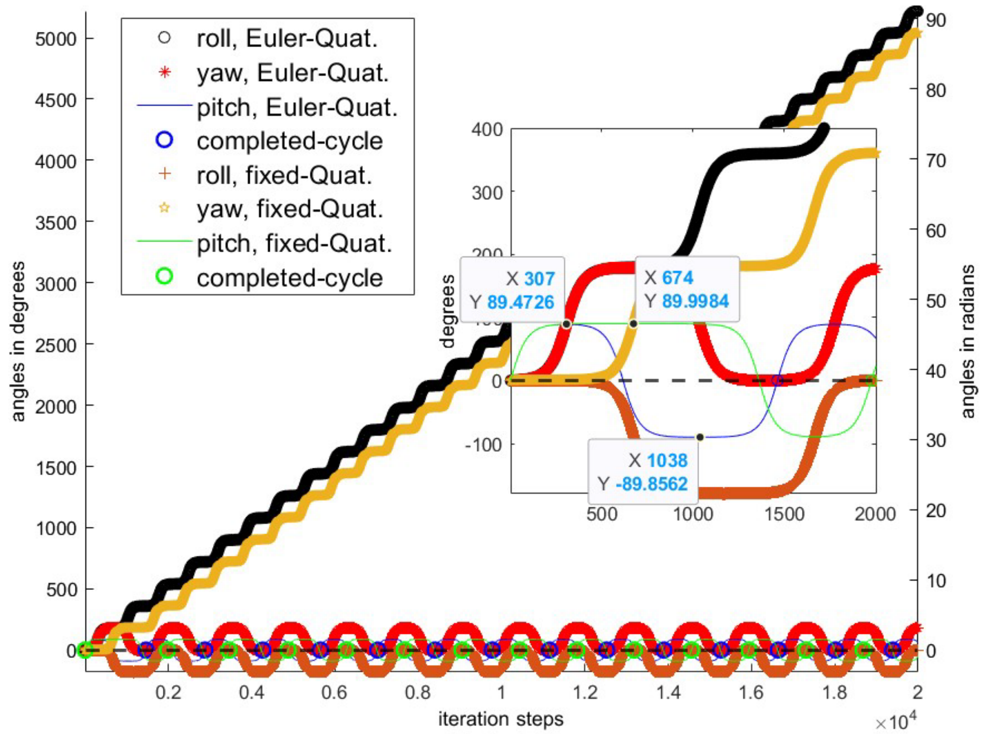


Figure 12. Angular solutions for relatively slow gyro roll and yaw rates of 0.01 °/s, but relatively fast gyro pitch rate of 1 °/s.

For comparison with the 3D plot of the body axes in Figure 10, Figure 13 shows the path of the body axes for the first 1300 iterations of Figure 12 (which captures most of the Euler-Quat, and much of the fixed-Quat first rotation cycle). We can see in Figure 13 that all the body axes begin aligned with the earth-fixed frame, but they will thereafter begin to follow a path that brings them into alignment with a cone-shaped pattern around an axis of rotation. The cone pattern is most clearly seen in Figure 10 but is also present in Figure 13, where the cone (or opening, or slant) angle is nearly equal to $\pi/2$ for the body x and z axes and almost 0° for the body y axes. This difference in cone angles is due, in this case, to the fact that the pitch gyro rate (Q) about $eb2$ (or body-y) is large, while the roll and yaw gyro rates (P and R) around $eb1$ and $eb3$, respectively, are small. Whereas in Figure 10, all the gyro rates were the same, and thus the cone angles are all the same. Under these conditions, Equation 177 for $\vec{w}' = \vec{w} = \overrightarrow{wb} = Peb1 + Qeb2 + Reb3$, would predict, as indicated in Figure 13, that the rotation axes should be nearly aligned with the body's $eb2$ axis. Nevertheless, the small positive values of the roll and yaw gyros would predict that the rotation axis is slightly down and to the right of $eb2$, which itself is nearly aligned with the earth's y axes. The tilt of the rotation axis and wide cone angle for $eb1$ and $eb3$ will lean their conic surfaces slightly away from the $-ee3$ and $-ee1$ axis but pass them close to the $+ee3$ and $+ee1$ axis. To show this, we can look for example at the specific iteration values of Figure 12 where the body's *eb1* axis (black-colored vector for Euler-Quat solution, and olive-colored vector for fixed-Quat solution) comes closest to the earth's $-ee3$ and the pitch angles are as close to $+\pi/2$ as they are going to get in the first cycle. As noted in Figure 12, this is at the 307th iteration for the Euler-Quat solution and the 674th for the fixed-Quat. If we zoom in on these iteration values for *eb1*, *eb2*, and *eb3* and plot them separately in Figure 14, we can see that *eb1* is tilted away from $-ee3$ but *eb3* is almost aligned with $+ee1$. A second observation that can be made from Figure 13, which was also stated and observed for Figure 10, is that the two solutions for the body axes occupy the same space (lie on the same conic surfaces). This is clearly seen in the magnified solution of Figure 14, where even though the iteration numbers are quite different (307 vs. 674) the body axes point in virtually the same directions. Since there is a time scale associated with each iteration, whose magnitude would be proportional to $\left|\cos\left(\theta \text{ or } \theta'\right)\right|$, a qualitative argument can be made (without proof here) that the total time elapsed to reach the 307 Euler-Quat iteration solution will be the same time as taken to reach the 674 fixed-Quat solution, since the time increments for the fixed-Quat summation will remain proportional to approximately cos(89.9984°) = 2×10^{-5} per iteration for the 300+ iterations from 307 to 674.

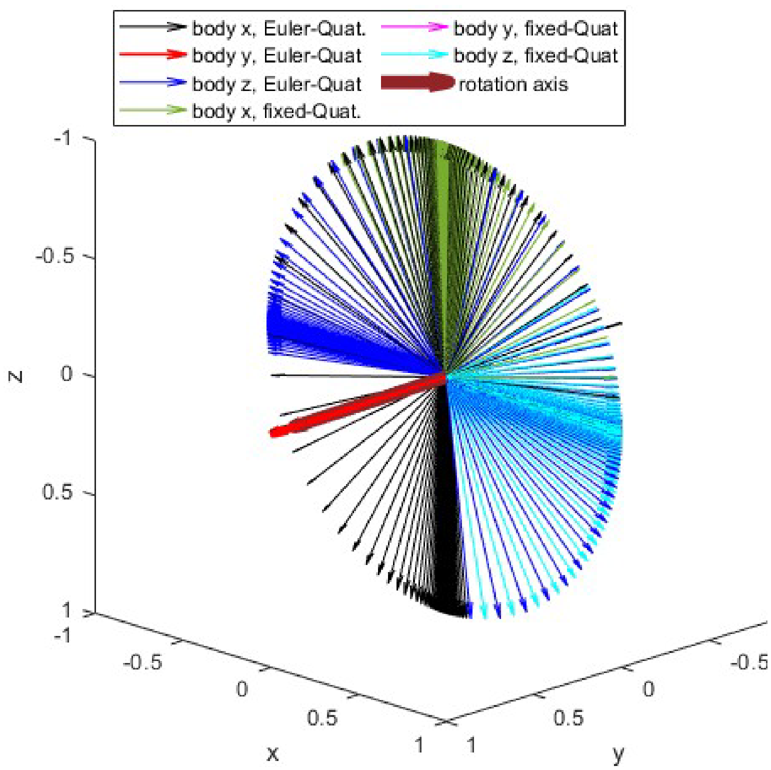


Figure 13. 3D plot of the unit-magnitude body axes (*x*, *y*, and *z* or *eb1*, *eb2*, and *eb3*) trajectories for the first 1300 iterations of Figure 12.

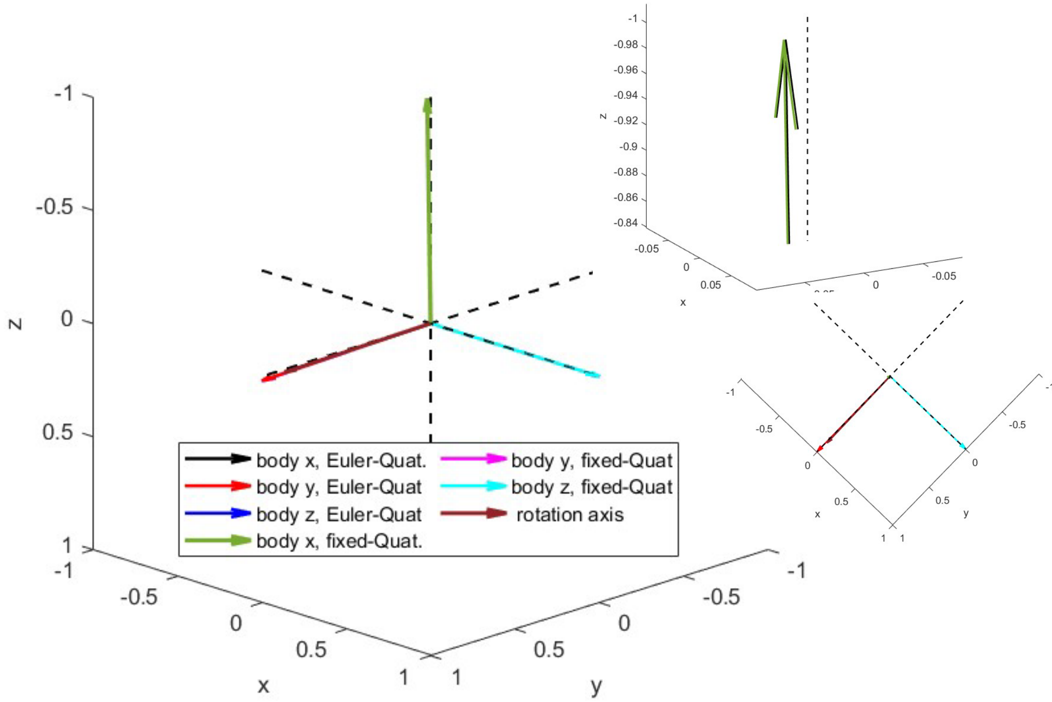


Figure 14. Magnification of body axes locations of Figure 13 at the iteration number for the passage of *eb1* closest to the $-z$ $(-ee3)$ and pitch angles closest to $+\pi/2$ within the first cycle.

Lastly, as was presented in the discussion of Figure 9, there is a slight difference in the maximum values of θ and θ' from the first cycle to later cycles. Having now discussed how the body axes transition from laying on perpendicular planes to lying on conic surfaces, the slight change in maximum pitch angles from the first to subsequent cycles can be attributed to this transition period. Additionally, there is a slight variation in the axis of rotation during this transition period, although it is difficult to discern.

From the three example cases examined thus far (Figures 9, 11, and 12), it might be speculated that the Euler-Quat cycles are always completed before the fixed-Quat cycles, in which case it might then be assumed that the Euler-Quat method (i.e., using the DCM^{-1} of Equation 26 or 95 versus using the DCM'^{-1} of Equation 162) is more efficient. However, which cycle is completed first is a consequence of the relative signs of the gyro ratios, as will be shown next.

If we had chosen the gyro pitch rate (about *eb2*) to vary at -1 °/s instead of $+1$ °/s while the gyro roll and yaw rates (about *eb1* and *eb3*, respectively) remained at $+0.01$ °/s, then the fixed-Quat cycles would be completed before the Euler-Quat cycles (Figure 15). Additional observations can be made in comparing these two opposite-sign gyro pitch rate cases: 1) the maximum first cycle values have switched magnitudes and iteration count (or phase) at which they occur; and 2) the sign of the Euler-Quat roll and fixed-Quat yaw angles are the same (both positive), but the sign of the Euler-Quat yaw and the fixed-Quat roll angles have changed directions between the two cases.

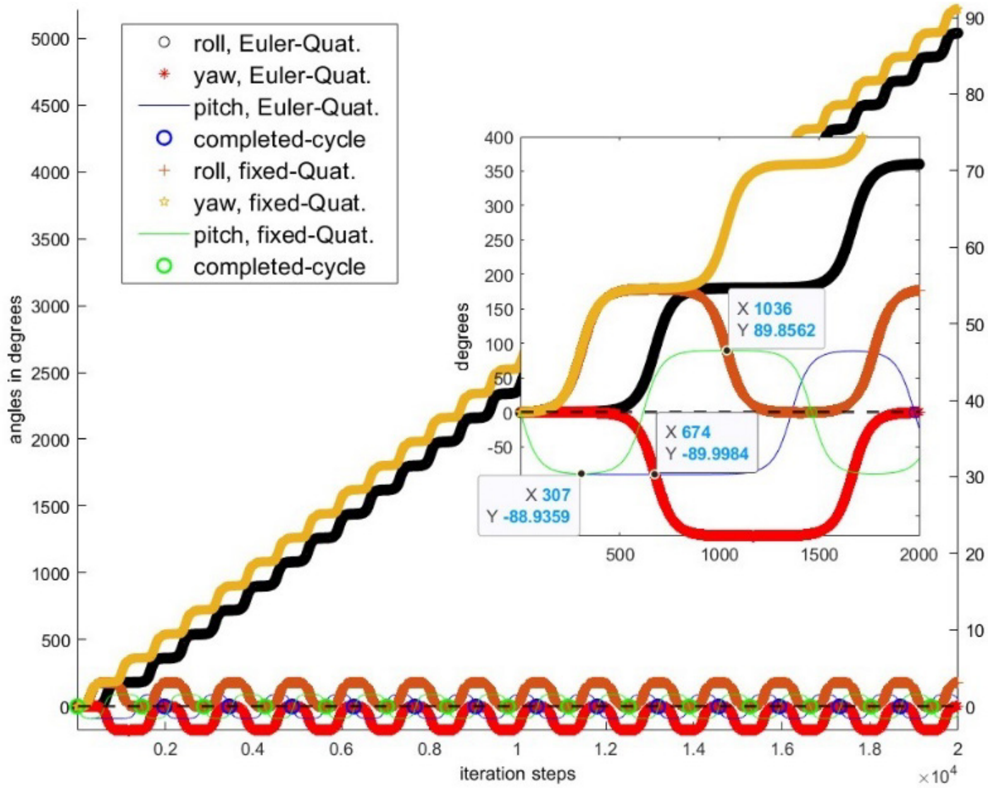


Figure 15. Angular solutions for gyro roll and yaw rates of 0.01 °/s (like Figure 12), but with a gyro pitch rate of −1 °/s (unlike Figure 12).

The 3D body rotation pattern corresponding to Figure 15 is shown in Figure 16. Both the Euler-Quat and fixed-Quat methods continue to track the same body motion, though the axis of rotation now lies close to the earth's negative y axis, so that the change in body axes from one time step to the next is CW, whereas it was CCW for the positive pitch rate case of Figure 14. In addition, if we isolate and magnify the orientation of the body axes at the 307th iteration for the fixed-Quat solution and the 674th iteration for the Euler-Quat solution, as shown in Figure 17, we see the two solutions for *eb1* are now closely aligned with +*ee3* (whereas for the magnified CCW rotating case of Figure 15, they were tilted away from −*ee3*, at the analogous phase angles). The fact that *eb1* is more closely aligned with the vertical axes when the CCW rotating pitch angles are close to $-\pi/2$ (Figure 17), than when the CW rotating pitch angle are close to $+\pi/2$ (Figure 14), is due to the fact that the *eb1* and *eb3* conic surfaces are mirrored in the x–z (or *ee1*–*ee3*) planes for these two opposite-sign gyro pitch rates.

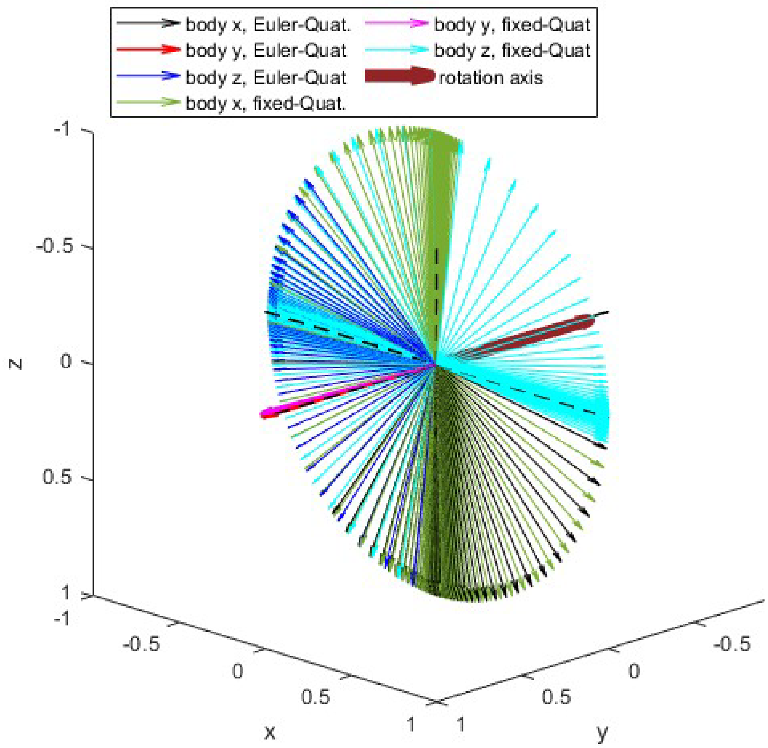


Figure 16. 3D plot of the body axes (*x*, *y*, and *z*, or *eb1*, *eb2* and *eb3*) trajectories for the first 1300 iterations of Figure 15.

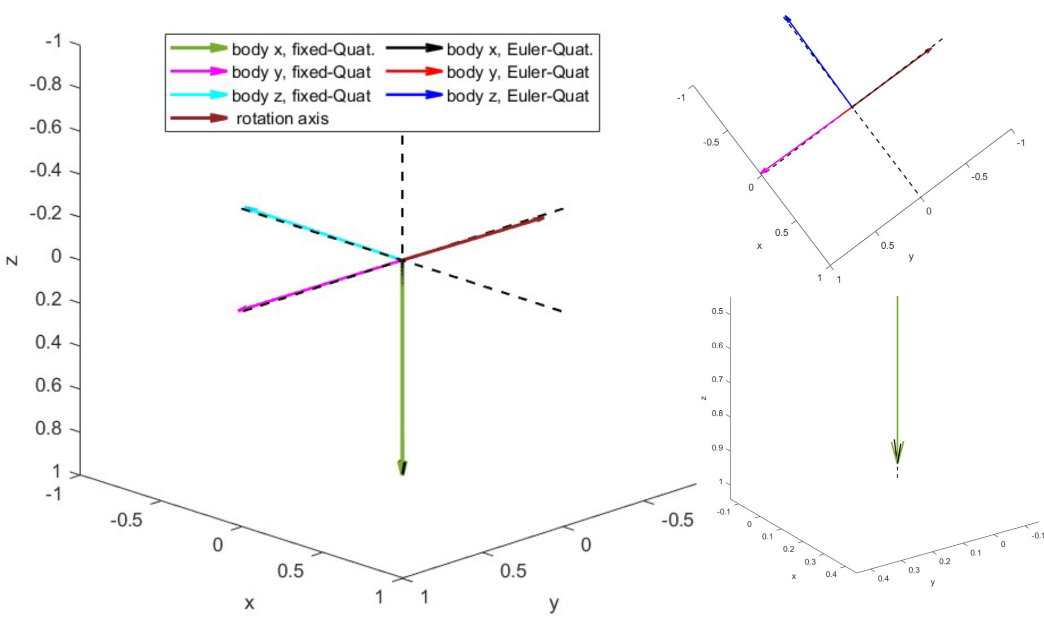
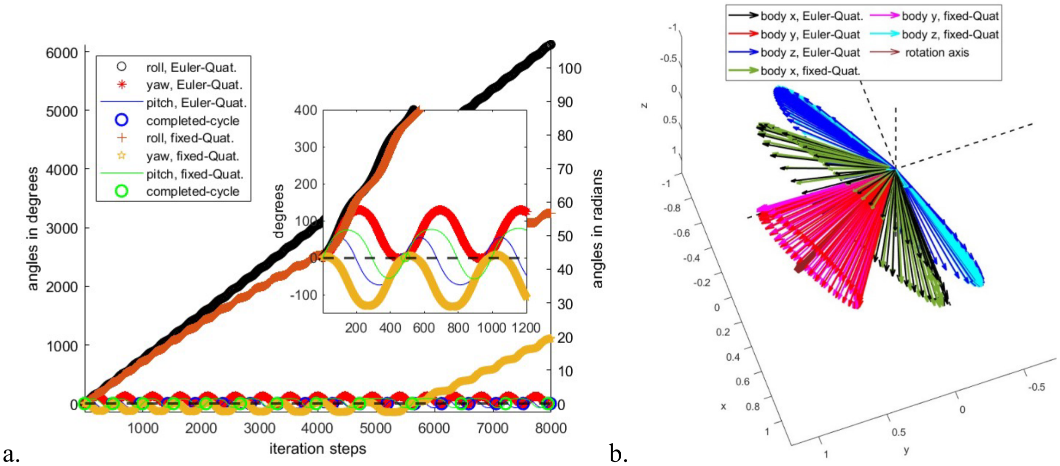


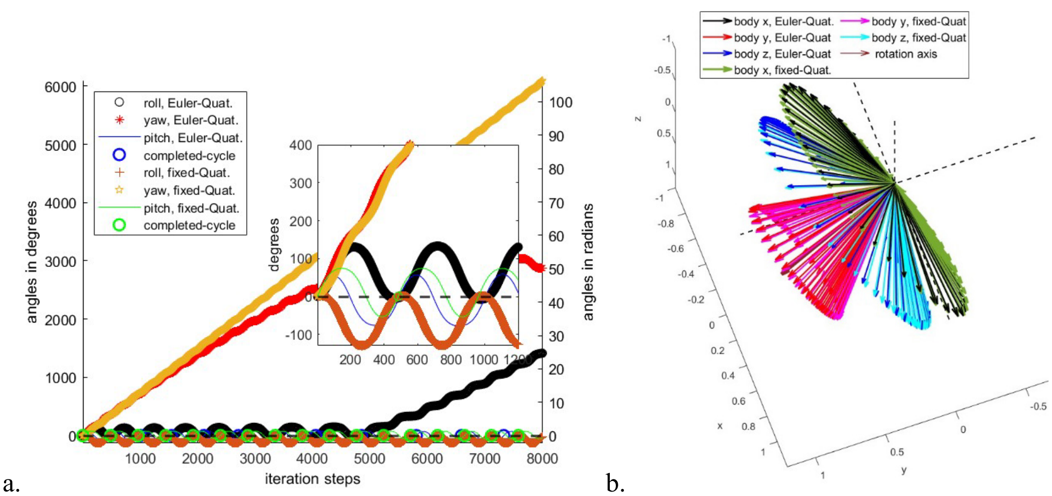
Figure 17. Magnification of body axes locations of Figure 16 at the iteration number for the passage of *eb1* closest to the $+z$ ($+ee3$) and pitch angles closest to $-\pi/2$ within the first cycle.

Other variations of gyro combinations are shown in Figures 18 and 19, where all gyro rates are different. For example, Figure 18 shows the angle and conic surfaces for a gyro yaw:pitch:roll ratio of 0.2:1.0:0.5 °/s, whereas Figure 19 shows the reverse ratio of 0.5:1.0:0.2 °/s.



a. b.

Figure 18. a) Angular solutions for gyro *R*:*Q*:*P* ratio of 0.2:1.0:0.5 °/s, b) 3D plot of the body axes trajectories for most of the first cycle.



a. b.

Figure 19. a) Angular solutions for gyro *R*:*Q*:*P* ratio of 0.5:1.0:0.2 °/s, b) 3D plot of the body axes trajectories for most of the first cycle.

Consistent with the previous patterns, it can be seen that the axis with the smallest gyro rate has the largest conic-section cone angle (*R* about body-*z*, or *eb3* in Figure 18; and *P* about body-*x*, or *eb1* in Figure 19), while the axis with the largest gyro rate (*Q* about *eb2* in both figures) has the smallest cone angle. Figures 18 and 19 also provide other examples of the concept of algorithm switching and how it has been used in the literature. In particular, it can be seen in the expanded inset of Figure 18a that when the extrinsic (fixed-Quat) solution for the pitch angle gets close to $\pm\pi/2$ (≅89.9°), the intrinsic (Euler-Quat) solution is not as close (≅87°). Hence solutions/authors address the singularity problem by switching from their starting algorithm (not necessarily intrinsic to extrinsic; there are many angle rotation sequences to choose from) when its pitch angle gets close to $\pm\pi/2$, to an alternative algorithm, then switching back to the starting algorithm when the pitch angle of the alternative algorithm itself gets close to $\pm\pi/2$. Switching back and forth between algorithms avoids division by cosine of the pitch angle closest to $\pm\pi/2$. But we have shown that you don't need to conduct algorithm switching, if you make your iteration time-step proportional to the cosine of the pitch angle.

As far as illustrative examples are concerned, up to this point we have solved the equations of kinematic motion for constant gyro rates for each iteration over tens of cycles. If the gyro rates change with time/iteration (as is likely to be the case in practical applications), the equations derived above can still be used. The only alteration would be that the gyro rates for each iterative solution would be changed appropriately in the matrix update equations for angular rates (Equations 47 and 170), as well as in the iteration time step (Equations 181 and 187). Figure 20 shows an example case where $R(t)$, $Q(t)$, and $P(t)$ change with each iteration. For illustrative purposes, an arbitrary time-dependent formulation was chosen with each gyro changing according to a simple trig function, as follows:

$$R(t) = (1.0\ °/\mathrm{s})\{ sin(s \times \pi/360) + cos(s \times \pi/360) \} \tag{195}$$

$$Q(t) = (1.0\ °/\mathrm{s})\, sin(s \times \pi/360) \tag{196}$$

$$P(t) = (1.0\ °/\mathrm{s})\, cos(s \times \pi/360) \tag{197}$$

where s is the iteration number/count. As can be seen in Figure 20, each axis moves on some form of trigonometric surface that is more complicated than a cone while maintaining perpendicularity with each other, and yet they are in some combination of harmonic motion, perhaps fitting the definition of a Lissajous surface. In this particular numerical solution, the iteration runs from $s = 1$ to 420 (to avoid visually overpopulating the image), in addition, to avoid time-step differences affecting the comparison of Euler-Quat versus fixed-Quat, the minimum time step for each iteration of Equations 181 and 187 was used for both solutions. The rotation axis in Figure 20 changes with each time/iteration step, as expected for time-varying gyro rates.

Figure 20 is the most "dynamic" example of how the kinematic equations of rotational motion developed here can be used to mathematically describe 3D body rotations without encountering singularities (as the report title indicates). The 3D plots we have shown examine the motion of the body frame in terms of plotting the projection of each of the body unit vectors onto the earth-fixed axes. If we were interested in plotting the motion of a particular body vector, we would simply be plotting the projection of that body vector onto each of the three body axes (resulting in some scaled version of each of the body axes) then plotting the projection of those scaled body axes onto the earth frame. Engineering-wise, it could be imagined that one or more of the axes in Figure 20 represents the motion of a robotic arm, a ground-based tracking system for a fixed-wing or rotary aircraft or spacecraft, as it executes a real or simulated task or mission.

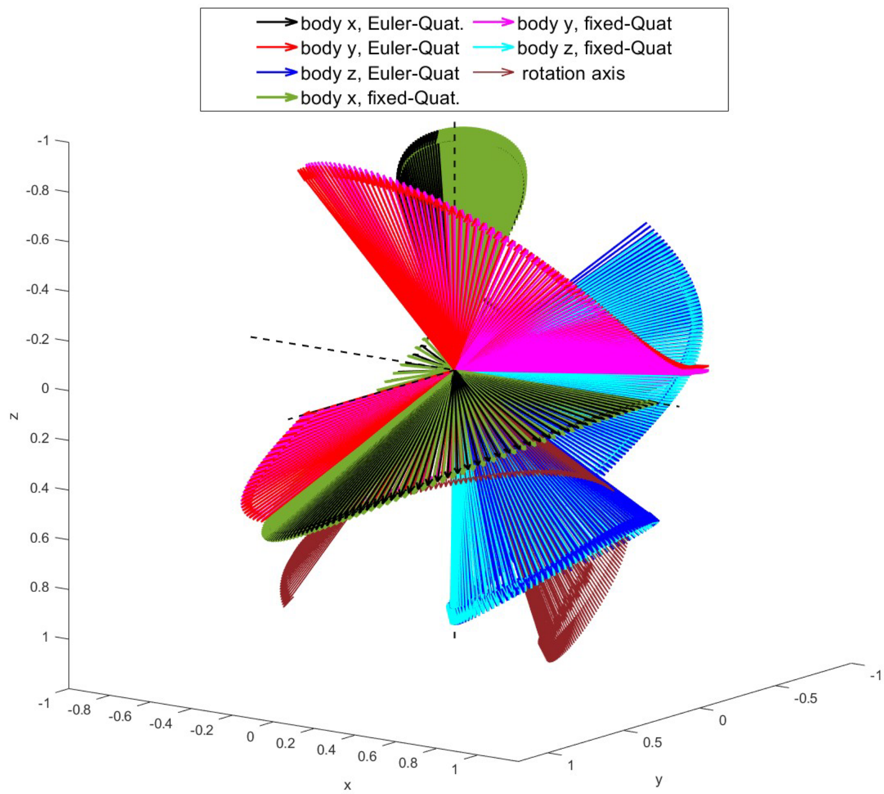


Figure 20. 3D plot of the body axes trajectories for the first 420 iterations of using iteration/time varying gyro rates *R*(*t*), *Q*(*t*), and *P*(*t*) given by Equations 195–197.

Having completed our comparison of the intrinsic versus extrinsic solutions (showing their equivalence in all cases examined), we will now compare the intrinsic DCM^{-1} solution of Equation 26 or 95 with the solution for updating $\vec{a}'$ using the fully quaternion-based (and traditional quaternion) method outlined in Figure 6.

We will not demonstrate the comparison over the same range of R:Q:P ratios as done in Figures 9–20 but rather look at one representative case. In particular, Figure 21 gives a 3D comparison between the intrinsic solution of Equation 26 or 95, with that based on Equations 121, 122, and 82 for the arbitrarily chosen gyro rates R, Q, and P in the ratio 0.5:2:1. The legend designates Euler-Quat for the Equation 26 or 95 solution, and Quat-matrix for the solution based upon Equations 121, 122, and 82. The body unit-vector trajectories are shown for roughly ¾ of the first cycle. At first glance, the two solutions look similar, as we expected they would. But upon closer inspection, it can be seen the body-*x* and body-*z* unit vectors (in particular), predicted by the $q(t + \Delta t) = \dot{q}\Delta t + q(t)$ solution, appear to increase in magnitude in comparison with the corresponding unit vector solutions using the Euler-Quat DCM^{-1} formulation. This was quantifiably verified, finding that after ≅¾ of the first rotation cycle the magnitude of body-*z* unit vector increased from its starting (defined) magnitude of 1.0 to 1.030, whereas the Euler-Quat-based unit vectors held constant at their starting values of 1.0. It might be speculated that the cause of this 3% increase over ¾ of the cycle is due to the approximation we are making in holding $\dot{q}$ at a constant value over each Δt time increment. As a check on this speculation, we computed the magnitude of $|q|$ after reaching ¾ of the cycle and found it to be 1.015 instead of remaining at the theoretical value of 1.0, used in all the relevant quaternion derivations.

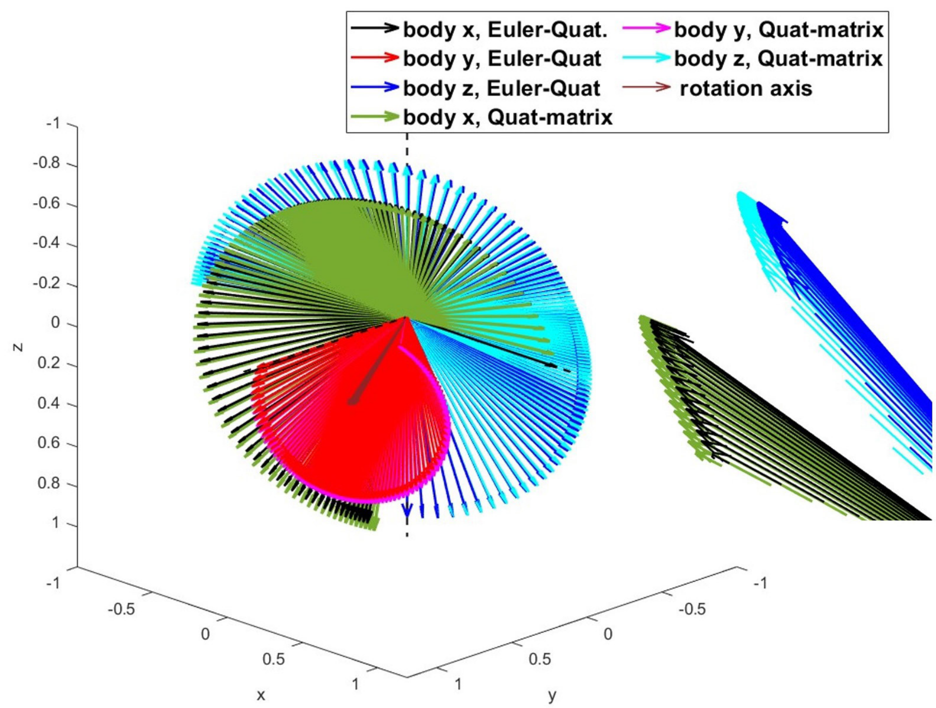


Figure 21. Comparing body axes trajectories over ≅¾ of first cycle, as solved using the $\mathbf{DCM^{-1}}$ (Euler-Quat) method with $\Delta t = (|cos(\theta)||)/\sqrt{1 + R^2 + Q^2 + P^2}$, vs. using $q(t + \Delta t) = \dot{q}\Delta t + q(t)$, as outlined in Figure 6, with gyro rates *R*, *Q*, and *P* in the ratio: 0.5:2.0:1.0.

In an attempt to remedy the error in maintaining unit-magnitude body axes, we renormalized q after each iteration as follows:

$$\tilde{q}(t + \Delta t) = (\dot{q}\Delta t + q(t))$$

$$q(t + \Delta t) = \tilde{q}(t + \Delta t)/|\tilde{q}(t + \Delta t)| \tag{198}$$

Figure 22 shows the results of using a normalized q (Equation 198) after each iteration in the q-matrix of Equation 82. Indeed, the body unit vector axes now remain unity for both solution methods, and clearly the solutions are now visually near identical (vs. Figure 21). Figure 22 was iteratively solved using the same time step, $\Delta t = (|cos(\theta)||)/\sqrt{1 + R^2 + Q^2 + P^2}$, for both the DCM^{-1} (Euler-Quat) and $q(t + \Delta t) = \dot{q}\Delta t + q(t)$ methods.

References for the need to renormalize the quaternion, such as shown in Equation 198, can be found in the literature (e.g., Phillips 2001, p. 728; Rogers 2003).

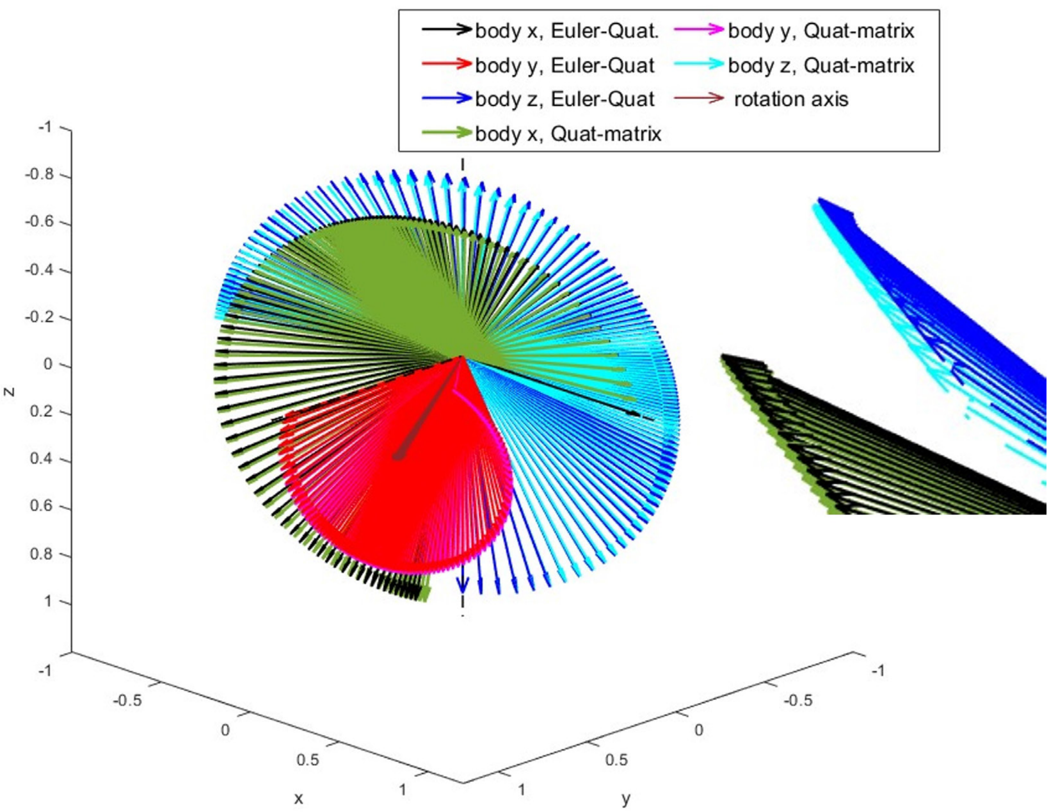


Figure 22. Same comparison as shown in Figure 21, but with $q(t + \Delta t)$ normalized according to Equation 198 after each iteration time step.

Computationally, squaring each coefficient of $\tilde{q}$, adding their squares together, and then taking the square root of that sum, involves eight math operations to obtain the magnitude of $|\tilde{q}(t + \Delta t)|$, then dividing each coefficient by that magnitude requires four more operations, for a total of 12 additional math operations to renormalize the q coefficients using Equation 198. If we add 12 more operations to the 75 math operations outlined in Figure 6, we get a new total of 87 operations to update $\vec{a}'$ for each iteration using the $q(t + \Delta t) = \dot{q}\Delta t + q(t)$ quaternion method. But, since we do not need to be concerned about a mathematical singularity when using this method, suppose we keep a constant time step; in particular, suppose we start with the case where $\Delta t = 1$ for the quaternion solution, versus the time step $\Delta t = (|cos(\theta)||)/\sqrt{1 + R^2 + Q^2 + P^2}$ for the Euler-Quat solution:

$$\Delta t(\text{Quat-matrix}) = 1 \qquad (199)$$

$$\Delta t(\text{Euler-Quat}) = (|cos(\theta)||)/\sqrt{1 + R^2 + Q^2 + P^2} = (|cos(\theta)||)/\sqrt{1 + |\vec{w}|^2} \qquad (200)$$

Figure 23 shows how this difference in time increments affects the solution, showing only the trajectory of the *z*-body axis, and only for the first 120 solution iterations to avoid clutter. We find that the quaternion solution completes roughly 20% more (1.2 times) of the rotational trajectory than the Euler-Quat solution does over the 120 iterations. But how do we determine that specific ratio?

To compute the total rotational angle that the body makes for a given number of numerical solution iterations, we need to sum the product of the body's rotational rate $|\vec{w}|$, given by Equation 194, times the time increments, which if the rate is constant for all time increments, becomes $|\vec{w}|$ times the elapsed time of rotation over the given number of iterations. That is,

$$N_Q = \textstyle\sum_1^s |\vec{w}| \Delta t(\text{Quat} - \text{matrix}) = |\vec{w}| \sum_1^s \Delta t(\text{Quat} - \text{matrix}) \qquad (201)$$

$$N_E = \textstyle\sum_1^s |\vec{w}| \Delta t(\text{Euler} - \text{Quat}) = |\vec{w}| \sum_1^s \Delta t(\text{Euler} - \text{Quat}) \qquad (202)$$

where N_Q is the total angle that the body has rotated as calculated by the quaternion solution, over *s* iterations, and N_E is that angle as calculated by Euler-Quat solution, with the particular Δt increments for each iteration given by Equations 199 and 200.

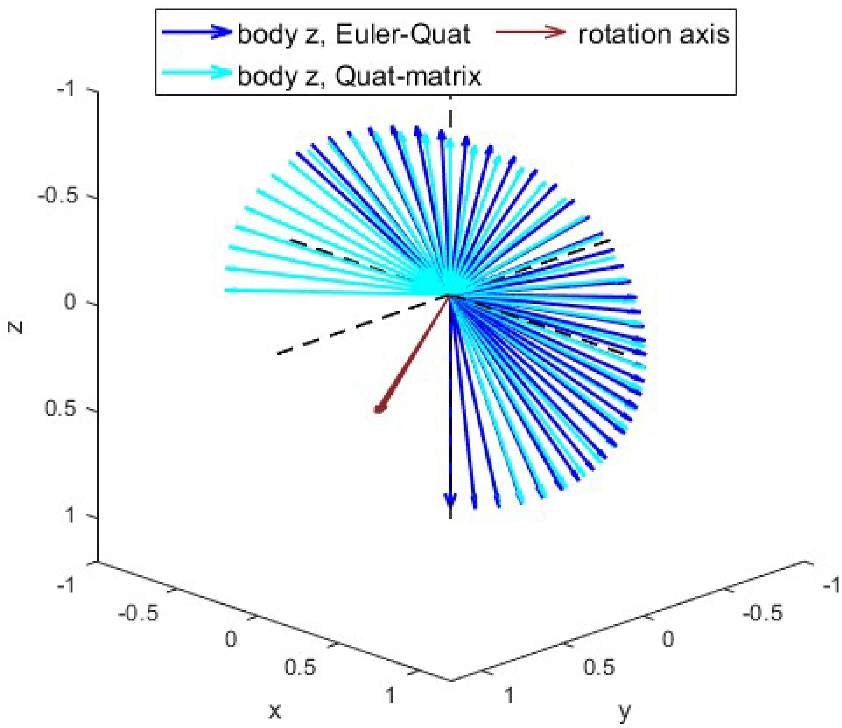


Figure 23. Same comparison as shown in Figure 22, but with a time step of unity, $\Delta t = 1$, for the normalized $q(t + \Delta t)$ solution, vs. the time step of $\Delta t = (|cos(\theta)||)/\sqrt{1 + |\vec{w}|^2}$ for the Euler-Quat solution (to avoid visual clutter, only the body z axes comparison is shown).

We stopped the number of iterations in the plot of Figure 23 before completing one full rotation of the Euler-Quat solution (i.e., before completing $N_E = 360°$ of yaw, pitch, and roll for the slower Euler-Quat solution). How much would this ratio change if we went a full cycle? Continuing the simulation for a full cycle, without showing the corresponding plot, we found (for this particular R, Q, and P ratio of 0.5:2.0:1.0) using Equation 201, that the quaternion solution predicted the motion of the rotating body through 551° of pitch, roll, and yaw, versus 360°, using Equation 202 for the Euler-Quat solution. So, over a full rotation cycle of the Euler-Quat solution, the quaternion formulation is 1.53 times (551°/360°) faster. However, before we compare efficiencies, we must add another nine math operations (ops), to compute the Δt in Equations 181, 200, and 202 (i.e., square each gyro value [3 ops], add those 3 squares, plus 1 together [3 ops], take that square root of that number and invert it [2 more ops], and multiply that result times Δt, for a total of 9 ops) to the 48 needed to update $\vec{a}'$ using the DCM^{-1} method of Figure 6, bringing the total to 57 ops. Since the quaternion solution takes 87 ops per iteration (itr) versus 57 ops/itr for the Euler-Quat solution, but the quaternion operation rotates the body 1.53 times further (on average) per its 87 ops than the Euler-Quat per its 57 ops, the computational advantage of the quaternion over the Euler-Quat solution is negligible (~1.0 = 1.53×[57/87]), at least for this example case where the maximum pitch angle $|\theta| \cong |-75.93°| = 75.93°$, and we use the Δt*'s* of Equations 199 and 200. We next look at how much of a speed advantage there might be if the pitch angles were even closer to 90°.

We expect the quaternion computational speed advantage to increase if the pitch angle approaches close to π/2, because Δt(Euler-Quat) $\propto (|cos(\theta)|$ for the Euler-Quat solution gets very small in the region of π/2, whereas we are currently treating Δt(Quat-matrix) as a constant (Equations 199 and 200). To get an idea of how large this advantage might be, we can look at the case of Figures 15–17, where θ reaches $-89.9984°$. In this case, the body's pitch rate about its y-axis is 100 times faster than about its yaw and roll axes, so the body frame does not have time to yaw or roll to any significant degree before it has rotated a complete cycle around its y-axis. Figure 24 shows the solution corresponding to Figures 15–17 up to the 674th iteration for only the body-z unit vectors (to avoid clutter), corresponding to θ reaching –89.9984°, or about ¼ of a full rotation cycle around the axis of rotation. The quaternion solution appears to rotate the body-z axis more than 4 times that angle, in fact, the quaternion solution for the body-z axes rotated it around the rotational axis a full 674° (note, in this simulation the quaternion solution is computing 1° of rotation per iteration). Using Equations 199–202, this translates into the quaternion solution computing 7.4 times more rotational trajectory than the Euler-Quat solution. If we continue to compute the solution until the Euler-Quat solution rotates the body through one complete cycle, we find the quaternion algorithm has rotated the body by 1,976° compared to the Euler-Quat's 360° rotation, so on average, for this high pitch angle case, the quaternion solution computes 1,976°/360° = 5.49 times faster than the Euler-Quat solution. Using the same logic to estimate the computational efficiency (as used for the gyro ratios of Figure 23), we can say for the gyro ratios of Figure 24, where the pitch angle is near (within less than 100th of a degree of) π/2, that on average, the quaternion solution will be on the order of 3.6 times more efficient than the Euler-Quat solution (i.e. symbolically: Quat-matrix/Euler-Quat $= 3.6 = 5.49 \times [57/87]$). Again, we are assuming that all math operations are computationally equivalent, and we are using the Δt*'s* of Equations 199 and 200 (in Section 9 we will modify the time steps to create a more equitable solution comparison).

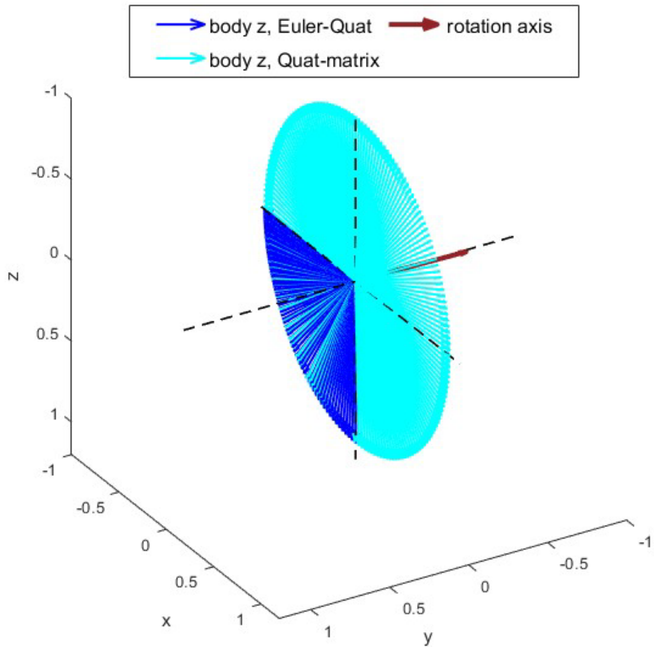


Figure 24. Same gyro ratios as shown in Figure 16, where θ reaches –89.9984°, with a time step of unity, $\Delta t = 1$, for the normalized $q(t + \Delta t)$ solution, vs. a time step of $\Delta t = (|cos(\theta)||)/\sqrt{1 + |\vec{w}|^2}$ for the Euler-Quat solution (to avoid visual clutter, only the body z axes comparison is shown).

Having just explored the comparison if the Euler pitch angle is much greater than 75°, what if the pitch angle is significantly less than the 75° maximum angle of Figure 23, then we can expect computational advantage would swing in favor of the Euler-Quat solution. Let us evaluate how much of an advantage can be gained with another simulation test case. Figure 25 plots the 3D trajectory using both solutions for the case where the yaw rate is 5 times more than both the pitch and roll rates, namely let R, Q, and P have the ratio 1.0:0.2:0.2. This plot looks different than the others, since the faster yaw rate will rotate the body through a full cycle before the pitch and roll rates have a significant effect on the Euler pitch angle; in fact, the maximum pitch angle (ignoring the sign) in this example is computed to be $-27.32°$. In this case, we find that the Euler-Quat solution rotates the body 360° in the same number of iterations as the quaternion solution uses to rotate the body 375°. Using the efficiency formulation applied to the results of Figure 23 and 24, we find for the Figure 25 case that the quaternion solution completes 1.04 times (375°/360°) more rotation, for the same number of iterations, than the Euler-Quat solution. But when we factor in the number of math operations per iteration (87 vs. 57), we conclude that on average in this lower pitch angle case, the Euler-Quat method is on the order of 1.5 times more efficient than the quaternion method. Or conversely, the Quat-matrix is 32% less efficient that the Euler-Quat method (Quat-matrix/Euler-Quat = 0.68 = 1.04×[57/87]) for the same amount of body rotation, at least for this range of maximum Euler pitch angles, $\lesssim 25°$.

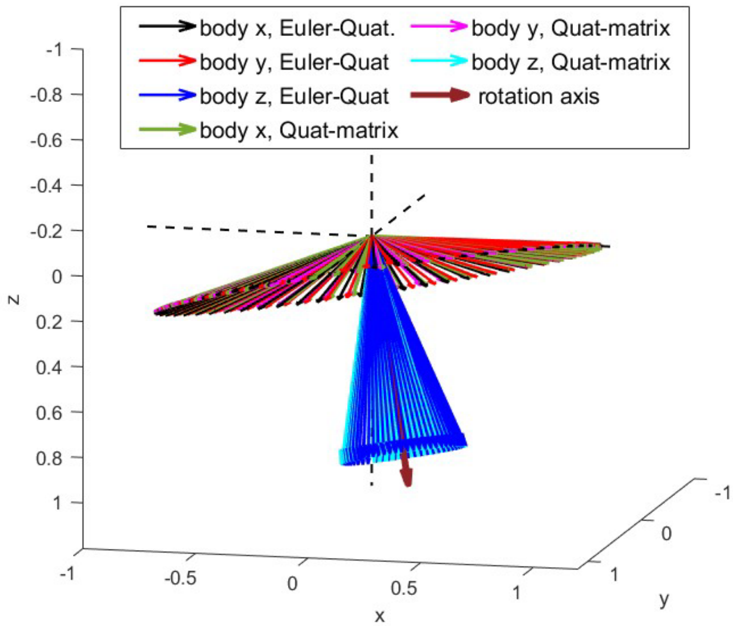


Figure 25. Comparing body axes trajectories over the first full Euler-Quat-computed cycle, as solved using the DCM^{-1} (Euler-Quat) method with $\Delta t = (|\boldsymbol{cos(\theta)}||)/\sqrt{\mathbf{1} + |\vec{\boldsymbol{w}}|^{\mathbf{2}}}$, vs. using $\boldsymbol{q(t + \Delta t) = \dot{q}\Delta t + q(t)}$, with gyro rates *R*, *Q*, and *P* in the ratio 1.0:0.2:0.2.

In summary, the debate over computational efficiency of the quaternion vs. the Euler methodology for solving the body rotation problem is unsettled, because it depends on the average magnitude of Euler pitch angle, the efficiency of evaluating trig functions vs. other math operations, and the time step size. We will examine the time step factor in more detail in Section 9.

Finally, we will compare the Euler-Quat solution using Equations 182–184 with the $\vec{w} \times \vec{a}'$ solution of Equation 151 as specified in Figure 6 (both of which use the time step of Equation 181 to update the Euler angles needed to evaluate $\vec{w}$ in Equation 106 or 107). Figure 26 shows this comparison for the same gyro rates used in Figures 21–23, namely *R*, *Q*, and *P* in the ratio 0.5:2:1. Like the first impression derived from Figure 21, the two solutions in Figure 23 look similar, as we expected they should. But once again, it can be seen upon closer inspection that the body-*x* and body-*z* unit vectors (in particular), predicted by the $\frac{d\vec{a}\prime}{dt}$ solution, appear to increase in magnitude in comparison with the corresponding unit vector solutions using the Euler-Quat DCM^{-1} formulation. This too was quantifiably verified, finding that after ≅¾ of the first rotation cycle the magnitude of $\frac{d\vec{a}\prime}{dt}$-based unit-vectors increased from their starting (defined) magnitude of 1.0 to 1.050, whereas the Euler-Quat–based unit vectors held constant at their starting values of

1.0. Suspecting the cause for the discrepancy is the slight error incurred by holding $\frac{d\vec{a}'}{dt} = \vec{w} \times \vec{a}'(t)$ constant over each time step Δt, we sought to resolve the error in a manner similar to that used in Equation 198. Namely, we chose to normalize the rotated vector output of Equation 151 after each iteration using the following formulation:

$$\tilde{\vec{a}}'(t+\Delta t) = (\vec{w} \times \vec{a}'(t))\, \Delta \boldsymbol{t} + \vec{a}'(t)$$
$$\vec{a}'(t+\Delta t) = \tilde{\vec{a}}'(t+\Delta t) / \left| \tilde{\vec{a}}'(t+\Delta t) \right| \tag{203}$$

Figure 27 shows that the renormalization of $\vec{a}'$ after each iteration has the same effect on preserving the body unit-vectors as renormalization of the q coefficients had in using the quaternion-based solution. However, the renormalization of Equation 203 requires nine operations (six to square the three $\vec{a}'$ vector components, add them together, then take their square root) followed by three more to divide the $\vec{a}'$ vector components by the vector magnitude. Thus, it will take 40 math operations outlined in Figure 7 plus these 9 more for a total of 49 operations to update $\vec{a}'$using Equation 151, versus the 48 needed using the DCM^{-1} solution of Figure 5 or 6, plus the 9 more for either method needed to evaluate the iteration time step Δt of Equation 181 for a total of 58 math operations for the $\vec{w} \times \vec{a}'$ solution, versus 57 using the DCM^{-1} solution.

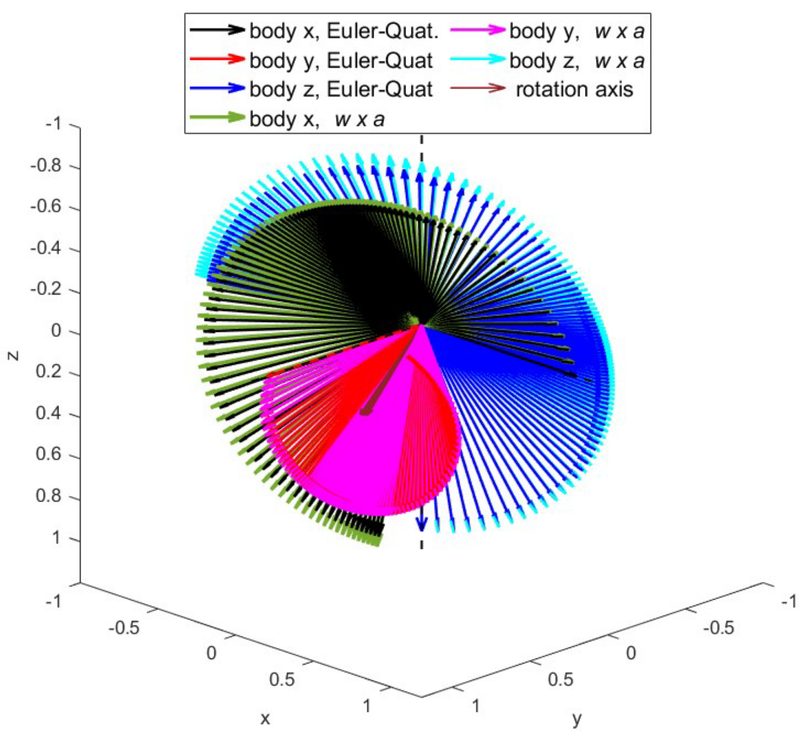


Figure 26. Comparing body axes trajectories over ≅ ¾ of first cycle, as solved using the $\mathbf{DCM^{-1}}$ (Euler-Quat) method, vs. solved using $\frac{d\vec{a}'}{dt} = \left(\vec{w} \times \vec{a}'(t)\right) \Delta t + \vec{a}'(t)$, with gyro rates R, Q, and P in the ratio: 0.5:2.0:1.0 for $\Delta t = (|\cos(\theta)||)/\sqrt{1+|\vec{w}|^2}$.

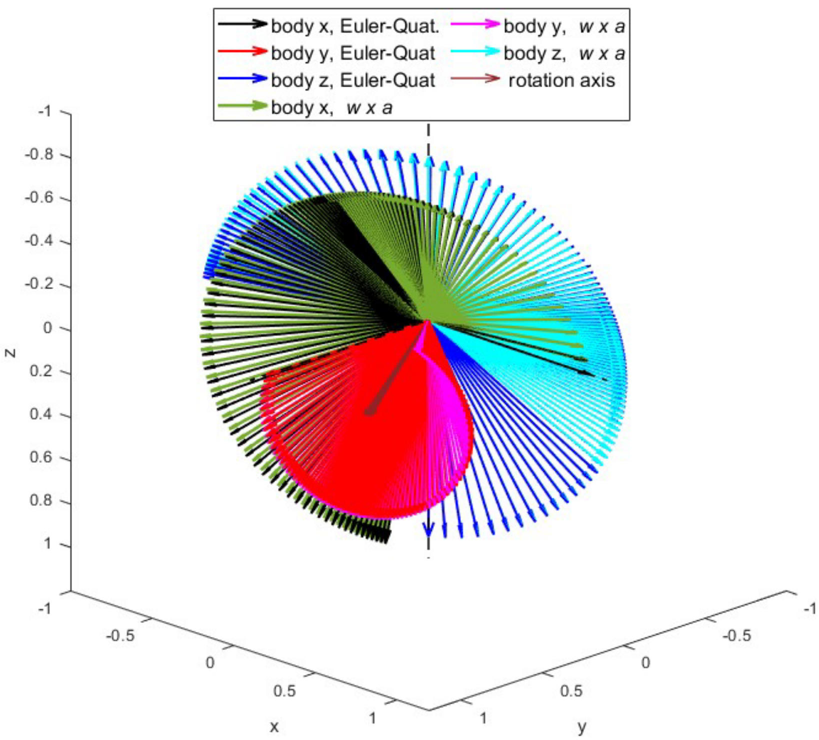


Figure 27. Same comparison as shown in Figure 25, but with $\vec{a}'$ normalized after each iteration according to Equation 203.

9. Generalizing the Time Step

In Section 8 we showed how the numerical integration time step can be used to avoid the mathematical singularity associated with the traditional Euler solution (denoted here as the Euler-Quat solution) by making that time step proportional to $|cos(\theta)|$. We also argued that it was reasonable to make it inversely proportional to $|\vec{w}|$ (such that the time step would be smaller for large $|\vec{w}|$ and larger for small $|\vec{w}|$), arriving at Equations 181 and 200 for the Euler-Quat time step. Although such a time step would eliminate the singularity-causing division by $cos(\theta)$ in the Euler angle solutions, Equations 182–184, it would also slow down the size of the angular changes per iteration as θ approaches $\pm\pi/2$. This latter effect can be seen in the plot–vector-density of the body axes as they rotate close to 90° (e.g., Figure 13), and it can also be discerned in Equation 202 where the dependence of the body's cumulative angle of rotation depends on its rotation rate $|\vec{w}|$ times the elapsed rotation time, the latter of which slows down when $cos(\theta)$ is near $\pm\pi/2$.

We can partially compensate for this inefficiency in size of the Euler angle updates by adding a scaling factor to the time step of Equation 200. For example, a scaling factor in the form of a power law with a maximum at $\theta = \pm\pi/2$ would help to offset the diminishing $cos(\theta)$ dependence. A scaling law that has this behavior is

$$\text{Euler Time Step Scaling factor} = \boldsymbol{A^{sin^{2n}(\theta)}} \tag{204}$$

We found that if we use A =10 and n = 4 in Equation 204, and then multiply that scaling factor times $cos(\theta)$, it slows the drop-off in the numerator of the Euler-Quat time step of Equation 200 as shown in Figure 28. We see there that the product of $(10^{sin^8(\theta)})cos(\theta)$ dips below unity (to a minimum of ~0.81 at ~43°), then rises above unity at ~59° (to a maximum of ~1.48 at ~75°) before crossing back under unity at ~84°. The overall average of this product from 0° to 90° is found to be 0.99, which is essentially the same as the value of unity used in the numerator of Equation 199 for the Quat-matrix time step.

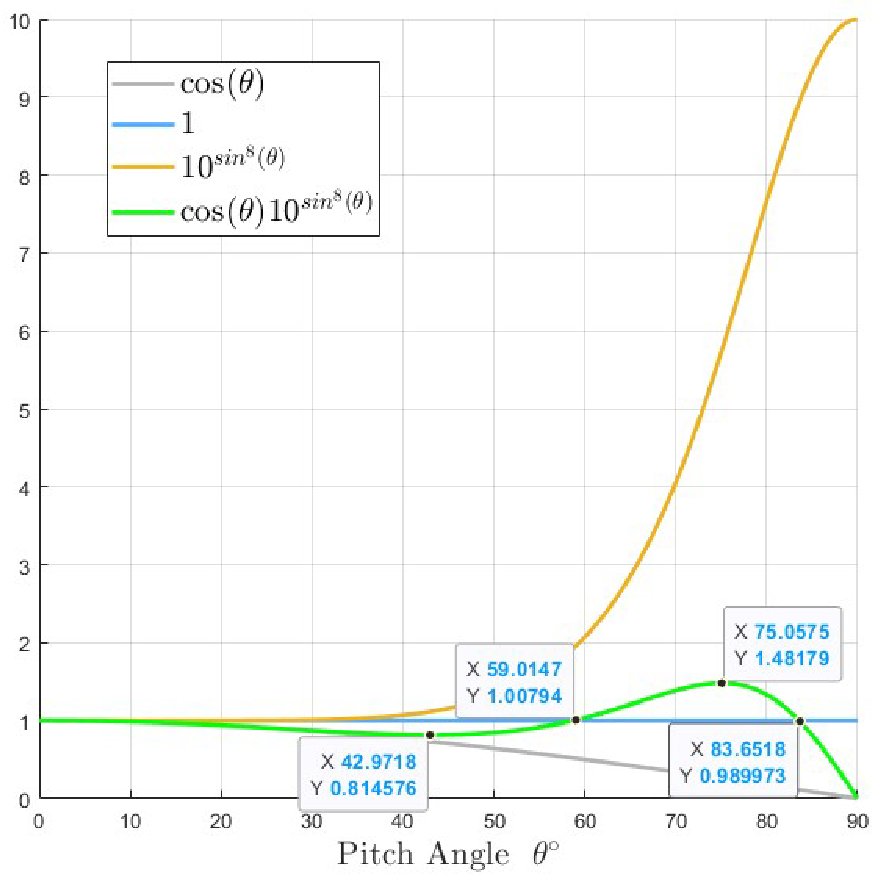


Figure 28. Effect of a power-law scaling factor of the form $10^{sin^8(\theta)}$ on minimizing the $cos(\theta)$ drop-off in the numerator of the Euler-Quat solution time step of Equation 200.

Inserting the $(10^{sin^8(\theta)})cos(\theta)$ scaling factor into Equation 200 yields a modified time step for the Euler-Quat solution of the form

$$\Delta t(\text{Euler-Quat}) = (10^{sin^8(\theta)})(|cos(\theta)||)/\sqrt{1+|\vec{w}|^2} \tag{205}$$

We will show in forthcoming examples how using this time step is more computationally efficient than using Equation 200. But first we should also acknowledge that using a simple time step of unity for the Quat-matrix solution ignores the desirable feature of having a time step that gets smaller for large $|\vec{w}|$ and larger for small $|\vec{w}|$, which can be provided by the inverse dependence on $\sqrt{1+|\vec{w}|^2}$, such as used in the Euler-Quat time step. Thus, to make the Quat-matrix time step more suitable for a variety of gyro rates, we should modify Equation 199 into the form

$$\Delta t(\text{Quat-matrix}) = 1/\sqrt{1+|\vec{w}|^2} \tag{206}$$

Exchanging the time step formula of Equation 205 for Equation 200 adds 9 more math operations (i.e., evaluating $sin^8(\theta)$ entails 7 ops, e.g., $sin^2(\theta)$ is 1 operation, $sin^3(\theta)$ is 2 operations, etc., then using a lookup table to evaluate 10 raised to the $sin^8(\theta)$ power is one more operation, and multiplying that value of $10^{sin^8(\theta)}$ times our previous expression, Equation 200, is the 9th math operation). When we add 9 more ops to the 57 we previously needed to evaluate each Euler-Quat itr, we get a new total of 66 ops/itr. Likewise, modifying Equation 199 into Equation 206 adds another 9 math operations (as enumerated previously in the discussion of computational efficiency following Figure 22) to the time step used for each Quat-matrix iteration, bringing that new total to 96 (= 87 + 9) ops/itr.

We would now like to reevaluate the computational efficiencies of the Euler-Quat versus the Quat-matrix solution methodologies using the more generalized time steps of Equations 205 and 206. Rather than reexamining every simulation case study of Section 8, we will look at the two extreme-efficiency examples. First, applying the time steps of Equation 205 and 206 to the case study of Figures 15–17 and 24, for the R:Q:P gyro ratio of 0.01: −1.0:0.01 (when the pitch angle θ is nearly π/2), we arrive at the comparison plot of Figure 29. For illustrative purposes the maximum itr count was stopped short of completing 360° for the Euler-Quat solution, to show that the Quat-matrix solution is still ahead of the Euler-Quat solution, but by far less than shown in Figure 24. When we run the itr count high enough that the Euler-Quat solution completes one full revolution, the average ratio of the Quat-matrix computed body rotation to the Euler-Quat solution is 494°/360° = 1.37, but the ratio of math ops per iteration for the Quat-matrix vs. the Euler-Quat is 96/66 = 1.45. Hence, the efficiency of the Quat-matrix versus the Euler-Quat in terms of average angle of rotation per math operation over a complete 360° is Quat-matrix/Euler-Quat = 0.94 (= 1.37 × [66/96], or = 1.37/1.45). That is, assuming all math operations are equivalent, the Quat-matrix method is 6% less efficient than the Euler-Quat method when the angle per iteration and the operations per iteration are both factored into the efficiency calculation. This is quite different from the analysis of Figure 24 where we found the average ratio for one complete cycle, using previous (less general) time steps of Equations 199 and 200, was Quat-matrix/Euler-Quat = 3.6. Hence, generalizing the time steps into Equations 205 and 206 has effectively canceled out the efficiency advantage that the quaternion approach had over the Euler angle solution, even for this large pitch angle condition.

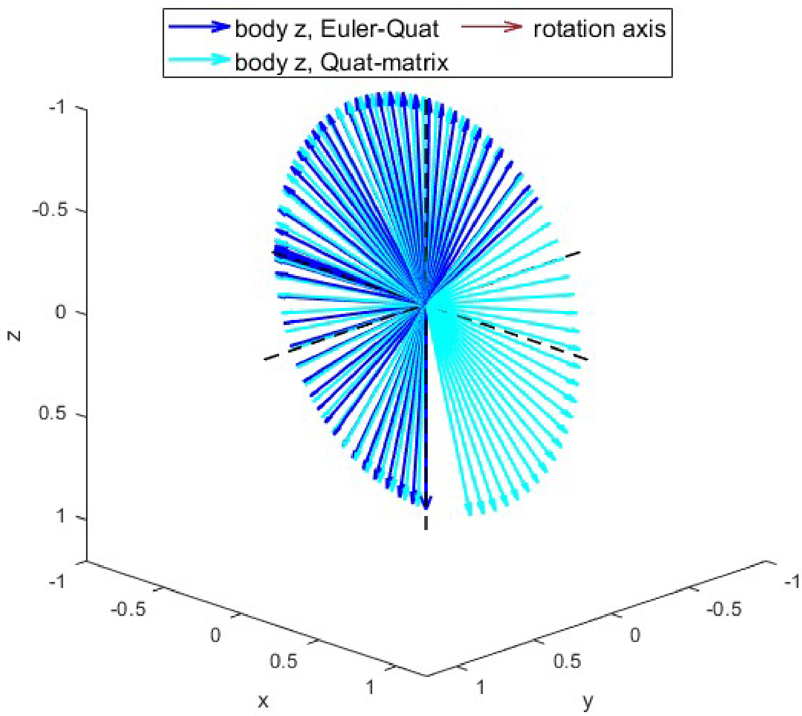


Figure 29. Same gyro ratios as shown in Figures 16 and 24, where θ reaches –89.9984°, with a time step of $\Delta t = 1/\sqrt{1+|\vec{w}|^2}$ (Equation 206) for the normalized $q(t+\Delta t)$ solution, vs. a time step of $\Delta t = (10^{sin^8(\theta)})(|cos(\theta)||)/\sqrt{1+|\vec{w}|^2}$ (Equation 205) for the Euler-Quat solution (to avoid visual clutter, only the body z axes comparison is shown).

Let us now look at the other extreme case studied in Section 8, that of Figure 25 for the gyro ratio of R:Q:P = 1.0:0.2:0.2, where the magnitude of the maximum pitch angle was relatively low, ~27°. Figure 30 shows the result of this comparison using the new/generalized time steps of Equations 205 and 206. Again, for illustrative purposes the maximum itr count was stopped short of completing 360° for the Euler-Quat solution so that it can be seen that the Quat-matrix solution is still slightly ahead of the Euler-Quat solution. When we run the itr count high enough that the Euler-Quat solution completes one full revolution, the average ratio of the Quat-matrix computed body rotation to the Euler-Quat body rotation is virtually the same as it was in the case of Figure 25, namely 374°/360° = 1.04. However, as with Figure 29, the ratio of math ops per itr is slightly different, instead of 87/57 = 1.52 for the case of Figure 25, it is now 96/66 = 1.45 for the Quat-matrix versus the Euler-Quat solution schemes. Hence, the efficiency of the Euler-Quat to the Quat-matrix in terms of average angle of rotation per math operation over a complete

360° is Quat-matrix/Euler-Quat = 0.72 (= 1.04 / 1.45). Thus, for the smaller pitch angle cases, the Quat-matrix solution is 28% less efficient than the Euler-Quat method (due almost entirely to the relative number of math operations per iteration required for the quaternion solution methodology).

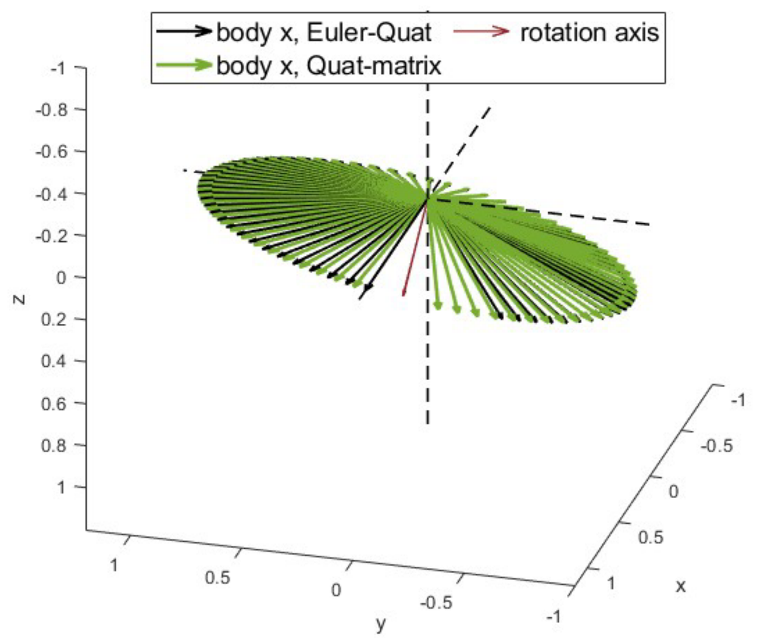


Figure 30. Same gyro ratios as shown in Figure 25, where $\boldsymbol{\theta}$ reaches –27.3°, but with a time step of Δt = $1/\sqrt{1+|\vec{w}|^2}$ (Equation 206) for the normalized $q(t+\Delta t)$ solution, vs. a time step of Δt = $(10^{sin^8(\theta)})(|cos(\theta)||)/\sqrt{1+|\vec{w}|^2}$ (Equation 205) for the Euler-Quat solution (to avoid visual clutter, only the body x axes comparison is shown).

In this section we generalized the time step formulas to be more efficient over all angles and gyro rates, using Equation 205 versus 200 for the Euler-Quat (DCM^{-1}) solution and using Equation 206 versus 199 for the Quat-matrix formulation. The difference in the relative efficiency change was significant. Using the former time steps, we found the Quat-matrix solution could be on average 3.6 times more efficient than the Euler-Quat method (e.g., Figure 24), but using the more generalized/efficient time steps of Equations 205 and 206, we found that the Euler-Quat method is always more efficient (provided all our trig and power-law function values are stored in look-up tables, with no additional operations required to obtain them), close to 30% more efficient for small pitch angles (e.g., Figure 30), though only 6% more efficient for pitch angles near π/2 (e.g., Figure 29).

If we need/want to use less memory by using linear interpolation between stored functional values, then instead of looking up trig or power law function values directly, we would need to perform two math operations to get each functional value (one multiplication and one addition); this would add 14 more math operations per iteration to the current number of 66 ops/itr for the Euler-Quat solution (two for each of the six trig functions [sine and cosine of each Euler angle], and two for each power law value used in the time step). If we recalculated Quat-matrix versus Euler-Quat solution efficiencies using 80 (= 14 + 66) ops/itr for the Euler-Quat solution, we would find for the case of Figure 29 (large pitch angle) that the Quat-matrix would be slightly more efficient than the Euler-Quat, namely 1.14 (= 1.37 × [80/96]), or 14% more efficient. For the case of Figure 30 (smaller pitch angles), the Quat-matrix/Euler-Quat ratio would be 0.87 (= 1.04 × [80/96]), or 13% less efficient. All in all, we might say that using linear interpolation between look-up table values for trig and power law functions, instead of using a look-up table to obtain every functional value, would on average (over the full range of pitch angles) remove any computational advantage that would favor either the Euler angle or the Quaternion solution methodologies.

Thus, we have shown by using this nonsingular, Euler–angle-based solution that the general assertion that the quaternion-based solution is computationally more efficient than the vector algebra solution (e.g., Phillips et al. 2001, p. 718; Kim 2013, p. 44; Ben-Ari 2014; Challis 2020, p. 9) is not universally true. In fact, it would not be true if using look-up tables for all trig and power law function values, but it would be true if using look-up tables (or power series expansions) of higher than linear order (quadratic, cubic, etc.) to compute trig functions.

10. Summary and Conclusions

We started by reviewing the single axis-angle Rodrigues formula for describing the rotation of a vector about an axis (Figure 1). Then we proceeded to show how the multiple–axis-angle Euler rotation formulation is established (Table 3) and put into matrix form, DCM or DCM^{-1} (Equation 10 or 12). We emphasized that the development of the Euler angle equations of motion (DCM or DCM^{-1}) are based on rotational axes that (intrinsically) change with the Euler angles and are not generally perpendicular to each other.

In practical applications, the rotational motion of the body is detected using gyros to measure the rotation rates about perpendicular body axes. To use such gyro measurements to update the Euler angles, it is necessary to add constraint equations (Equations 31–33) consistent with the gyro and orthogonality conditions. These constraints establish solutions for the Euler angular rates as a function of the gyro rates (Equations 47) that can become undefined (i.e., a singularity can arise) when the Euler pitch angle θ reaches $\pm\pi/2$.

Because quaternion algebra is commonly referenced in the literature as the method to use to avoid this aforementioned Euler angle singularity, quaternion algebra is introduced and applied to our rigid body rotational analysis. We first showed how quaternions can be used to describe the single axis rotation problem presented by Rodrigues (Equation 75), then how they can be used to describe rotations about multiple axes (Equation 82). If such a quaternion formulation is used to model rotations about the Euler axes, then that quaternion solution reduces to (Equation 95) the same result as given by the Euler DCM^{-1} equation (Equation 26), with its incumbent singularity problem (Equation 47). However, if we solve for the time-rate of change of the quaternion coefficients as a function of the body's angular velocity (Equations 110–121), we find a way to update the quaternion coefficients from time zero without needing trig functions (i.e., without incurring any singularity problem; Equation 122). Although this formulation has, number-wise, more math operations (ops) per solution iteration (itr), 87 ops/itr (75 as outlined in Figure 7, combined with 12 more for the operations needed to renormalize the quaternion coefficients after each iteration, Equation 198), than the DCM^{-1}, 57 ops/itr (48 as outlined in Figure 6, plus 9 more to evaluate Δt for each time step, Equation 181), we showed in the numerical solutions using a constant quaternion time step (Section 8, e.g., Figure 24) that if the pitch angle is routinely near $\pm\pi/2$, then the quaternion solution can be nearly 4 times more efficient than using the DCM^{-1} to update the solution for a rotating vector, $\vec{a}'$. Whereas if the pitch angle is small, the update may be computationally more efficient by using the DCM^{-1} method (e.g., simulation of Figure 25). However, the time steps used in these simulations were sub-optimal and were improved in Section 9.

An additional vector algebra-based solution for tracking a rotating body was presented, relating the time derivative of a rotating vector $\vec{a}'$ to the cross product of its angular velocity with the vector itself, namely $\vec{w} \times \vec{a}'$, as shown in Equation 151. This solution is the conventional one found in most introductory physics texts, and derived there using vector algebra, but it was shown here that it can also be derived using quaternion algebra. This $\vec{w} \times \vec{a}'$ solution, outlined in Figure 7, has fewer intermediate computations needed per iteration to update $\vec{a}'$ than the DCM^{-1} method; however, the numerical solution shows that more equations are needed to preserve the body's unit vector magnitude (Equation 203) such that it is essentially no more efficient than solving for $\vec{a}'$ using the DCM^{-1}.

Lastly, we showed that updating $\vec{a}'$ can be done using the traditional algorithm switching method; in this case, using the extrinsic DCM'^{-1} solution of Equation 162. As expected—and reassuringly—the extrinsic solution scheme produces the same solution for the body's rotation as a function of the gyro rates, with the same number of math operations as using the intrinsic DCM^{-1} formulation (Figures 10, 13, 16, 18–20).

In Section 9, we adjusted the time steps used in the numerical solutions to be more efficient for both vector and quaternion algebra methods. For the quaternion method we included a gyro rate dependence in the time step, which increased the ops/itr from 87 to 96. For the Euler angle method we added a pitch–angle-dependent scaling factor to the time step, which increase the ops/itr from 57 to 66. With these generalizations, the simulations showed that: 1) the Euler method would always be more efficient than the quaternion method, provided look-up tables are used to acquire every trig function value; 2) if linear interpolation of a look-up table was used to obtain trig values, then the ops/itr for the Euler method would increase from 66 to 80 ops/itr, and the simulations showed the efficiencies of the Euler and quaternion methods would generally be the same over the full range of angles; 3) if cubic or higher order interpolation of look-up table values was used, the number of ops/itr used in the Euler method would exceed that used for quaternions, and hence the quaternion method would always be more efficient.

The author believes the most novel aspect of this report is the demonstration that all three vector algebra-based solution methods (using the intrinsic or extrinsic Direction Cosine Matrices, as well as the $\vec{w} \times \vec{a}'$ based solution) can solve the rotating body problem without incurring a mathematical singularity. This was done by making the solution iteration time-step dependent on the cosine of the pitch angle (and thus cancelling out the singularity-causing division by the cosine of this angle; Equations 181 and 187). In addition, the report shows through numerical simulations that it is possible for vector algebra-based solutions to be more computationally efficient than the quaternion solution if look-up tables are used for trig and power law functions.

Example solutions are plotted over a 100-fold range of gyro rates (Figures 9–27, 29 and 30), showing both the angular change as well as the 3D body axes change with time (or iteration count). Confidence in the solution methodologies is gained by seeing that all four solution methods predict the same body axes trajectories.

11. References

[ANSI/AIAA] R-0004-1992. Recommended practice for atmospheric and space flight vehicle coordinate systems. American National Standards Institute/American Institute of Aeronautics and Astronautics; 1992. Appendix E, p. 57.

Baker MJ. Physics - kinematics - angular velocity ($\vec{W}$). EuclideanSpace - Mathematics and Computing; c2023a. https://www.euclideanspace.com/physics/kinematics/angularvelocity/

Baker MJ. Quaternion differentiation. EuclideanSpace - Mathematics and Computing; c2023b. https://www.euclideanspace.com/physics/kinematics /angularvelocity /QuaternionDifferentiation2.pdf

Battin RH. An introduction to mathematics and methods of astrodynamics. Revised ed. AIAA; 1999. p. 87.

Ben-Ari M. A tutorial on Euler angles and quaternions. Weizmann Institute of Science Teaching; 2014.

Challis JH. Quaternions as solution to determining the angular kinematics of human movement. BMC Biomed Eng. 2020;2(5).

Chudá H. Universal approach to derivation of quaternion rotation formulas. MATEC Web of Conferences (CSCC 2019). 2019;292:01060. p. 4.

Diebel J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Stanford University; 2006. p. 13.

Dykes J, Costello M, Fresconi F, Cooper G. Periodic projectile linear theory for aerodynamically asymmetric projectiles. J Aerospace Eng. 2013:228:11.

Etkin B. Dynamics of atmospheric flight. John Wiley and Sons; 1972. p. 126.

Fresconi F, Celmins I, Silton SI. Theory, guidance, and flight control for high maneuverability projectiles. Army Research Laboratory (US); 2014. Report No.: ARL-TR-6767. p. 5.

Graf B. Quaternions and dynamics. ArXiV. 2007 Feb. https://arxiv.org/pdf/0811.2889

Günasti G. Quaternions algebra, their applications in rotations and beyond quaternions [thesis]. 2016. p. 1. https://www.diva-portal.org

Henderson DM. Euler angles, quaternions, and transformation matrices for space shuttle analysis. National Aeronautics and Space Administration (US); 1977. Report No.: NASA-CR-151435.

Ilg MD. Guidance, navigation, and control for munitions [thesis]. Drexel University; 2008. p. 22.

Jia YB. Quaternions and rotations. Com S, 477/577 Notes; 2022.

Kang CW, Park CG. Euler angle based attitude estimation avoiding the singularity problem. Proc of the 18th World Congress, The International Federation of Automatic Control; 2011 Aug 28–Sept 2; Milano, Italy.

Kartiwa A, Supriatna AK, Rusyaman E, Sulaiman J. Review of quaternion differential equations: historical development, applications, and future direction. Axioms. 2023;12:483. p. 10. https://doi.org/10.3390/axioms12050483

Kim P. Rigid body dynamics for beginners: Euler angles and quaternions. CreateSpace Independent Publishing Platform; 2013.

Kuipers JB. Quaternions and rotation sequences: a primer with applications to orbits, aerospace and virtual reality. Princeton University Press; 1999. p. 207.

Kuipers JB. Quaternions and rotation sequences. International Conference on Geometry, Integrability and Quantization; Varna, Bulgaria. Coral Press; 2000. p. 128.

Narayan A. How to integrate quaternions. Ashwin Narayan; 2017 Sep 10. https://www.ashwinnarayan.com/post/how-to-integrate-quaternions/

Phillips WF, Bailey CE, Gerber GA. Review of attitude representations used for aircraft kinematics. J Aircraft. 2001;38(4).

Rogers RM. Applied mathematics in integrated navigation systems. 2nd ed. AIAA; 2003. p. 29.

Sabatini AM. Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis. Med Biol Eng Comput. 2005;43(1):2.

Schwab AL. Quaternions, finite rotation and Euler parameters. Laboratory for Engineering Mechanics, Delft University of Technology; 2002. p. 14. http://bicycle.tudelft.nl/schwab/Publications/quaternion.pdf

Singla P, Mortari D, Junkins JL. How to avoid singularity for Euler angle set? Adv Astronaut Sci. 2005;119:1409–1426.

Sonderegger JW. Vertical take-off and landing control via dual-quaternions and sliding mode [PhD dissertation and Master's theses]. Embry-Riddle Aeronautical University; 2022.

Ueno K, Ambar RB, Sagara S. Quaternion-based adaptive attitude control for a winged rocket. Proceedings of the 12th International Conference on Motion and Vibration Control; 2014.

Well KH, Weaver UA. Aircraft trajectory optimization using quaternions — comparison of a nonlinear programming and a multiple shooting approach. FAC Proceedings. 1984;17(2):1601.

Appendix A. Derivation of Quaternion Derivatives

Any and all of the quaternion derivative equations (Equations 112–115) can be derived using trig function identities and simplifications. For illustrative purposes, the derivation of $\dot{q}_k$ (Equation 115) will be given below. Differentiating q_k (Equation 94):

$$\begin{aligned}\dot{q}_k = -\left(\frac{1}{2}\right)[&\left\{sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+cos\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\dot{\phi}\\ &+\left\{cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)cos\left(\frac{\theta}{2}\right)\right\}\dot{\theta}\\ &-\left\{cos\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)sin\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\dot{\psi}\,]\end{aligned} \tag{A-1}$$

From Equation 106, we can solve for:

$$\dot{\phi} = \frac{wi\,cos(\psi)+wj\,sin(\psi)}{cos\,(\theta)} \tag{A-2}$$

$$\dot{\theta} = wj\,cos(\psi) - wi\,sin(\psi) \tag{A-3}$$

$$\dot{\psi} = wk + \frac{sin\,(\theta)}{cos\,(\theta)}[wi\,cos(\psi)+wj\,sin(\psi)] \tag{A-4}$$

and substitute these into Equation A-1,

$$\begin{aligned}\dot{q}_k = -\tfrac{1}{2}[&\left\{sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)sin\left(\tfrac{\psi}{2}\right)+cos\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\psi}{2}\right)sin\left(\tfrac{\theta}{2}\right)\right\}\left\{\frac{wi\,cos(\psi)+wj\,sin(\psi)}{cos\,(\theta)}\right\}\\ &+\left\{cos\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\theta}{2}\right)sin\left(\tfrac{\psi}{2}\right)+sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\psi}{2}\right)cos\left(\tfrac{\theta}{2}\right)\right\}\{wj\,cos(\psi)-wi\,sin(\psi)\}\\ &-\left\{cos\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)cos\left(\tfrac{\psi}{2}\right)+sin\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\psi}{2}\right)sin\left(\tfrac{\theta}{2}\right)\right\}\\ &\times\left\{wk+\frac{sin\,(\theta)}{cos\,(\theta)}[wi\,cos(\psi)+wj\,sin(\psi)]\right\}]\end{aligned} \tag{A-5}$$

Collecting just the w_j terms first, we have:

$$\begin{aligned}-\tfrac{1}{2}wj\,[&\left\{sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)sin\left(\tfrac{\psi}{2}\right)+cos\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\psi}{2}\right)sin\left(\tfrac{\theta}{2}\right)\right\}\left\{\frac{sin(\psi)}{cos\,(\theta)}\right\}\\ &-\left\{cos\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)cos\left(\tfrac{\psi}{2}\right)+sin\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\psi}{2}\right)sin\left(\tfrac{\theta}{2}\right)\right\}\left\{sin\,(\theta)\frac{sin(\psi)}{cos\,(\theta)}\right\}\\ &+\left\{cos\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\theta}{2}\right)sin\left(\tfrac{\psi}{2}\right)+sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\psi}{2}\right)cos\left(\tfrac{\theta}{2}\right)\right\}\{cos(\psi)\}\,]\end{aligned} \tag{A-6}$$

Substituting the half angle formula for $sin(\theta)$

$$sin(\theta) = 2\,sin\left(\tfrac{\theta}{2}\right)cos\left(\tfrac{\theta}{2}\right) \tag{A-7}$$

into the second line of A-6 and collecting terms:

$$-\tfrac{1}{2}wj\,[\begin{Bmatrix} sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)(1-2\,sin^2\left(\frac{\theta}{2}\right)) \\ +\,cos\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)(1-2\,cos^2\left(\frac{\theta}{2}\right)) \end{Bmatrix}\left\{\frac{sin(\psi)}{cos\,(\theta)}\right\}$$

$$+\left\{cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)cos\left(\frac{\theta}{2}\right)\right\}\{cos(\psi)\}\,] \tag{A-8}$$

Using the trig identities:

$$1-2\,sin^2\left(\tfrac{\theta}{2}\right)=\ cos\,(\theta) \tag{A-9}$$

$$1-2\,cos^2\left(\tfrac{\theta}{2}\right)=\ -cos\,(\theta) \tag{A-10}$$

in A-8 cancels the division by $cos\,(\theta)$, and we are left with

$$-\tfrac{1}{2}wj\,[\left\{sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)-cos\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\{sin(\psi)\}$$

$$+\left\{cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)cos\left(\frac{\theta}{2}\right)\right\}\{cos(\psi)\}\,] \tag{A-11}$$

Substituting the half angle formulas for $sin(\psi)$ and $cos(\psi)$ into A-11, factoring common functions, collecting and cancelling like terms, we arrive at:

$$-\frac{1}{2}wj\,[-sin\left(\frac{\psi}{2}\right)cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)\left(sin^2\left(\frac{\theta}{2}\right)+cos^2\left(\frac{\theta}{2}\right)\right)$$

$$+\,sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)cos\left(\tfrac{\psi}{2}\right)\left(sin^2\left(\tfrac{\theta}{2}\right)+cos^2\left(\tfrac{\theta}{2}\right)\right)\,] \tag{A-12}$$

Substituting

$$sin^2\left(\tfrac{\theta}{2}\right)+cos^2\left(\tfrac{\theta}{2}\right)=1 \tag{A-13}$$

into A-12, rearranging terms and recognizing, therein, the form of Equation 92 for q_i, we have for the wj term of A-5 for $\dot{q}_k$:

$$-\tfrac{1}{2}wj\,\left\{sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\psi}{2}\right)cos\left(\tfrac{\theta}{2}\right)-cos\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\psi}{2}\right)sin\left(\tfrac{\theta}{2}\right)\right\}=-\tfrac{1}{2}wjq_i \tag{A-14}$$

Similarly, if we collect the wi of A-5, we have:

$$-\frac{1}{2}wi\,[\left\{sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+cos\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\left\{\frac{cos(\psi)}{cos\,(\theta)}\right\}$$

$$-\left\{cos\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)sin\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\left\{sin\,(\theta)\frac{cos(\psi)}{cos\,(\theta)}\right\}$$

$$+\left\{cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)cos\left(\frac{\theta}{2}\right)\right\}\{-sin(\psi)\}\,] \tag{A-15}$$

Following the same trig substitutions and collection of terms, A-15 becomes:

$$-\tfrac{1}{2}wi\,[\left\{sin\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)-cos\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\}\{cos(\psi)\}$$

$$-\left\{cos\left(\frac{\phi}{2}\right)sin\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right)+sin\left(\frac{\phi}{2}\right)cos\left(\frac{\psi}{2}\right)cos\left(\frac{\theta}{2}\right)\right\}\{sin(\psi)\}\,] \tag{A-16}$$

Similarly, substituting the half angle formulas for $sin(\psi)$ and $cos(\psi)$ into A-16, employing identities, simplifying, rearranging terms and comparing with Equation 93, we find A-16, for the $w_i$term of $\dot{q}_k$, is identical to:

$$-\tfrac{1}{2}wi\left\{-cos\left(\tfrac{\phi}{2}\right)sin\left(\tfrac{\theta}{2}\right)cos\left(\tfrac{\psi}{2}\right) - sin\left(\tfrac{\phi}{2}\right)cos\left(\tfrac{\theta}{2}\right)sin\left(\tfrac{\psi}{2}\right)\right\} = \tfrac{1}{2}wiq_j \qquad \text{(A-17)}$$

Lastly, if we look at the wk term of A-5, for $\dot{q}_k$, and Equation 91 for q_o, we see that term will equate to:

$$\tfrac{1}{2}wk\ \left\{cos\left(\frac{\phi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) + sin\left(\frac{\phi}{2}\right)sin\left(\frac{\psi}{2}\right)sin\left(\frac{\theta}{2}\right)\right\} = \tfrac{1}{2}wkq_o \qquad \text{(A-18)}$$

Thus, summing up the results of A-1 to A-5, reaching A-14, A-17 and finally A-18, we have derived Equation 115, namely:

$$\dot{q}_k = -\tfrac{1}{2}wjq_i + \tfrac{1}{2}wiq_j + \tfrac{1}{2}wkq_o = -\,(1/2)(\ wjq_i - wiq_j - wkq_o) \qquad \text{(A-19)}$$

Again, this Appendix showed the derivation of Equation 115 for $\dot{q}_k$, the other quaternion derivates in Equations 112–114, are derived in a similar fashion.

Appendix B. Alternative Derivation of $\dot{q}q^{-1}$

A slightly different way to derive the relationship of Equation 143 ($\dot{q}q^{-1} = (1/2)\vec{w}$) is shown below.

First, suppose we return to the original expression for q, as used in Equation 90, namely

$$q(\phi, \theta, \psi) = q_\phi \, q_\theta \, q_\psi \tag{B-1}$$

Differentiating Equation C-1 we get:

$$\dot{q} = \frac{dq_\phi}{dt} q_\theta q_\psi + q_\phi \frac{dq_\theta}{dt} q_\psi + q_\phi q_\theta \frac{dq_\psi}{dt} \tag{B-2}$$

and

$$q^{-1} = (q_\phi \, q_\theta \, q_\psi)^{-1} = q_\psi^{-1} q_\theta^{-1} q_\phi^{-1} \tag{B-3}$$

(note, in order that $qq^{-1} = 1$, the inverse of the product of quaternion components must be reversed; such that, it is the reverse product of inverse components.)

Hence, using Equation B-2 and B-3,

$$\begin{aligned} \dot{q}q^{-1} = {} & \frac{dq_\phi}{dt} q_\theta q_\psi (q_\psi^{-1} q_\theta^{-1} q_\phi^{-1}) \\ & + q_\phi \frac{dq_\theta}{dt} q_\psi (q_\psi^{-1} q_\theta^{-1} q_\phi^{-1}) + q_\phi q_\theta \frac{dq_\psi}{dt} (q_\psi^{-1} q_\theta^{-1} q_\phi^{-1}) \\ = {} & [\frac{dq_\phi}{dt} q_\phi^{-1}] + q_\phi [\frac{dq_\theta}{dt} q_\theta^{-1}] q_\phi^{-1} + q_\phi q_\theta [\frac{dq_\psi}{dt} q_\psi^{-1}] q_\theta^{-1} q_\phi^{-1} \end{aligned} \tag{B-4}$$

If we individually differentiate q_ϕ, q_θ, q_ψ, as given by Equations 84, 87, and 89, and perform the geometric product (or complex number) multiplications inside the square brackets of Equation B-4, followed by completing any subsequent rotations in which the square brackets are nested, we will find that

$$\begin{aligned} \dot{q}q^{-1} = {} & (1/2)([\, \dot{\phi} \, cos(\theta) cos(\psi) - \dot{\theta} \, sin(\psi) \,] \, i \\ & + [\, \dot{\theta} \, cos(\psi) + \dot{\phi} \, cos(\theta) sin(\psi) \,] \, j + [\, \dot{\psi} - \dot{\phi} \, sin(\theta)] \, k \,) \end{aligned} \tag{B-5}$$

and usings Equations 103–105 in Equation B-5 nets:

$$\dot{q}q^{-1} = (1/2)(\overrightarrow{wi} + \overrightarrow{wj} + \overrightarrow{wk} \,) \tag{B-6}$$

which is another approach to arrive at Equation 143.

Appendix C. Other Formulations for $\frac{d\vec{a}'}{dt}$

In addition, there are other formulations for $\frac{d\vec{a}\prime}{dt}$ than that given in Equation 145, though none of them are as computationally efficient.

For example, suppose we tried to find $\frac{d\vec{a}\prime}{dt}$ by time differentiating the q matrix (Equation 82) directly. On the left of Equation 82 are the coefficients of $\vec{a}'$, while the right equates those coefficients to the product of q matrix with the $\vec{a}$ matrix. Since $\vec{a}$ is a fixed vector in the body frame, time-differentiating the right would equate to time differentiating only the elements of q matrix, this is a lengthy process but it would yield:

$$\begin{bmatrix} \dot{a}_i' \\ \dot{a}_j' \\ \dot{a}_k' \end{bmatrix} = 2\begin{bmatrix} q_o\dot{q}_o + q_i\dot{q}_i - q_j\dot{q}_j - q_k\dot{q}_k & \dot{q}_jq_i + q_j\dot{q}_i - \dot{q}_oq_k - q_o\dot{q}_k & \dot{q}_oq_j + q_o\dot{q}_j + \dot{q}_kq_i + q_k\dot{q}_i \\ \dot{q}_oq_k + q_o\dot{q}_k + \dot{q}_jq_i + q_j\dot{q}_i & q_o\dot{q}_o + q_j\dot{q}_j - q_i\dot{q}_i - q_k\dot{q}_k & \dot{q}_kq_j + q_k\dot{q}_j - \dot{q}_oq_i - q_o\dot{q}_i \\ \dot{q}_kq_i + q_k\dot{q}_i - \dot{q}_oq_j - q_o\dot{q}_j & \dot{q}_oq_i + q_o\dot{q}_i + \dot{q}_jq_k + q_j\dot{q}_k & q_o\dot{q}_o + q_k\dot{q}_k - q_i\dot{q}_i - q_j\dot{q}_j \end{bmatrix}\begin{bmatrix} a_i \\ a_j \\ a_k \end{bmatrix} \quad \text{(C-1)}$$

In theory then, and to review, we would need to obtain new ψ, θ, and ϕ values from $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$ (Equation 47) multiplied by a time interval Δt, and added to the old/current ψ, θ, and ϕ values, which as we have enumerated previously (following Equation 98) would require 15 math operations. These new ψ, θ, and ϕ values are then substituted into Equations 91–94 to update the q components (14 operations) and substituted into Equation 107 to update the $\vec{w}$ components (33 operations), both of which are needed to evaluate the $\dot{q}$ components using Equations 112–115 (which requires 24 math operations). The q and $\dot{q}$ components can then be substituted into the C-1 matrix, requiring 63 math operations, and lastly, multiplying that 3 × 3 matrix times the 3 × 1 $\vec{a}$ matrix would add another 9 multiplications and 6 additions. In total then it would take 164 (= 15 + 14 + + 33 + 24 + 63 +9 + 6) operations to evaluate the left of Equation C-1, for $\frac{d\vec{a}\prime}{dt}$. Once that is determined there would be an additional 6 operations needed to combine $\frac{d\vec{a}\prime}{dt}\Delta t$ with the existing $\vec{a}'$; the grand total would thus be 170 math operations; clearly a more costly calculation than the 48 operations needed to update $\vec{a}'$ using the $\vec{w} \times \vec{a}'$ formulation of Equation 145.

The final digression is yet another way to evaluate the time derivative of $\vec{a}'$ by time-differentiating the DCM^{-1} matrix expression for $\vec{a}'$. Doing so, yields:

$$\begin{bmatrix} \dot{a}'_i \\ \\ \dot{a}'_j \\ \\ \dot{a}'_k \end{bmatrix} = \begin{bmatrix} -c_\theta s_\psi \dot{\psi} - s_\theta c_\psi \dot{\theta} & \begin{matrix} \{-s_\phi s_\theta s_\psi - c_\psi c_\phi\}\dot{\psi} \\ + \{s_\phi c_\theta c_\psi\}\dot{\theta} \\ + \{c_\phi s_\theta c_\psi + s_\psi s_\phi\}\dot{\phi} \end{matrix} & \begin{matrix} \{-c_\phi s_\theta s_\psi + c_\psi s_\phi\}\dot{\psi} \\ + \{c_\phi c_\theta c_\psi\}\dot{\theta} \\ + \{-s_\phi s_\theta c_\psi + s_\psi c_\phi\}\dot{\phi} \end{matrix} \\ \\ c_\theta c_\psi \dot{\psi} - s_\theta s_\psi \dot{\theta} & \begin{matrix} \{s_\phi s_\theta c_\psi - s_\psi c_\phi\}\dot{\psi} \\ + \{s_\phi c_\theta s_\psi\}\dot{\theta} \\ + \{c_\phi s_\theta s_\psi - c_\psi s_\phi\}\dot{\phi} \end{matrix} & \begin{matrix} \{c_\phi s_\theta c_\psi + s_\psi s_\phi\}\dot{\psi} \\ + \{c_\phi c_\theta s_\psi\}\dot{\theta} \\ + \{-s_\phi s_\theta s_\psi - c_\psi c_\phi\}\dot{\phi} \end{matrix} \\ \\ -c_\theta \dot{\theta} & -s_\theta s_\phi \dot{\theta} + c_\theta c_\phi \dot{\phi} & -s_\theta c_\phi \dot{\theta} - c_\theta s_\phi \dot{\phi} \end{bmatrix} \begin{bmatrix} a_i \\ \\ a_j \\ \\ a_k \end{bmatrix} \tag{C-2}$$

If we break down the number of operations in Equation C-2, there are 13 independent trig-function 2-tuples and 8 such 3-tuples, hence 21 multiplications (plus the one singlet). There are 8 add/subtracts within the coefficients of the angular rates and 21 multiplications of those coefficients with those angular rates. When we sum together all the coefficient-multiplied angular rates in each element of the 3 × 3 matrix it adds another 12 operations, so that the total number of operations just to fully expand out the matrix would be 62 operations (= 21 + 8 + 21 + 12). When we multiply the 3 × 3 with the 3×1 $\vec{a}$ matrix, it requires 9 multiplications and 6 adds. Then again, to update $\vec{a}'$ we need the 6 operations involved with combining $\frac{d\vec{a}'}{dt}\Delta t$ with the existing $\vec{a}'$. Hence, using this third approach requires 81 (= 62 + 9 + 6 + 6) operations, which is less than the previous 170 using Equation C-1, but still nearly twice the number (48) needed if we use either $\vec{w} \times \vec{a}'$ formulation of Equation 145 or DCM^{-1} formulation of Equation 95.

List of Symbols, Abbreviations, and Acronyms

ψ (ψ')	Rotation in radians of a body about the earth's *z*-axis, called yaw (for both intrinsic and extrinsic rotation analysis)
θ (θ')	For an intrinsic rotation analysis: rotation in radians of a body about the body's *y*-axis (after rotation of the body about the earth's *z*-axis), called pitch (or, with a prime post-superscript for extrinsic rotation analysis: rotation about the earth's *y*-axes after rotation about the earth's *z*-axis)
ϕ (ϕ')	For an intrinsic rotation analysis: rotation in radians of a body about the body's *x*-axis (when preceded by the body's rotation about the earth's *z*-axis, then about the body's *y*-axis), called roll (or, with a prime post-superscript for extrinsic rotation analysis: rotation about the earth's *x*-axes, after rotation about the earth's *z*, then *y* axes, in that order)
Δt	Time interval (or change in time)
$\dot{\square}$	Time derivative of any variable inside dashed box
3/4D	Three-/four-dimensional
ARL	Army Research Laboratory
CCW	Counterclockwise
CW	Clockwise
DCM(DCM')	Direction Cosine Matrix for intrinsic (extrinsic) axes of rotation
DCM^{-1}(DCM'^{-1})	Inverse DCM for intrinsic (extrinsic) axes of rotation
DEVCOM	U.S. Army Combat Capabilities Development Command
ee1	Unit vector along the earth's *x*-axis
ee2	Unit vector along the earth's *y*-axis
ee3	Unit vector along the earth's *z*-axis
eb1	Unit vector along the rotating body's *x*-axis
eb2	Unit vector along the rotating body's *y*-axis
eb3	Unit vector along the rotating body's *z*-axis

N_E	Total number of body revolutions calculated by an Euler-Quat.–based solution over *s* iterations.
N_Q	Total number of body revolutions by a quaternion-based solution over *s* iterations.
P ≡ *wb1*	Gyro rate measured about the body's *x*-axis, or angular rate about the body's *eb1* axis
Q ≡ *wb2*	Gyro rate measured about the body's *y*-axis, or angular rate about the body's *eb2* axis
$q(\phi,\theta,\psi)$ = $q_\phi q_\theta q_\psi$ axes	Composite quaternion for cumulative rotation about intrinsic
$q_{\phi(\phi'),\ \theta(\theta'),\ \psi(\psi')}$	Quaternions for rotations about individual intrinsic (extrinsic) axes
$q_{o,i,j,k}$	Quaternion coefficients for $q(\phi,\theta,\psi)$
$q'(\phi',\theta',\psi')$ $= q'_{\phi'}q'_{\theta'}q'_{\psi'}$ axes	Composite quaternion for cumulative rotation about extrinsic
$q'_{o,i,j,k}$	Quaternion coefficients for $q'(\phi',\theta',\psi')$
R ≡ *wb3*	Gyro rate measured about the body's *z*-axis, or angular rate about the body's *eb3* axis
rad.	Radians
s	Number of numerical iterations performed for a given body-rotation solution
t	Time
trig	Trigonometry
wi ≡ *wbi*	Angular rate of the body projected onto the earth's *ee1* axis
wj ≡ *wbj*	Angular rate of the body projected onto the earth's *ee2* axis
wk ≡ *wbk*	Angular rate of the body projected onto the earth's *ee3* axis

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB