

# Quaternion to Euler Angle Conversion for Arbitrary Rotation Sequence Using Geometric Methods

Noel H. Hughes\*

*Braxton Technologies, Colorado Springs, Colorado, 80915*

## Nomenclature

$\hat{e}$	= normalized Euler rotation axis
$i<1-3>$	= indices of first, second and third Euler rotation, e.g. if rotation sequence is 3-2-1, i1 is 3, i2 is 2, and i3 is 1
$i<1-3>_n$	= next circular index following $i<1-3>$ , e.g. if $i2 = 3$ , $i2_n = 1$
$i<1-3>_{nn}$	= next circular index following $i<1-3>_n$ , e.g. if $i2_n = 3$ , $i2_{nn} = 2$
$Q$	= Quaternion = $q_1 q_2 q_3 q_4$
$R$	= operator: rotation of vector by a quaternion
$v_i$	= unit vector along the $i^{\text{th}}$ Euler rotation axis
$v_{i_n}$	= unit vector along the $i_n^{\text{th}}$ Euler rotation axis
$V$	= a general 3 component vector; if used in quaternion multiplication, augmented with a fourth element equal to zero
$X$	= vector cross product operator
$\theta_{<1-3>}$	= Euler angles
$\phi$	= Euler rotation angle
$\otimes$	= quaternion multiplication
*	= superscript operator: conjugate
•	= vector dot product operator

## Introduction

Conversion of a set of Euler angles, using any arbitrary rotation order, to the equivalent quaternion is a simple exercise, consisting of generating the three quaternions corresponding to the three Euler angles and performing the quaternion multiplications in the appropriate order. Conversion of a quaternion to the equivalent Euler angles is a somewhat more complex task. While no literature survey can be all inclusive, all references to this conversion process found recently, references [1-3] are typical, perform the conversion by generating the rotation matrix (the direction cosine matrix) both from quaternions and from the Euler angles for a particular rotation sequence, equating similar matrix elements and solving for the Euler angles in terms of the quaternion elements. This process is algebraically complex, requires a different formulation for each of the twelve rotation sequences and is mathematically inelegant.

This technical note presents a method of converting a quaternion to the Euler angles for any specified rotation sequence utilizing geometric methods. The method requires only minor logical differences for repeated and non-repeated rotation sequences and for circular and non-circular order sequences. Also, in support of the method,

---

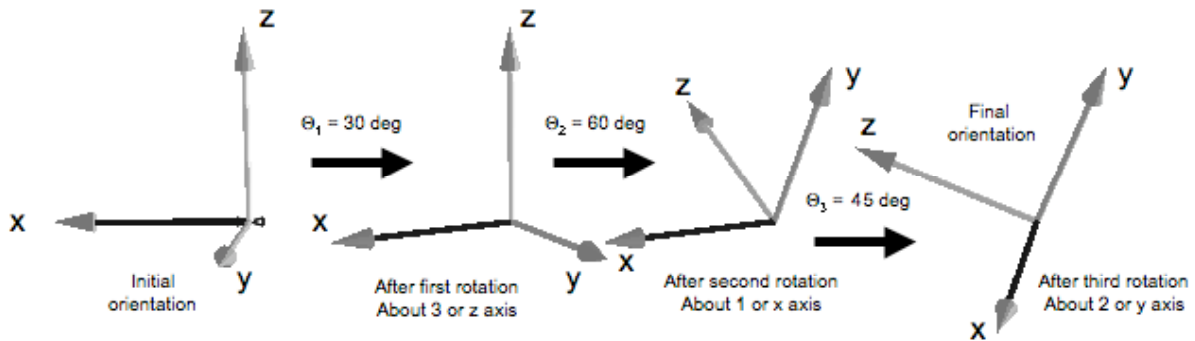
\* Senior Aerospace Engineer, Braxton Technologies, 770 Wooten Rd., Suite 105, Colorado Springs, CO 80915  
AIAA Member

several definitions and fundamental quaternion and vector mathematical constructs are presented without derivation or proof. A single numerical example is included to support understanding of the method.

## Definitions and Mathematical Constructs

### Euler Angles

Leonard Euler was one of the giants of 18<sup>th</sup> century mathematics. Many developments are attributed to him including several designated as Euler’s Theorem. The one of interest here states “Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis” [2]. The angles of these three rotations are commonly defined as Euler angles and the axes of rotation designated as axes 1, 2, and 3 or x, y, and z. The order in which the axes of rotation are taken is referred to as the Euler rotation sequence; there are twelve of these sequences: 1-2-3 (x, y, z), 1-2-1 (x, y, x) and so on including all combinations with no two succeeding rotations about the same axis. Figure 1 depicts, as an example, a 3-1-2, or zxy rotation sequence where the z rotation Euler angle is 30 deg, the x angle is 45 degrees and the y angle is 45 degrees.



**Fig. 1 Euler Rotation Example: 3-1-2 or zxy Euler Rotation Sequence**

**Euler angles = 30, 60, 45 degrees**

For this example:

$$i1 = 3 \tag{1}$$

$$i2 = 1 \tag{1a}$$

$$i3 = 2 \tag{1b}$$

Euler angles as applied to the Maritime and Aerospace industries are often called the roll or bank angle, about the x axis, the pitch or attitude angle, about the y axis, and the yaw or heading angle, about the z axis; the reference, or initial, coordinate frame is frequently z axis positive down, x axis horizontal North and the y axis located to form a right handed coordinate frame. As such, they provide a certain level of intuitive understanding; however, they also have two inherent disadvantages:

1) Ambiguity - For small values of Euler angles the Euler Rotation Sequence may not be important. However, for large angles, the rotation sequence becomes critical; for example, for a given set of three Euler angles, the result of a 1-2-3 rotation sequence is very different from that of a 3-2-1 sequence. There is no industry accepted standard rotation sequence; thus, there is an inherent risk of mistaken assumption of rotation sequence in performing analysis and communicating using Euler angles.

2) Singularities - Any set of Euler angles where the second rotation aligns the axes of the first and third rotations causes a singularity. For an Euler Rotation Sequence where the first and third axes are the same, called a repeated axis sequence, singularities occur for second rotation angles of zero and 180 degrees; for non-repeated axis sequences singularities occur at +/- 90 degrees. The mechanical manifestation of this mathematical singularity is the dreaded "gimbal lock" of the stable reference platform made infamous in the movie *Apollo 13*. At a singularity a number of potentially disastrous effects occur, including: the first and third rotations degenerate into a single rotation and the angular derivatives, or equations of motion, become infinite. This was not a serious problem for mariners and most aeronauts before the advent of the space age; if a ship had achieved a pitch angle of 90 degrees, the fact that its Euler angle derivatives had become infinite was probably of little concern to the crew. However, since, in general, all attitudes are equally likely for spacecraft, Euler angles do not lend themselves well to analysis applied to astronautics or to highly maneuverable aircraft.

## **Quaternions**

Another of Euler's Theorems states: the most general displacement of a rigid body with one point fixed is a rotation about some axis [1]. Thus any rotation of a rigid body can be described by defining an axis of rotation, often called the Euler rotation axis, and a rotation angle, the Euler rotation angle. In October of 1843, William Rowan Hamilton formulated quaternions [3], utilizing this Euler's theorem, as a method of representing rotations. A common, although not universal, definition of a quaternion is

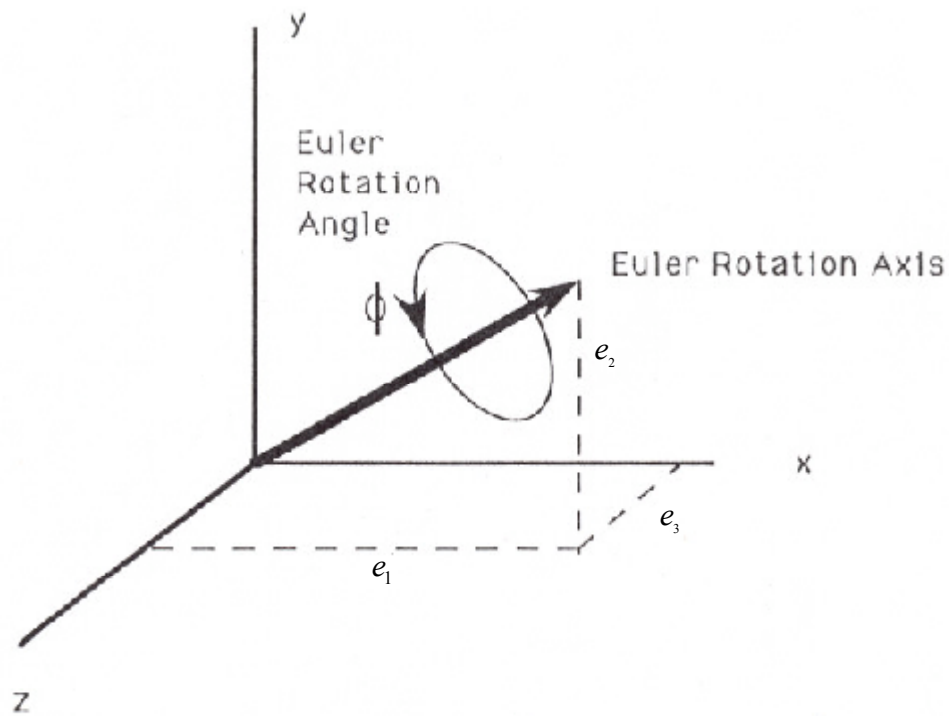
$$Q = \hat{e} \sin(\phi/2) \quad \cos(\phi/2) = e_1 \sin(\phi/2) e_2 \sin(\phi/2) e_3 \sin(\phi/2) \cos(\phi/2) \quad (2)$$

$$= q_1 q_2 q_3 q_4 \quad (3)$$

$$= q_1 i q_2 j q_3 k q_4 \quad (4)$$

The four elements are not independent, being related by the requirement that  $|Q| = 1$

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (5)$$



**Fig. 2 Euler Rotation Axis and Angle**

Figure 2 graphically depicts the Euler rotation axis and angle (not to be confused with the Euler angles).

It is interesting to note that the Euler rotation axis is unchanged by any quaternion rotation about this axis and is, in general, the only vector that is unchanged.

For the example depicted in Fig. 1:

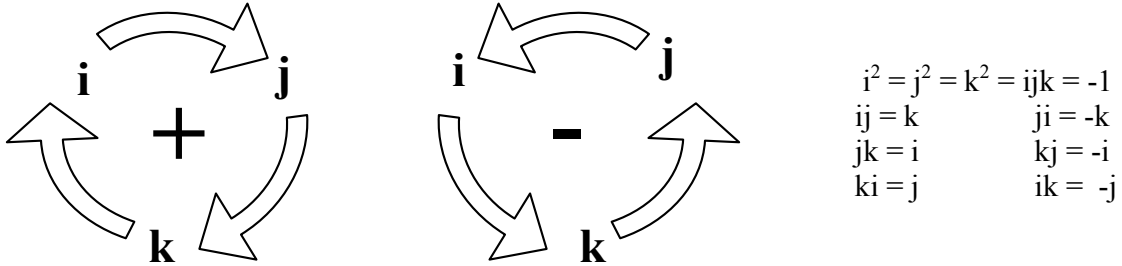
$$Q = 0.360423 \quad 0.439679 \quad 0.391904 \quad 0.723317 \quad (6)$$

$$\hat{e} = 0.5220 \quad 0.6367 \quad 0.5676 \quad (7)$$

and

$$\phi = 43.671 \tag{8}$$

### Quaternion Multiplication



**Fig. 3 Quaternion Multiplication Rules**

Quaternion multiplication is commonly performed using the form shown in Eq. (4), above. The elements in each quaternion are multiplied by all elements in the other utilizing the rules shown, above, in Figure 3 which depicts the rules for quaternion multiplication both graphically and using hyper complex notation.

Quaternion multiplication is associative, that is,

$$Q_a \otimes Q_b \otimes Q_c = (Q_a \otimes Q_b) \otimes Q_c = Q_a \otimes (Q_b \otimes Q_c) \tag{9}$$

Quaternion multiplication is not, in general, commutative, so

$$Q_a \otimes Q_b \neq Q_b \otimes Q_a \tag{10}$$

### Quaternion Identity, and Inverse or Conjugate

The quaternion multiplicative identity is simply

$$Q_I = 0.0 \ 0.0 \ 0.0 \ 1.0 \tag{11}$$

and

$$Q \otimes Q_I = Q_I \otimes Q = Q \tag{12}$$

The multiplicative inverse, or conjugate, of a quaternion is

$$Q^{-1} = Q^* = -q_1 - q_2 - q_3 q_4 = q_1 q_2 q_3 - q_4 \quad (13)$$

and

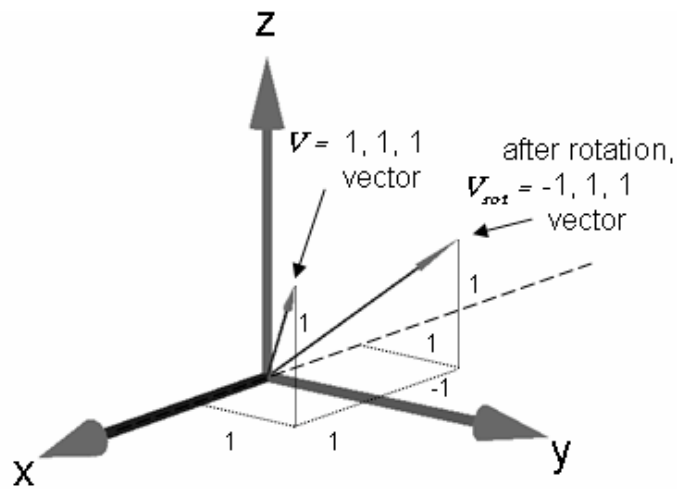
$$Q \otimes Q^* = Q^* \otimes Q = Q_I \quad (14)$$

Multiplication of a quaternion by its conjugate and by the identity quaternion are among the few multiplicative operations that are commutative.

### Rotating a Vector

A vector,  $V$ , can be rotated about a given axis by a given angle using the quaternion,  $Q$ , generated using that axis and angle:

$$V_{rot} = Q R V = Q \otimes V \otimes Q^* \quad (15)$$



**Fig. 4 Vector before and after rotation**

As an example, Fig. 4 graphically depicts the vector 1, 1, 1 before and after being rotated by the quaternion 0.0 0.0 0.707107 0.707107 which represents a 90 degree rotation about the Z, or 3, axis; the resultant rotated vector is -1, 1, 1.

Interestingly, transformation of a vector from one coordinate frame to another is similar to the rotation operation except that the conjugate of the quaternion is used; the transformation operation is not used in the present technical note.

## Quaternion to Euler Angle Conversion

### Fundamental Concept

The underlying concept of the present method of converting a given quaternion,  $Q_G$ , to the corresponding Euler angles of the given rotation sequence,  $i_1 i_2 i_3$ , is to perform the quaternion rotation operations on selected unit vectors and derive the required Euler angles from the results.

### Calculating the First and Second Euler angles

The first step in the current method is performing the quaternion rotation operation on a unit vector along the third, or  $i_3$ , Euler rotation axis.

$$v_{3rot} = Q_G R v_3 \quad (16)$$

Returning to the example shown in Fig. 1 and starting with the quaternion equivalent to the Euler rotations:

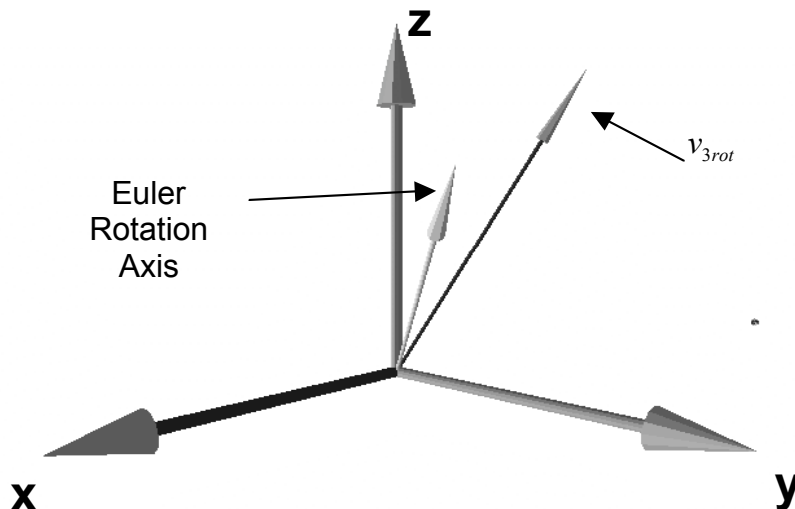
$$Q_G = 0.360423 \ 0.439679 \ 0.391904 \ 0.723317 \quad (17)$$

and, for the given rotation sequence 3-1-2

$$v_3 = 0 \ 1 \ 0 \quad (18)$$

Performing the operation of Eq. (15)

$$v_{3rot} = -0.2500 \ 0.4330 \ 0.8660 \quad (19)$$



**Fig. 5 Euler Rotation Axis and Rotated Third Euler Rotation Axis,  $v\mathfrak{3}_{rot}$**

Since the position of vector  $v\mathfrak{3}_{rot}$  is unaffected by the third Euler angle, the first two Euler angle can be determined from this vector. Visual inspection of Fig. 5 indicates that the first Euler angle,  $\theta_1$ , the rotation about the  $i1$  or  $z$  axis, is the four quadrant inverse tangent of  $-v\mathfrak{3}_{rot}(1)$  and  $v\mathfrak{3}_{rot}(2)$ , or  $-v\mathfrak{3}_{rot}(i1_n)$  and  $v\mathfrak{3}_{rot}(i1_{nn})$ , restricted to  $\pm 180$  deg, and that the second Euler angle,  $\theta_2$ , the rotation about the  $i2$  or  $y$  axis, is the inverse sine of  $v\mathfrak{3}_{rot}(3)$ , or  $v\mathfrak{3}_{rot}(i1)$ . Performing these two operations using  $v\mathfrak{3}_{rot}$  from the example returns

$$\theta_1 = 30 \text{ degrees} \quad (20)$$

$$\theta_2 = 60 \text{ degrees} \quad (21)$$

This process is generalized via a similar development to one of four similar processes, depending on whether the Euler rotation sequence has the first and third rotation about the same axis, a repeated axis sequence, and whether the first and second rotation axes are in circular order, that is, 12, 23, or 31.

*Non-circular, Repeated Axis*

This case includes the three Euler rotation sequences 131, 212, and 323.

$$\theta_1 = \tan^{-1} (v\mathfrak{3}_{rot}(i1_{nn}), v\mathfrak{3}_{rot}(i1_n)) \quad (22)$$

$$\theta_2 = \cos^{-1} (v\mathfrak{3}_{rot}(i1)) \quad (23)$$

*Non-circular, Non-repeated axis*

This case includes the three Euler rotation sequences 132, 213, and 321.

$$\theta_1 = \tan^{-1} (v\mathfrak{3}_{rot}(i1_{nn}), v\mathfrak{3}_{rot}(i1_n)) \quad (24)$$

$$\theta_2 = -\sin^{-1} (v\mathfrak{3}_{rot}(i1)) \quad (25)$$

*Circular, Repeated Axis*

This case includes the three Euler rotation sequences 121, 232, and 313.

$$\theta_1 = \tan^{-1} (v\mathfrak{3}_{rot}(i1_n), -v\mathfrak{3}_{rot}(i1_{nn})) \quad (26)$$

$$\theta_2 = \cos^{-1} (v\mathfrak{3}_{rot}(i1)) \quad (27)$$



### *Circular, Non-repeated Axis*

This case includes the three Euler rotation sequences 123, 231, and 312.

$$\theta_1 = \tan^{-1} (-v3_{rot}(iI_n), v3_{rot}(iI_{nn})) \quad (28)$$

$$\theta_2 = -\sin^{-1}(v3_{rot}(iI)) \quad (29)$$

### **Calculating the Third Euler Angle**

Having calculated the first two Euler angles,  $\theta_1$  and  $\theta_2$ , the third is calculated by generating two vectors whose only difference is that effected by the third rotation and deriving the included angle between the two; this is effected via the following procedure:

#### *Generate Quaternion from First Two Euler Angles*

First generate the two quaternions representing the two Euler angles. Using the nomenclature of Eqs. (2)

$$Q_1 = 0.0 \ 0.0 \ 0.0 \ \cos(\theta_1/2) \quad (30)$$

except

$$q_{i1} = \sin(\theta_1/2) \quad (31)$$

and

$$Q_2 = 0.0 \ 0.0 \ 0.0 \ \cos(\theta_2/2) \quad (32)$$

except

$$q_{i2} = \sin(\theta_2/2) \quad (33)$$

Then

$$Q_{12} = Q_1 \otimes Q_2 \quad (34)$$

*Generate the unit vector along the  $i3_n$  axis.*

$$v3_n = 0.0 \ 0.0 \ 0.0 \quad (35)$$

except

$$v3_n(i3_n) = 1.0 \quad (36)$$

#### *Perform Quaternion Rotations*

Perform the quaternion rotation operation on  $v3_n$  twice, using both  $Q_{12}$  and  $Q_G$ .

$$v3_{n12} = Q_{12} \otimes v3_n \quad (37)$$

$$v3_{nG} = Q_G \otimes v3_n \quad (38)$$

*Calculate the Included Angle*

The magnitude of the third Euler angle is then the angle between these two vectors.

$$|\theta_3| = \cos^{-1}(v3_{n12} \cdot v3_{nG}) \quad (39)$$

The sign, or sense, of  $\theta_3$  is determined by performing the cross product of  $v3_{n12}$  and  $v3_{nG}$  and determining if this vector is parallel or anti-parallel to  $v3_{rot}$ .

$$v_c = v3_{n12} \times v3_{nG} \quad (40)$$

$$m = v_c \cdot v3_{rot} \quad (41)$$

and finally

$$\theta_3 = \text{sign}(m) |\theta_3| \quad (42)$$

Returning to the example of Fig. 1, using Eqs. (29-41)

$$Q_1 = 0.0 \ 0.0 \ 0.2588 \ 0.9659 \quad (43)$$

$$Q_2 = 0.5 \ 0.0 \ 0.0 \ 0.8660 \quad (44)$$

$$Q_{12} = 0.4830 \ 0.1294 \ 0.2241 \ 0.8365 \quad (45)$$

$$v3_n = 0.0 \ 0.0 \ 1.0 \quad (46)$$

$$v3_{n12} = 0.4330 \ -0.7500 \ 0.5000 \quad (47)$$

$$v3_{nG} = 0.9186 \ -0.1768 \ 0.3536 \quad (48)$$

$$v3_{n12} \cdot v3_{nG} = 0.7071 \quad (49)$$

$$|\theta_3| = 45 \text{ degrees} \quad (50)$$

$$v3_{n12} \times v3_{n12} = -0.1768 \ 0.3062 \ 0.6124 \quad (51)$$

$$\theta_3 = 45 \text{ degrees} \quad (52)$$

This agrees with the original Euler angles from the example.

## **Implementation and Testing**

The author implemented the present method in a MATLAB<sup>†</sup> function as well as a companion function that converts a set of three Euler angles and a specified Euler rotation sequence to the equivalent quaternion. A driver routine was then developed which executed the two functions sequentially and then compared the quaternion input to the first function to the quaternion output of the second. Each input quaternion was used to generate all twelve sets of Euler rotation sequence angles; each of these twelve sets of Euler angles was then converted back to the equivalent quaternion and compared to the original quaternion. The comparison was effected by multiplying the input quaternion by the conjugate of the output quaternion and summing the absolute values of the first three elements of the resultant quaternion; the comparison was considered successful if this sum was less than  $10^{-6}$  implying that the two quaternions were a very close match. This test system was exercised in two modes as described, below; in addition, a small number of resultant test cases were then graphically modeled using SOAP (Satellite Orbit Analysis Program).

#### *Selected Rotation Test Mode*

A series of tests were run wherein a set of Euler angles was selected with the second angle at a singular point for each Euler rotation sequence and the first and third angles selected randomly between  $\pm 180$ ; the input test quaternion was then generated from these Euler angles. Comparison was made between both the selected and output Euler angles and the input and output quaternion.

#### *Random Rotation Test Mode*

Another set of tests was run with quaternions generated by a software routine that, for each test case, selected four numbers at random between  $\pm 1.0$ , then normalized the set of four. Several hundred cases were generated by this method and tested.

#### *SOAP Testing*

A small number of cases were input to SOAP and the results verified visually. Two identical solid models were created, one based on a quaternion coordinate frame, the other on three Euler angle rotations. The quaternion and Euler angles from several test cases were input and the orientation of the two bodies compared. Only a small number of cases were implemented because of the time involved in manually entering data.

#### *Testing Results*

Without exception, all test cases compared successfully in that the input and output quaternions matched. In the selected rotation test mode, as might be expected, the first and third Euler angles did not always match those of the

---

<sup>†</sup> MATLAB is a registered trademark of The Mathworks

original Euler angles used to generate the input quaternion. However, in those cases where they did not match, the sum of the first and third resultant Euler angles matched that of the original Euler angle set.

### **Conclusion**

Even within the technical community, quaternions have acquired the reputation of being difficult to understand. The method presented in this technical note may serve to increase the level of understanding within the community and will provide another avenue for converting quaternions, which lend themselves well to various aerospace analytical tasks, to Euler angles, with which the community may be more familiar, but which increase the difficulty and risk of many types of analysis.

### **References**

- [1] Kuipers, Jack B., *Quaternions and Rotation Sequences A Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press, Princeton, NJ, 1999, pp., 168, 83
- [2] Amoruso, Michael J., "Euler Angles and Quaternions in Six Degree of Freedom Simulations of Projectiles", US Army Armament, Munitions and Chemicals Command, Picatinny Arsenal, New Jersey, March, 1996, p. 35
- [3] Dam, Erik B., Koch, Martin, Lillholm, Martin, "Quaternions, Interpolation and Animation", Technical Report DIKU-TR-98/5, Department of Computer Science, University of Copenhagen, Denmark, July, 1998, pp. 93, 7