# Quaternion and Its Application in Rotation Using Sets of Regions

Logah Perumal*

Multimedia University, Faculty of Engineering and Technology,
Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

Received 12 July 2011; received in revised form 01 August 2011; accepted 05 September 2011

## Abstract

This paper is written to aid the readers to understand application of Euler angles and quaternion in representing rotation of a body in 3-dimensional Euclidean space, $\Re^3$. Application of quaternion would later require conversion of the quaternion to Euler angles. This is to enable quaternion to be compatible with other applications which use Euler rotation sequence to represent rotation. Thus, a framework to convert a quaternion, which is produced from a random rotation sequence to Euler angles with any specified rotation sequence is proposed and demonstrated here, to aid practitioners to use quaternion in their applications. This will also enable quaternion to be applied in arbitrary sequence onto applications developed using certain rotation sequence of Euler angles. Finally, a program is developed using Matlab-simulink software to demonstrate application of quaternion in maneuvering orientation of a missile flying in 3D space. Six degree of freedom (6DoF) block, which employs Euler rotation sequence of XYZ, is used to aid users to graphically see the maneuvering of the missile's orientation as it flies in 3-dimensional Euclidean space. Quaternion, which is produced from random rotation sequence keyed in by the user, is converted to Euler angles with rotation sequence XYZ by using the proposed method.

IJETI

**Keywords:** Euler angles, Quaternion, Rotation sequence, Matlab Simulink, orientation, Gimbal, Gimbal lock, Singularity, Sets of region

## 1. Introduction

Euler angles have been used to describe orientation of objects in 2D and 3D space since its formulation by Leonhard Euler. One of the drawbacks of the Euler angles caused limitations in representing orientation of a body. The drawback/disadvantage stated above is known as gimbal lock **[**1**]**. Gimbal lock is a phenomenon in which one of the rotation axes realigns with the other axis and eventually causes loss of one degree of freedom. In 1843, Sir William Rowan Hamilton introduced a new theory to describe orientation of a body in space. It is known as quaternion **[**2**]**, which uses vector of four dimensions and quaternion products to represent orientation. The quaternion has many advantages over Euler angles and one of them is that the gimbal lock phenomenon is avoided.

There have been many attempts to visualize quaternion and several papers/books have been written to describe quaternion in simplest way possible. Such attempts can be seen in **[**3, 4, 5**]**. These papers focus solely on explaining quaternion and its application in rotation. Conversion between Euler angles and quaternion is not presented in these papers. Conversion between Euler angles and quaternion is required in order to enable quaternion to be compatible with other applications which

* Corresponding author. E-mail logah.perumal@mmu.edu.my
Tel.: 06-2523287; Fax: 06-2316552

use Euler angles sequence to represent rotation. This paper is written to aid readers to understand application of Euler angles and quaternion in representing rotation of a body in 3D, with emphasis on new method to convert quaternion to Euler angles. A new method to convert a quaternion produced from arbitrary rotation sequence to Euler angles with any specified sequence is proposed and validated in this paper. Finally, a program is developed using Matlab-Simulink software to demonstrate application of quaternion in maneuvering orientation of a missile flying in 3D space.

This paper is arranged as follows. Sec. 2 is used to describe standards used throughout the paper. Sec. 3 is used to describe gimbals and gimbal lock phenomena using Euler angles. Euler angles and its application in representing rotation of a body is explained in Sec. 4 while Sec. 5 is used to explain quaternion and its application in representing rotation of a body. Sec. 6 is used to describe advantages and application of quaternion. Singularity phenomena using Euler angles and Quaternion is also described in this Section. A new method to convert quaternion produced from arbitrary rotation sequence to Euler angles with XYZ sequence is proposed and validated in Sec. 7. A sample program to demonstrate application of proposed method is presented in Sec. 8. The paper is finally concluded in Sec. 9.

## 2. Coordinate Systems and Standards Used in the Paper

Three coordinate systems are described in this paper. Those are:

(1) The inertial/earth coordinate system (XYZ), which is fixed in the Euclidean space.

(2) The body coordinate system (xyz), which is attached and moves together with the moving body in 3-dimensional Euclidean space.

(3) The mixed coordinate system (xyZ) which is combination of inertial/earth coordinate system and body coordinate system.

Orientation of a moving body in 3-dimensional Euclidean space can be described by:

(1) Using three angles measured from the inertial/earth coordinate system, referred to as Cartesian angles (θ, ϕ and ψ).

(2) Using three angles measured from mixed axis of rotation system (combination of inertial/earth coordinate system and body coordinate system), which is known as Euler angles (α, β and γ).

Fig. 1 shows the coordinate systems and angle conventions used in the paper:

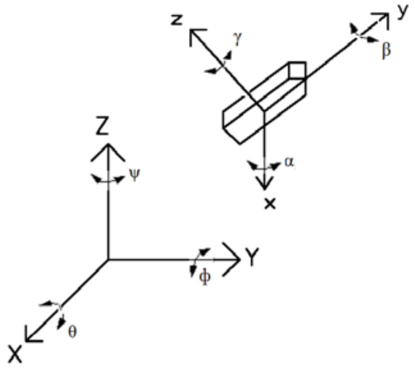

Fig. 1 Coordinate systems and angle conventions

Copyright © TAETI

## 3. Gimbals, Gimbal Lock and Euler Angles

A gimbal is a ring that is free to rotate about an axis. Combination of 3 gimbals as a set, with each gimbal free to rotate about one of the three axes (X, Y and Z) would then enable representation of rotation in 3-dimensional Euclidean space, $\Re^3$. The gimbals are mounted together in such a way that the innermost gimbal is free to rotate about its axis without affecting the intermediate and outermost gimbals. The intermediate gimbal is free to rotate about its axis, but it will cause the innermost gimbal to follow the rotation of intermediate gimbal and the outermost gimbal will remain unaffected. Rotation of the outermost gimbal about its axis of rotation will cause both intermediate and innermost gimbals to follow the rotation of the outermost gimbal. The gimbals are allowed to rotate about their respective axes only. Gimbals are used in many applications such as in gyroscopes, in inertial measurement unit and etc. Gimbals in inertial measurement unit are used to detect orientation of a body such as aircrafts and spacecrafts by using three Euler angles. Sensors installed onto the gimbals read out three Euler angles which describe orientation of a body in 3-dimensional space (each angle indicates rotation about one of the three axes). The Euler angles are expressed in terms of mixed axis of rotation system (combination of inertial coordinate system and body coordinate system), since the innermost gimbal rotates around an axis fixed in the moving body, the intermediate gimbal rotates around line of node which follows the outermost gimbal and the outermost gimbal rotates around an axis fixed in the inertial coordinate system [6].

The nature of gimbals' rotation as described above leads to an undesired phenomenon, known as gimbal lock. Gimbal lock is a phenomenon in which one of the rotation axes realigns with the other axis and eventually causes loss of one degree of freedom. For example, if the intermediate gimbal is rotated such a way that the innermost gimbal follows the rotation until it aligns with the outermost gimbal (outermost gimbal is unaffected by the rotation of the intermediate gimbal), then any following rotations about the outermost gimbal would be equivalent to rotation about the innermost gimbal. Thus, there is a loss of one degree of freedom here. Very well-known gimbal lock incident occurred in Apollo 11 moon mission when computer error occurred and assumed the spacecraft to be in gimbal lock orientation [1]. Gimbal lock can be avoided by using fourth gimbal, but it will add complexity to the system. Another solution would be to reset the device by moving the other gimbals to a random orientation when it is near to gimbal lock orientation. New and more reliable methods have been investigated such as strap down system (which employs quaternion) and application of fluid bearings or flotation chamber **[7]**. These methods eliminate the need for gimbals entirely.

Some coordinate systems in mathematics employ the analogy of real gimbals and thus the orientation is given in terms of Euler angles. Gimbal lock phenomenon cannot be avoided whenever orientation of a body is represented using Euler angles (except for rotation sequences with repeated axis of rotation). Gimbal lock in mathematics is referred to as singularities. Thus, from this point onwards, gimbal lock would be referred as singularities. Singularities in mathematics are avoided by changing the representations (by switching between two different sets of Euler angles) whenever an object nears a singularity, and it is known as dual Euler method [8, 9]. Another simpler way to avoid singularities would be by using Quaternion.

## 4. Euler Angles: Coordinate Rotation, 3D Rotation, Rotation Sequence and Normalized DCM

As described in Sec. 3, there are three components of Euler angles which can be defined as:

$$\begin{aligned}
&\mathbf{e} = [\alpha, \beta, \gamma]^T \\
&\textit{where} \\
&\mathbf{e} = \textit{Euler angle vector} \\
&\alpha = \textit{rotation angle about } x-\textit{axis} \\
&\beta = \textit{rotation angle about } y-\textit{axis} \\
&\gamma = \textit{rotation angle about } z-\textit{axis}
\end{aligned} \tag{1}$$

Rotation about a single axis is known as coordinate rotation. Consider $\mathbf{t} \in \Re^3$, a vector in inertial coordinate system. The vector **t**, when rotated by an angle about a single axis would produce new vector $\mathbf{t'} \in \Re^3$ which can be described in body coordinate system through the relation:

$$\begin{aligned}
&\mathbf{t'} = R_m(\theta)\mathbf{t} \\
&\textit{where} \\
&\mathbf{t'} = \textit{rotated vector described in body coordinate system} \\
&R_m(\theta) = \textit{Coordinate rotation matrix} \\
&\mathbf{t} = \textit{vector in inertial coordinate system} \\
&m = \textit{axis of rotation } x, y, z \textit{ or rotation sequence} \\
&\theta = \begin{cases} \alpha \textit{ if } m = x \\ \beta \textit{ if } m = y \\ \gamma \textit{ if } m = z \end{cases}
\end{aligned} \tag{2}$$

Coordinate rotation matrix for each axis is given by:

$$R_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{3}$$

$$R_Y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{4}$$

$$R_Z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The rotation process can be reversed and the vector **t** can be represented by the relation:

$$\mathbf{t} = R_m^T(\theta)\mathbf{t'} \tag{6}$$

Leonhard Euler showed that any rotation can be accomplished by a sequence of three rotations about the coordinate axes [10]. This is most practically done by practitioners, by combining rotations. Since there are three axes, thus there are total of 27 possible rotation sequences. Only 12 of them are used since consecutive rotations around the same axis would reduce the degree of freedom to one or two degree of freedom (rotation sequences with consecutive repeating axes are: XXX, YYY, ZZZ, XXY, XXZ, YYX, YYZ, ZZX, ZZY, XYY, YXX, ZXX, ZYY, XZZ and YZZ). The 12 rotation sequences with three degree of freedom are: XYZ, XZY, YXZ, YZX, ZXY, ZYX, XYX, ZYZ, ZXZ, YXY, XZX, and YZY.

IJETI

Rotation matrix for combined rotations, which is also known as direction cosine matrix (DCM), can be obtained by multiplying the rotation matrix for respective axes according to the sequence as shown below:

$$
\begin{aligned}
&R_{abc} = R_a(\theta)R_b(\theta)R_c(\theta)\\
&where\\
&R_{abc} = Rotation\ matrix\\
&abc = rotation\ sequence\\
&a,b,c = \begin{cases} X \\ Y \\ Z \end{cases}\\
&\theta = \begin{cases} \alpha\ if\ \ rotation\ axis = x \\ \beta\ if\ \ rotation\ axis = y \\ \gamma\ if\ \ rotation\ axis = z \end{cases}
\end{aligned}
\tag{7}
$$

For the rotation sequence of XYZ:

$$
\begin{aligned}
&R_{XYZ} = R_X(\alpha)R_Y(\beta)R_Z(\gamma)\\
&R_{XYZ} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}\\
&R_{XYZ} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}
\end{aligned}
\tag{8}
$$

Now, applying the **Eq. 2**:

$$
\begin{aligned}
&\mathbf{t}' = R_{abc}\mathbf{t}\\
&\mathbf{t}' = R_{XYZ}\mathbf{t}\\
&\mathbf{t}' = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}\mathbf{t}
\end{aligned}
\tag{9}
$$

IJETI

Each rotation sequence yields different results and some applications might use different rotation sequence than the other. For example, rotation sequence ZXZ is commonly used in the field of engineering, robotics and in the study of gyroscopic motion while rotation sequence of XYZ is commonly used in the field of aerospace engineering and computer graphics **[8]**. Rounding errors accumulate when several rotations are accomplished using DCM, due to the trigonometric functions in DCM and causes the matrix to become non-orthogonal. The rounding error can be reduced by using normalized DCM:

$$
\begin{aligned}
&R_{abc}^{norm} = R_{abc}(R_{abc}^{T}R_{abc})^{-1/2}\\
&where\\
&R_{abc}^{norm} = Normalized\ \ DCM\ \ with\ rotation\ sequence\ abc\\
&a,b,c = \begin{cases} X \\ Y \\ Z \end{cases}
\end{aligned}
\tag{10}
$$

For the rotation sequence of XYZ:

Copyright © TAETI

$$R_{abc}^{norm} = R_{XYZ}^{norm} = R_{XYZ}(R_{XYZ}^{T} R_{XYZ})^{-1/2} \tag{11}$$

## 5. Quaternion: Coordinate Rotation, 3D Rotation, Rotation Sequence and Normalized DCM

Gimbal lock can be avoided by adding a forth gimbal (as mentioned earlier in Sec. 3). Similarly in mathematics, singularity can be avoided by using quaternion, which consists of four elements. Quaternion rotates a body in inertial coordinate system instead of mixed axis of rotation system as Euler angles. Quaternion consists of a scalar part and three imaginary parts, which can be represented as:

$$\begin{aligned}
&Q = [q_0, \mathbf{q}]^T \\
&where \\
&Q = quaternion\ vector\ with\ \ four\ \ elements \\
&q_0 = scalar\ part \\
&\mathbf{q} = vector\ part \\
&\quad = [q_1 i, q_2 j, q_3 k] \\
&i, j, k = three\ imaginary\ axes\ in\ \Re^3 \\
&i = rotation\ about\ X-axis \\
&j = rotation\ about\ Y-axis \\
&k = rotation\ about\ Z-axis
\end{aligned} \tag{12}$$

A quaternion with zero scalar part is known as pure quaternion. A pure quaternion is given by:

$$Q = [0, \mathbf{q}]^T \tag{13}$$

A quaternion with only a scalar part is called a real quaternion. A real quaternion is given by:

$$Q = [q_0, 0]^T \tag{14}$$

Norm of a quaternion is given as:

$$|Q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{15}$$

A quaternion with norm 1 is known as unit quaternion. Quaternion can be written in terms of angle and axis of rotation as:

$$\begin{aligned}
&Q = [q_0, \mathbf{q}]^T \\
&q_0 = \cos\left(\frac{\theta}{2}\right) \\
&\mathbf{q} = \sin\left(\frac{\theta}{2}\right)\mathbf{n} \\
&where \\
&\mathbf{n} = \begin{cases} i \\ j \\ k \end{cases} \\
&\theta = \begin{cases} \alpha\ if\ \ \mathbf{n} = i \\ \beta\ if\ \ \mathbf{n} = j \\ \gamma\ if\ \ \mathbf{n} = k \end{cases}
\end{aligned} \tag{16}$$

Thus, quaternion can also be written generally as:

IJETI

Copyright © TAETI

$$Q=\cos\left(\frac{\theta}{2}\right)+\sin\left(\frac{\alpha}{2}\right)i+\sin\left(\frac{\beta}{2}\right)j+\sin\left(\frac{\gamma}{2}\right)k \tag{17}$$

Conjugate of a quaternion is:

$$Q^{*}=\cos\left(\frac{\theta}{2}\right)-\sin\left(\frac{\alpha}{2}\right)i-\sin\left(\frac{\beta}{2}\right)j-\sin\left(\frac{\gamma}{2}\right)k \tag{18}$$

Consider coordinate rotation of $\mathbf{t}\in\Re^{3}$, a vector in inertial coordinate system. The vector **t**, when rotated by an angle about a single axis would produce a new vector $\mathbf{t}'\in\Re^{3}$ which can be described in inertial coordinate system through the relation:

$$\begin{aligned}&\mathbf{t}'=Q_n\times\mathbf{t}\times Q_n^{*}\\&where\\&\mathbf{t}'=Rotated\ vector\ described\ in\ inertial\ coordinate\ system\\&\mathbf{t}=Vector\ in\ inertial\ coordinate\ system\\&Q=Quaternion\ vector\ with\ 4\ elements\\&Q^{*}=Conjugate\ of\ Q\\&n=Rotation\ Axis\ or\ Rotation\ sequence\end{aligned} \tag{19}$$

Quaternion for rotation about individual axis is given by:

$$Q_i=Q_X=\cos\left(\frac{\alpha}{2}\right)+\sin\left(\frac{\alpha}{2}\right)i+\sin\left(\frac{0}{2}\right)j+\sin\left(\frac{0}{2}\right)k=\cos\left(\frac{\alpha}{2}\right)+\sin\left(\frac{\alpha}{2}\right)i \tag{20}$$

$$Q_j=Q_Y=\cos\left(\frac{\beta}{2}\right)+\sin\left(\frac{0}{2}\right)i+\sin\left(\frac{\beta}{2}\right)j+\sin\left(\frac{0}{2}\right)k=\cos\left(\frac{\beta}{2}\right)+\sin\left(\frac{\beta}{2}\right)j \tag{21}$$

$$Q_k=Q_Z=\cos\left(\frac{\gamma}{2}\right)+\sin\left(\frac{0}{2}\right)i+\sin\left(\frac{0}{2}\right)j+\sin\left(\frac{\gamma}{2}\right)k=\cos\left(\frac{\gamma}{2}\right)+\sin\left(\frac{\gamma}{2}\right)k \tag{22}$$

Multiplication in quaternion is carried out using Hamilton’s Rules, discovered by Sir William Rowan Hamilton (1805-1865) [2];

$$\begin{aligned}&i^2=j^2=k^2=ijk=-1\\&ij=k\\&jk=i\\&ki=j\\&ji=-k\\&kj=-i\\&ik=-j\end{aligned} \tag{23}$$

Conjugate of a quaternion is used in Eq. 19 in order to complete a rotation and to obtain a pure quaternion [11]. The Eq. 19 can be rewritten in general form as:

$$\begin{aligned}
&\mathbf{t}' = R^Q \mathbf{t} \\
&where \\
&\mathbf{t}' = Rotated\ vector\ described\ in\ inertial\ coordinate\ system \\
&\mathbf{t} = Vector\ in\ inertial\ coordinate\ system \\
&R^Q = Quaternion\ rotation\ matrix
\end{aligned} \tag{24}$$

The general quaternion rotation matrix is given by:

$$R^Q = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2q_0q_1 + 2q_2q_3 \\ -2q_0q_2 + 2q_1q_3 & 2q_0q_1 + 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{25}$$

Successive rotations can be combined using similar method presented in Sec. 4. The 12 rotation sequences with three degree of freedom are: XYZ, XZY, YXZ, YZX, ZXY, ZYX, XYX, ZYZ, ZXZ, YXY, XZX, and YZY. Quaternion for a combined rotation can be obtained by multiplying the quaternion representations for the respective axes according to the sequence.

$$\begin{aligned}
&Q_{abc} = Q_a \times Q_b \times Q_c \\
&where \\
&abc = rotation\ sequence \\
&a,b,c = \begin{cases} X \\ Y \\ Z \end{cases}
\end{aligned} \tag{26}$$

For example, quaternion for rotation sequence of XYZ is given by:

$$\begin{aligned}
&Q_{XYZ} = Q_X \times Q_Y \times Q_Z \\
&Q_{XYZ} = q_0^{XYZ} + q_1^{XYZ} i + q_2^{XYZ} j + q_3^{XYZ} k \\
&where \\
&q_0^{XYZ} = \cos\left(\gamma/2\right)\cos\left(\beta/2\right)\cos\left(\alpha/2\right) + \sin\left(\gamma/2\right)\sin\left(\beta/2\right)\sin\left(\alpha/2\right) \\
&q_1^{XYZ} i = \cos\left(\gamma/2\right)\cos\left(\beta/2\right)\sin\left(\alpha/2\right) - \sin\left(\gamma/2\right)\sin\left(\beta/2\right)\cos\left(\alpha/2\right) \\
&q_2^{XYZ} j = \cos\left(\gamma/2\right)\sin\left(\beta/2\right)\cos\left(\alpha/2\right) + \cos\left(\beta/2\right)\sin\left(\gamma/2\right)\sin\left(\alpha/2\right) \\
&q_3^{XYZ} k = \cos\left(\beta/2\right)\sin\left(\gamma/2\right)\cos\left(\alpha/2\right) - \cos\left(\gamma/2\right)\sin\left(\beta/2\right)\sin\left(\alpha/2\right)
\end{aligned} \tag{27}$$

Now applying Eq. 24;

$$\begin{aligned}
&\mathbf{t}' = Q_{XYZ} \times \mathbf{t} \times Q_{XYZ}^* \\
&\mathbf{t}' = R^Q \mathbf{t} \\
&\mathbf{t}' = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2q_0q_1 + 2q_2q_3 \\ -2q_0q_2 + 2q_1q_3 & 2q_0q_1 + 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \mathbf{t} \\
&where \\
&q_0 = q_0^{XYZ} \\
&q_1 = q_1^{XYZ} \\
&q_2 = q_2^{XYZ} \\
&q_3 = q_3^{XYZ}
\end{aligned} \tag{28}$$

IJETI

Rounding errors accumulate when several rotations are accomplished, due to the trigonometric functions. The rounding error can be reduced by using normalized quaternion. A quaternion can be normalized by using the relation:

$$Q^{norm} = \frac{Q}{|Q|} = \frac{q_0 + q_1 i + q_2 j + q_3 k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}}$$
$$where$$
$$Q^{norm} = Normalized\ \ quaternion \tag{29}$$

The normalized general quaternion rotation matrix (using normalized quaternion) is then given by:

$$R^{Q,norm} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_0q_3+2q_1q_2 & 1-2q_1^2-2q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2 \end{bmatrix} \tag{30}$$

## 6. Advantages of Quaternion

Representation of rotation using quaternion has advantages compared to DCM [6, 9, 12, 13]**;**

(1) Quaternion is more compact compared to DCM (DCM; 9 elements, quaternion; 4 elements). This can be seen by comparing Eq. 17 with Eq. 3-5.

(2) DCM involves more number of arithmetic operations and thus requires longer computation time compared to quaternion.

(3) Extracting the angle and axis of rotation is simpler using quaternion.

(4) The normalization of a rotation matrix is computationally more expensive than normalization of a quaternion rotation matrix. This can be seen by comparing between Eq. 10 with Eq. 29.

(5) Singularity is avoided entirely, as described below.

Singularity occurs when a body is rotated about mixed coordinate system (using Euler angles), because two of the axes (innermost and intermediate gimbals) move in accordance to the rotation. This provides possibility for the axes to align with each other and causes loss of one degree of freedom, as discussed in Sec. 3. Singularity is avoided entirely when quaternion is used, since a body is rotated about inertial coordinate system in which the axes are fixed, and there is no possibility for the axes to align with each other. In [14], it is wrongly mentioned that singularity could occur with quaternion, if not used properly for sequence of XYZ. Proof of avoidance of singularity when quaternion is used is presented below.

First, occurrence of singularity using Euler angle is demonstrated using XYZ rotation sequence. Later, quaternion is applied in similar sequence, to show that singularity can be avoided entirely, when quaternion is used. Consider rotation about X-axis by 0 degree, followed by rotation about Y-axis by 90 degrees and finally rotation about Z-axis by an angle of γ. Resulting DCM for this case can be obtained by substituting $\alpha = 0, \beta = 90$ degrees and γ = γ degrees into Eq. 8;

$$R_{XYZ} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}$$
$$R_{XYZ} = \begin{bmatrix} 0 & 0 & -1 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ \cos(\gamma) & \sin(\gamma) & 0 \end{bmatrix} \tag{31}$$

Copyright © TAETI

Now, consider rotation about X-axis by an angle of $\alpha$ , followed by rotation about Y-axis by 90 degrees and finally rotation about Z-axis by an angle of 0 degree. Resulting DCM for this case can be obtained by substituting $\alpha = -\alpha$ degrees, $\beta = 90$ degrees and γ = 0 degree into Eq. 8 again;

$$R_{XYZ} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}$$
$$R_{XYZ} = \begin{bmatrix} 0 & 0 & -1 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ \cos(\alpha) & \sin(\alpha) & 0 \end{bmatrix} \tag{32}$$

It can be seen that both cases yield similar DCM. This is because rotation about Y-axis by 90 degrees causes the Z-axis and X-axis to align with each other and any consecutive rotation about the Z-axis and X-axis yield same results, thus singularity occurs. This can be avoided by using quaternion.

Recreating the first case by substituting $\alpha = 0, \beta = 90$ degrees and γ = γ degree into Eq. 27 gives:

$$\begin{aligned} q_0^{XYZ} &= \frac{1}{\sqrt{2}}\cos\left(\gamma/2\right) \\ q_1^{XYZ} i &= -\frac{1}{\sqrt{2}}\sin\left(\gamma/2\right) \\ q_2^{XYZ} j &= \frac{1}{\sqrt{2}}\cos\left(\gamma/2\right) \\ q_3^{XYZ} k &= \frac{1}{\sqrt{2}}\sin\left(\gamma/2\right) \end{aligned} \tag{33}$$

IJETI

Recreating the second case by substituting $\alpha = -\alpha \ \deg rees, \beta = 90\, d\ egrees$ and $\gamma = 0\ \ d\ egree$ into Eq. 27 gives;

$$\begin{aligned} q_0^{XYZ} &= \frac{1}{\sqrt{2}}\cos\left(\alpha/2\right) \\ q_1^{XYZ} i &= \frac{1}{\sqrt{2}}\sin\left(\alpha/2\right) \\ q_2^{XYZ} j &= \frac{1}{\sqrt{2}}\cos\left(\alpha/2\right) \\ q_3^{XYZ} k &= -\frac{1}{\sqrt{2}}\sin\left(\alpha/2\right) \end{aligned} \tag{34}$$

Now, it can be seen that both cases using quaternion yield different answers, thus avoid singularity.

Quaternion can be used to represent rotation, reflection and scaling. Quaternion has been used in many areas such as in molecular modeling [15], detection of trajectories of particles in fluid flows [16], gyroscopic motion, navigation and guidance [17], control theory [18], orbital mechanics of satellites [19], electrical technology [20], robotics [21], signal processing, physics and flight simulators [22]. Another application which has greatly benefitted from quaternion is in the development of computer games, graphics and 3D virtual worlds [23]. Quaternion is used in these applications to obtain smooth rotation by interpolating between orientations, known as spherical linear interpolation (slerp) [12, 24]. This paper focuses on application of quaternion in representing rotation.

Copyright © TAETI

## 7. Conversion Between Euler Angles and Quaternion

Conversion from Euler angles to quaternion is straight forward as described in Sec. 5 and Eq. 17. However, conversion of quaternion to Euler angles can be done by several ways. In most cases, conversion from quaternion to Euler angles is done by first generating the rotation matrix (DCM) both from quaternion and Euler angles for a given rotation sequence and then the similar matrix elements are compared and solved for the Euler angles [8, 25, 11]. In [6, 26], the authors proposed geometric methods to convert given quaternion to Euler angles, which reduces the computational time and complexity since it does not involve any matrix generation. In [6], the author proposed a geometric method to convert quaternion to equivalent Euler angles sequence. Similar method is described in [26]. Advantage of the geometric method is that conversion is done without involving any matrices, which causes reduction in computational time and complexity. In [27], the author proposed a method to convert a given quaternion to any Euler angles sequence by first deriving two different expressions involving elements of the quaternion. Both expressions are mathematically simplified and rewritten individually and later compared for similar term and solved for the angles. The drawbacks of this method are that the method is not certain to work for rotations taking place around the same axis twice and special steps need to be taken to deal with singularities.

All the methods proposed in the referred papers above are restricted to conversion (quaternion to Euler angles) within similar rotation sequence, except for method proposed in [27]. This imposes restrictions onto rotation sequence for quaternion applicable onto an application developed based on particular Euler angles sequence. A new method is proposed in this paper to convert a quaternion produced from arbitrary rotation sequence to Euler angles with specific rotation sequence XYZ (but not limited to this sequence), without involving any matrices. The proposed method is also proved to be valid for all rotation sequences, including rotations that are taking place around the same axis twice.

### *7.1. Conversion of Quaternion with Arbitrary Rotation Sequence to Euler Angles with XYZ Sequence*

A frame is defined as a coordinate system in $\Re^3$ and it is represented by a vector of 3 dimensions (coordinates) [28]. Three frames are defined in this paper; Body frame (body coordinate system; xyz), Cartesian frame (inertial coordinate system; XYZ), and mixed frame as described in Sec. 2. Since quaternion rotates a body within inertial coordinate system as shown in Sec. 5, thus Cartesian frame can be used to represent attitude of a rotated body, through use of Cartesian coordinates.

A set of regions can be defined by imposing restrictions onto the Cartesian coordinates of $\mathbf{t}'$ in the Cartesian frame. There are three types of regions; octants, quadrants and axes. Total of 26 regions (which consist of 8 octants, 12 quadrants and 6 axes) can be defined in the Cartesian frame using relation **[**29, 30**]**:

$$\begin{aligned}
&x(n) \quad for \quad X=\begin{cases}<0\\>0\\=0\end{cases};\ Y=\begin{cases}<0\\>0\\=0\end{cases};\ Z=\begin{cases}<0\\>0\\=0\end{cases}\\
&where\\
&x = Region\ type\\
&\quad=\begin{cases}Oct\ \ for\ Octant\\Quad\ \ for\ Quadrant\\Axe\ \ for\ Axis\end{cases}\\
&n = Region\ numbering\\
&X,Y,Z = Cartesian\ coordinates\ of\ \mathbf{t}'
\end{aligned} \tag{35}$$

The 8 Octants are defined by:

$$\begin{array}{|lll|}\hline Oct(1) & for & X>0 \\ & & Y<0 \\ & & Z<0 \\ Oct(2) & for & X>0 \\ & & Y>0 \\ & & Z<0 \\ Oct(3) & for & X<0 \\ & & Y>0 \\ & & Z<0 \\ Oct(4) & for & X<0 \\ & & Y<0 \\ & & Z<0 \\ \hline\end{array} \qquad \begin{array}{|lll|}\hline Oct(5) & for & X<0 \\ & & Y<0 \\ & & Z>0 \\ Oct(6) & for & X>0 \\ & & Y<0 \\ & & Z>0 \\ Oct(7) & for & X>0 \\ & & Y>0 \\ & & Z>0 \\ Oct(8) & for & X<0 \\ & & Y>0 \\ & & Z>0 \\ \hline\end{array}$$

The 12 Quadrants are defined by:

$$\begin{array}{|lll|}\hline Quad(9) & for & X>0 \\ & & Y=0 \\ & & Z<0 \\ Quad(10) & for & X<0 \\ & & Y=0 \\ & & Z<0 \\ Quad(11) & for & X<0 \\ & & Y=0 \\ & & Z>0 \\ Quad(12) & for & X>0 \\ & & Y=0 \\ & & Z>0 \\ \hline\end{array} \qquad \begin{array}{|lll|}\hline Quad(9) & for & X>0 \\ & & Y=0 \\ & & Z<0 \\ Quad(10) & for & X<0 \\ & & Y=0 \\ & & Z<0 \\ Quad(11) & for & X<0 \\ & & Y=0 \\ & & Z>0 \\ Quad(12) & for & X>0 \\ & & Y=0 \\ & & Z>0 \\ \hline\end{array} \qquad \begin{array}{|lll|}\hline Quad(9) & for & X>0 \\ & & Y=0 \\ & & Z<0 \\ Quad(10) & for & X<0 \\ & & Y=0 \\ & & Z<0 \\ Quad(11) & for & X<0 \\ & & Y=0 \\ & & Z>0 \\ Quad(12) & for & X>0 \\ & & Y=0 \\ & & Z>0 \\ \hline\end{array}$$

The 6 Axes are defined by:

$$\begin{array}{|lll|}\hline Axe(21) & for & X=0 \\ & & Y=0 \\ & & Z>0 \\ Axe(22) & for & X=0 \\ & & Y>0 \\ & & Z=0 \\ Axe(23) & for & X>0 \\ & & Y=0 \\ & & Z=0 \\ \hline\end{array} \qquad \begin{array}{|lll|}\hline Axe(24) & for & X<0 \\ & & Y=0 \\ & & Z=0 \\ Axe(25) & for & X=0 \\ & & Y<0 \\ & & Z=0 \\ Axe(26) & for & X=0 \\ & & Y=0 \\ & & Z<0 \\ \hline\end{array}$$

Cartesian angles and Euler angles can then be calculated based on the region in which the rotated vector $\mathbf{t'}$ lies. The vector $\mathbf{t'}$ is also known as visualizing quaternion, since it is used as a visualizing tool to specify the regions [29, 30]. Euler angles with XYZ sequence can then be computed using corresponding formulas for the respective region by using the following arrangements (Arrangement 1):

$$\begin{array}{|l|}\hline Oct(1),\ Oct(2) \\ \alpha=\alpha \\ \beta=\left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma=\tan^{-1}\left(Y/X\right) \\ if\ \ \gamma\geq 40\ \deg rees,\ then\ \beta_{new}=\left|\tan^{-1}\left(Z/Y\right)\right| \\ \hline\end{array} \qquad \begin{array}{|l|}\hline Oct(3) \\ \alpha=\alpha \\ \beta=\left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma=\left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right| \\ if\ \ \gamma\geq 130\ \deg rees,\ then\ \beta_{new}=\left|\tan^{-1}\left(Z/X\right)\right| \\ \hline\end{array}$$

$$\begin{array}{|l|}\hline Oct(4) \\ \alpha=\alpha \\ \beta=\left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma=-\left[\left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right|\right] \\ if\ \ \gamma\geq 130\ \deg rees,\ then\ \beta_{new}=\left|\tan^{-1}\left(Z/X\right)\right| \\ \hline\end{array} \qquad \begin{array}{|l|}\hline Oct(5) \\ \alpha=\alpha \\ \beta=-\left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma=3.142+\left|\tan^{-1}\left(Y/X\right)\right| \\ if\ \ \gamma\geq 225\ \deg rees,\ then\ \beta_{new}=-\left|\tan^{-1}\left(Z/Y\right)\right| \\ \hline\end{array}$$

$$\boxed{\begin{array}{l} Oct(6),\ Oct(7) \\ \alpha = \alpha \\ \beta = -\left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = \tan^{-1}\left(Y/X\right) \\ if\ \ \gamma \geq 40\ \deg rees,\ then\ \beta_{new} = -\left|\tan^{-1}\left(Z/Y\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Oct(8) \\ \alpha = \alpha \\ \beta = -\left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma = 3.142 - \left|\tan^{-1}\left(Y/X\right)\right| \\ if\ \ \gamma \geq 130\ \deg rees,\ then\ \beta_{new} = -\left|\tan^{-1}\left(Z/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Quad(9) \\ \alpha = \alpha \\ \beta = \left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Quad(10) \\ \alpha = \alpha \\ \beta = \left(3.142/2\right) + \left|\tan^{-1}\left(X/Z\right)\right| \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Quad(11) \\ \alpha = \alpha \\ \beta = 3.142 + \left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Quad(12) \\ \alpha = \alpha \\ \beta = -\left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Quad(13) \\ \alpha = \alpha \\ \beta = -\left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma = -\left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Quad(14) \\ \alpha = \alpha \\ \beta = \left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma = -\left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Quad(15) \\ \alpha = \alpha \\ \beta = 90^0 + \left|\tan^{-1}\left(Y/Z\right)\right| \\ \gamma = -\left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Quad(16) \\ \alpha = \alpha \\ \beta = 180^0 + \left|\tan^{-1}\left(Z/Y\right)\right| \\ \gamma = -\left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Quad(17) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = 3.142 - \left|\tan^{-1}\left(Y/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Quad(18) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = \left|\tan^{-1}\left(Y/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Quad(19) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = -\left|\tan^{-1}\left(Y/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Quad(20) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = -\left[\left(3.142/2\right) + \left|\tan^{-1}\left(X/Y\right)\right|\right] \end{array}}$$

$$\boxed{\begin{array}{l} Axe(21) \\ \alpha = \alpha \\ \beta = -\left(3.142/2\right) \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Axe(22) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = \left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Axe(23) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = 0^0 \end{array}}$$

$$\boxed{\begin{array}{l} Axe(24) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = 3.142 \end{array}}$$

$$\boxed{\begin{array}{l} Axe(25) \\ \alpha = \alpha \\ \beta = 0^0 \\ \gamma = -\left(3.142/2\right) \end{array}}$$

$$\boxed{\begin{array}{l} Axe(26) \\ \alpha = \alpha \\ \beta = \left(3.142/2\right) \\ \gamma = 0^0 \end{array}}$$

Based on the Arrangement 1, it can be seen that the angle $\alpha$ remains fixed for the entire region. This is due to the fact that in XYZ Euler angles sequence, the innermost gimbal rotates about x-axis in Body frame. Thus, total of four angles can be extracted when quaternion is converted to Euler angles using the method proposed. Those angles are; rotation angle about X-axis in Cartesian frame, rotation angle about Y-axis in Cartesian frame, rotation angle about Z-axis in Cartesian frame, and rotation angle about x-axis in Body frame. These four angles can be used to represent rotation of a body in Euclidean space $\Re^3$, as shown in upcoming Sec. 8**.** It can also be seen that $\beta$ is dependent on $\gamma$ in regions 1 to 8. This is because the intermediate gimbal rotates around line of node, which follows the outermost gimbal, as explained in Sec. 3. Thus, the rotation angle of the intermediate gimbal is dependent on rotation of the outermost gimbal.

Euler angles with different rotation sequence other than XYZ can be extracted by modifying the formulas for each region accordingly. Another way to obtain Euler angles with different sequence other than XYZ would be by using intra-Euler angle conversion provided in [8]. The intra-Euler angle conversion can be applied onto Euler angles with sequence of XYZ (which is extracted using the method proposed) to obtain Euler angles with desired rotation sequence. Similarly, Cartesian angles can be computed using the following arrangements (Arrangement 2):

$$\boxed{\begin{array}{l} Oct(1) \\ \alpha = -\left|\tan^{-1}\left(Y/Z\right)\right| \\ \beta = \left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = -\left|\tan^{-1}\left(Y/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Oct(2) \\ \alpha = \left|\tan^{-1}\left(Y/Z\right)\right| \\ \beta = \left|\tan^{-1}\left(Z/X\right)\right| \\ \gamma = \left|\tan^{-1}\left(Y/X\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Oct(3) \\ \alpha = \left|\tan^{-1}\left(Y/Z\right)\right| \\ \beta = \left(3.142/2\right) + \left|\tan^{-1}\left(X/Z\right)\right| \\ \gamma = \left(3.142/2\right) + \left|\tan^{-1}\left(X/Y\right)\right| \end{array}}$$

$$\boxed{\begin{array}{l} Oct(4) \\ \alpha = -\left|\tan^{-1}\left(Y/Z\right)\right| \\ \beta = \left(3.142/2\right) + \left|\tan^{-1}\left(X/Z\right)\right| \\ \gamma = -\left[\left(3.142/2\right) + \left|\tan^{-1}\left(X/Y\right)\right|\right] \end{array}}$$

$$\begin{aligned}&Oct(5)\\ &\alpha = -\left[\left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\right]\\ &\beta = (3.142)+\left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = -\left[\left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right|\right]\end{aligned}$$

$$\begin{aligned}&Oct(6)\\ &\alpha = -\left[\left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\right]\\ &\beta = -\left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = -\left|\tan^{-1}\left(Y/X\right)\right|\end{aligned}$$

$$\begin{aligned}&Oct(7)\\ &\alpha = \left[\left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\right]\\ &\beta = -\left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = \left|\tan^{-1}\left(Y/X\right)\right|\end{aligned}$$

$$\begin{aligned}&Oct(8)\\ &\alpha = \left[\left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\right]\\ &\beta = 3.142+\left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = \left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right|\end{aligned}$$

$$\begin{aligned}&Quad(9)\\ &\alpha = 0\\ &\beta = \left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = 0\end{aligned}$$

$$\begin{aligned}&Quad(10)\\ &\alpha = 0\\ &\beta = \left(3.142/2\right)+\left|\tan^{-1}\left(X/Z\right)\right|\\ &\gamma = 3.142\end{aligned}$$

$$\begin{aligned}&Quad(11)\\ &\alpha = 0\\ &\beta = 3.142+\left|\tan^{-1}\left(X/Z\right)\right|\\ &\gamma = 3.142\end{aligned}$$

$$\begin{aligned}&Quad(12)\\ &\alpha = 3.142\\ &\beta = -\left|\tan^{-1}\left(Z/X\right)\right|\\ &\gamma = 0\end{aligned}$$

$$\begin{aligned}&Quad(13)\\ &\alpha = \left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\\ &\beta = 0\\ &\gamma = -\left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Quad(14)\\ &\alpha = -\left|\tan^{-1}\left(Y/Z\right)\right|\\ &\beta = 0\\ &\gamma = -\left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Quad(15)\\ &\alpha = \left|\tan^{-1}\left(Y/Z\right)\right|\\ &\beta = 0\\ &\gamma = \left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Quad(16)\\ &\alpha = \left(3.142/2\right)+\left|\tan^{-1}\left(Z/Y\right)\right|\\ &\beta = 0\\ &\gamma = \left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Quad(17)\\ &\alpha = \left(3.142/2\right)\\ &\beta = 3.142\\ &\gamma = \left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right|\end{aligned}$$

$$\begin{aligned}&Quad(18)\\ &\alpha = \left(3.142/2\right)\\ &\beta = 0\\ &\gamma = \left|\tan^{-1}\left(Y/X\right)\right|\end{aligned}$$

$$\begin{aligned}&Quad(19)\\ &\alpha = -\left(3.142/2\right)\\ &\beta = 0\\ &\gamma = -\left|\tan^{-1}\left(Y/X\right)\right|\end{aligned}$$

$$\begin{aligned}&Quad(20)\\ &\alpha = -\left(3.142/2\right)\\ &\beta = 3.142\\ &\gamma = -\left[\left(3.142/2\right)+\left|\tan^{-1}\left(X/Y\right)\right|\right]\end{aligned}$$

$$\begin{aligned}&Axe(21)\\ &\alpha = 3.142\\ &\beta = -\left(3.142/2\right)\\ &\gamma = 0\end{aligned}$$

$$\begin{aligned}&Axe(22)\\ &\alpha = \left(3.142/2\right)\\ &\beta = 0\\ &\gamma = \left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Axe(23)\\ &\alpha = 0\\ &\beta = 0\\ &\gamma = 0\end{aligned}$$

$$\begin{aligned}&Axe(24)\\ &\alpha = 0\\ &\beta = 3.142\\ &\gamma = 3.142\end{aligned}$$

$$\begin{aligned}&Axe(25)\\ &\alpha = -\left(3.142/2\right)\\ &\beta = 0\\ &\gamma = -\left(3.142/2\right)\end{aligned}$$

$$\begin{aligned}&Axe(26)\\ &\alpha = 0\\ &\beta = \left(3.142/2\right)\\ &\gamma = 0\end{aligned}$$

### 7.2. *Testing of the Proposed Method*

The method proposed is tested in Matlab-simulink environment. Simulink model developed for testing is as shown in Fig. 2 below:

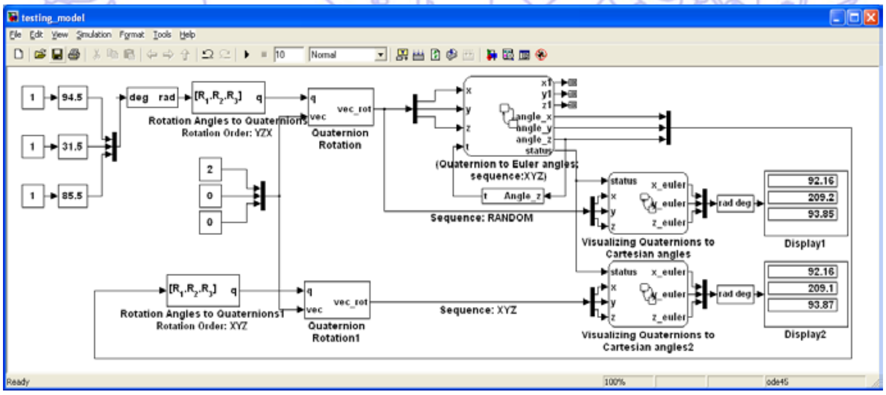

Fig. 2 Simulink model developed for testing of the proposed method

A vector $\mathbf{t} = [2 \quad 0 \quad 0]^T$ is rotated using three Euler angles. The Euler angles are first converted to quaternion with random rotation sequence using ‘Rotation Angles to Quaternions’ block which employs Eq. 27. Then, visualizing quaternion $\mathbf{t}^{'}$ is computed using ‘Quaternion Rotation’ block, which employs Eq. 28. The visualizing quaternion is sent to ‘Quaternion to Euler angles sequence: XYZ’ block, in which it is converted to Euler angles with sequence XYZ, using the proposed method, shown by Arrangement 1 in Sec. 7.1. The Euler angles produced by the proposed method (with XYZ rotation sequence) are converted to quaternion with equivalent rotation sequence and then visualizing quaternion $\mathbf{t}^{'}$ is computed using another ‘Quaternion Rotation’ block.

Copyright © TAETI

Two individual (but similar) ‘Visualizing Quaternion to Cartesian angles’ blocks are developed using the Arrangement 2 described in Sec. 7.1 and used to convert visualizing quaternions which are produced from random rotation sequence and from XYZ rotation sequence to Cartesian angles separately. The Cartesian angles computed using the two individual ‘Visualizing Quaternion to Cartesian angles’ blocks can be compared to validate the proposed method. The test is carried out for all 12 sets of Euler rotation sequences, covering all 26 regions. The results are presented in Table 1 below.

Table 1 Simulation results for 12 sets of Euler rotation sequences, covering all 26 regions.

| Random rotation sequence | Random rotation angles, in degrees | Cartesian angles produced from random rotation sequence, in degrees (Display1) | Cartesian angles produced from XYZ rotation sequence, in degrees (Display2) | Region | % Difference |
|---|---|---|---|---|---|
| ZYX | [-78.3 -33.3 85.5] | [-1.987 80.24 -11.4] | [-1.947 80.42 -11.4] | 1 | [2.01, -0.22, 0] |
| | [-20.7 -18.9 35.1] | [14.3 27.01 7.401] | [14.18 27.21 7.401] | 2 | [0.84, -0.74, 0] |
| ZYZ | [-36 266.4 -0.9] | [36.01 94.26 95.85] | [35.87 94.25 95.86] | 3 | [0.39, 0.01, -0.01] |
| | [-78.3 -18.9 108.9] | [-64.17 176.2 -172.2] | [-63.92 176.2 -172.2] | 4 | [0.39, 0, 0] |
| ZXY | [-6.3 125.1 107.1] | [-176.3 258.1 -163] | [-176.5 258.6 -163] | 5 | [-0.11, -0.19, 0] |
| | [-43.2 259.2 13.5] | [-171.2 -56.19 -13.09] | [-171.4 -56.88 -13.09] | 6 | [-0.12, -1.23, 0] |
| ZXZ | [-7.2 -46.5 -67.5] | [95.48 -16.86 72.47] | [95.74 -17.63 72.47] | 7 | [-0.27, -4.57, 0] |
| | [-302.4 144 -43.2] | [119.9 261.2 95.1] | [120 261.2 95.12] | 8 | [-0.08, 0, -0.02] |
| YXZ | [-50.4 0 0] | [0 50.4 0] | [0 50.4 0] | 9 | [0, 0, 0] |
| | [-136.8 0 0] | [0 136.8 180] | [0 136.8 180] | 10 | [0, 0, 0] |
| YXY | [151.2 0 7.2] | [0 248.4 180] | [0 248.4 180] | 11 | [0, 0, 0] |
| | [144 0 223.2] | [180 -7.2 0] | [180 -7.2 0] | 12 | [0, 0, 0] |
| YZX | [0 90 45] | [135 0 -90.01] | [135 0 -90.01] | 13 | [0, 0, 0] |
| | [0 90 -77.4] | [-12.6 0 -90.01] | [-12.6 0 -90.01] | 14 | [0, 0, 0] |
| YZY | [-7.2 -90 0] | [82.8 0 90.01] | [82.81 0 90.01] | 15 | [-0.01, 0, 0] |
| | [7.2 -90 0] | [97.21 0 90.01] | [97.24 0 90.01] | 16 | [-0.03, 0, 0] |
| XYZ | [0 0 -122.4] | [90.01 180 122.4] | [90.01 180 122.4] | 17 | [0, 0, 0] |
| | [0 0 -7.2] | [90.01 0 7.2] | [90.01 0 7.2] | 18 | [0, 0, 0] |
| XYX | [0 76.25 -90] | [-90.01 -1.33e-14 -76.25] | [-90.01 -1.38e-14 -76.25] | 19 | [0, -3.76, 0] |
| | [0 119.447 -90] | [-90.01 180 -119.5] | [-90.01 180 -119.4] | 20 | [0, 0, 0.08] |
| XZY | [0 0 90] | [180 -90 0] | [180 -90 0] | 21 | [0, 0, 0] |
| | [0 -90 0] | [90.01 0 90] | [90.01 0 90] | 22 | [0, 0, 0] |
| | [0 360 0] | [90.01 0 1.403e-14] | [90.01 0 1.403e-14] | 23 | [0, 0, 0] |
| XZX | [0 -180 0] | [90.01 180 180] | [90.01 180 180] | 24 | [0, 0, 0] |
| | [0 90 0] | [-90.01 0 -90] | [-90.01 0 -90] | 25 | [0, 0, 0] |
| | [5 90 -90] | [-9 90.01 -90.01] | [-9 90.01 -90.02] | 26 | [0, 0, -0.01] |

From Table 1, it can be seen that Cartesian angles computed using the two individual ‘Visualizing Quaternion to Cartesian angles’ blocks match each other without significant difference. Small variations between the angles are caused by accumulation of rounding errors due to the trigonometric functions.

## 8. Sample Program: Maneuvering of a Missile Flying in 3D Space

A program is developed using Matlab-simulink software to demonstrate application of quaternion in maneuvering orientation of a missile flying in 3D space. The missile can be rotated using 4 angles, where each angle represents rotation about Body x-axis, rotation about inertial X-axis, rotation about inertial Y-axis, and rotation about inertial Z-axis, respectively as mentioned in Sec. 7.1. Fig. 3 shows the program developed in Matlab-Simulink environment. State flow is utilized to program the 26 regions with respective formulas.

Copyright © TAETI

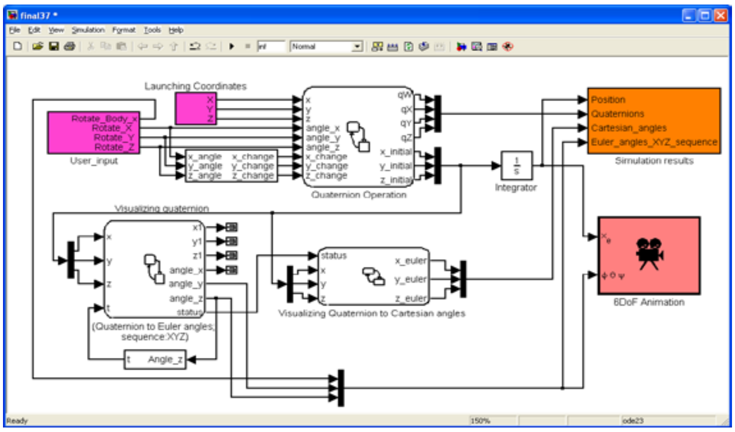

Fig. 3 Simulink program developed to visualize rotation of a body in 3 dimensional Euclidean space using Quaternion

‘Quaternion Operation’ block employs Eqs. 20-22 to calculate quaternion for each coordinate rotation, whenever user keys in new commands. Eqs. 24 and 25 are used to calculate visualizing quaternion for respective rotation commands. The visualizing quaternion is then sent to ‘Quaternion to Euler angles; sequence XYZ’ whereby it is converted to Euler angles with sequence XYZ, using the proposed method. The visualizing quaternion is also sent to ‘Visualizing Quaternion to Cartesian angles’ block, developed to convert the visualizing quaternion to Cartesian angles using Arrangement 2 shown in Sec. 7.1. Missile’s initial launching coordinates can be set in the ‘Launching Coordinates’ block before starting the simulation. Six degree of freedom (6DoF) block, which employs Euler rotation sequence of XYZ, is used to aid users to graphically see the maneuvering of the missile’s orientation as it flies in 3-dimensional Euclidean space. Simulation results can be viewed in ‘Simulation results’ block.

The aerodynamic forces and moment contributions are not considered in the model. The program represents kinematics of a missile and the missile’s projectile path can be altered by the user. The program developed can be used as an educational tool for first year Physics and Mathematics classes. The program can also be used as a medium to design and test attitude controllers. Sample program outputs are shown in Fig. 4.

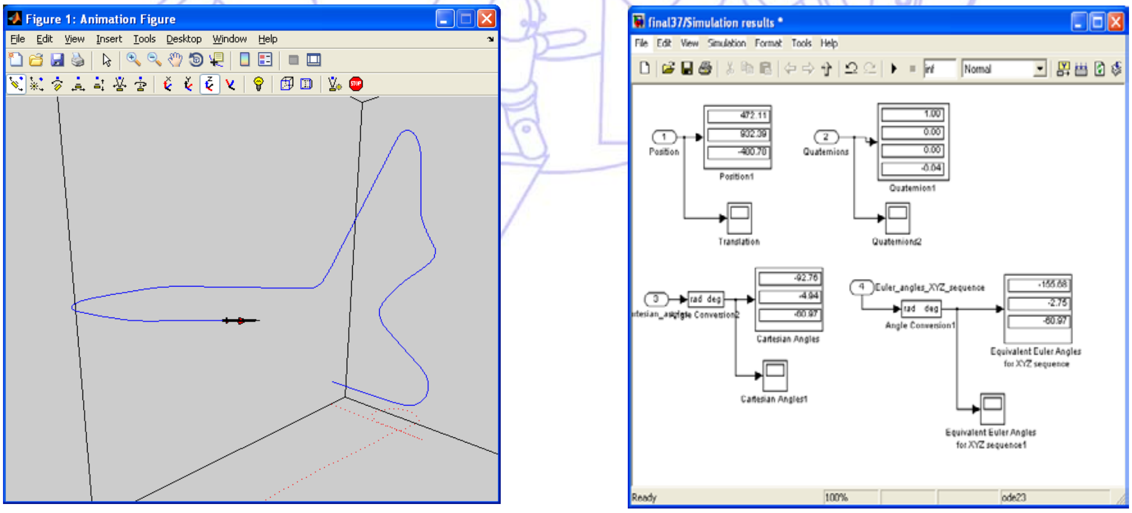
Figure 1: Animation Figure
File Edit View Insert Tools Desktop Window Help
final37/Simulation results *
File Edit View Simulation Format Tools Help
Normal
1
Position
472.11
932.39
-400.70
Position1
Translation
2
Quaternions
1.00
0.00
0.00
-0.04
Quaternion1
Quaternion2
3
rad deg
Cartesian_angles Conversion2
-92.76
-4.94
-60.97
Cartesian Angles
Cartesian Angles1
4
Euler_angles_XYZ_sequence
rad deg
Angle Conversion1
-155.68
-2.75
-60.97
Equivalent Euler Angles for XYZ sequence
Equivalent Euler Angles for XYZ sequence1
Ready
100%
ode23

Fig. 4 Sample program outputs (a) Missile’s orientation can be viewed in 3-dimensional Euclidean space and (b) Missile’s orientation are shown in separate block

Similar utility is provided in MATLAB/SIMULINK Aerospace toolbox, but it is limited to one particular rotation sequence at a given time. Main advantage of proposed program is that missile can be rotated at any random orientation (covering all 12 rotation sequences) at a given time, without facing any gimbal lock phenomena.

Copyright © TAETI

## 9. Conclusion and Future Work

Application of Euler angles and quaternion in representing rotations have been explained in this paper, in an attempt to expose readers to advantages of quaternion compared to Euler angles. A new method has been proposed in this paper, to convert a given quaternion (produced from arbitrary rotation sequence) to Euler angles with specific rotation sequence. This would enable quaternion to be compatible with pre-existing applications which utilizes specific Euler angles sequence. A sample program has been developed to demonstrate the application of quaternion in representing rotation using the proposed method. Main advantage of proposed program is that missile can be rotated at any random orientation (covering all 12 rotation sequences) at a given time, without facing any gimbal lock phenomena. The program can also be used as an education tool for undergraduate first year Physics and Mathematics classes.

Application of quaternion in control theories would require optimization process to be carried out to increase efficiency of the controller. It has been found out that optimization involving quaternion is impossible to be carried out using most of standard optimization techniques. Several optimization techniques involving quaternion have been proposed, but none are fully satisfactory. Thus, an in-depth study on optimization involving quaternion would be beneficial to fulfill the existing shortages.

## References

[1] Eric M. Jones and Paul Fjeld, “Gimbal angles, gimbal lock, and a forth gimbal for christmas,” http://www.hq.nasa.gov/alsj/gimbals.html, April 29, 2011.

[2] W.R. Hamilton, “On a new species of imaginary quantities connected with a theory of quaternions**,”** Proc. of the Royal Irish Academy, vol. 2, 1844, pp. 424-434.

[3] David H. Eberly, and Ken Shoemake, Game physics morgan kaufmann, pp. 507-544, 2004.

[4] James Jennings, “Towards an intuitive understanding of quaternion rotations,” Quantum, 2008.

[5] Andrew J. Hanson, Visualizing quaternions, Morgan Kaufmann Series, Canada, 2006.

[6] “Euler angles,”http://goo.gl/Uo0Nm.

[7] “Inertial navigation system,” http://goo.gl/u3JHI.

[8] James Diebel, “Representing attitude: Euler angles, Unit Quaternions, and Rotation Vectors,” Matrix, Citeseer, 2006.

[9] Sung-Sik Shin, Jung-Hoon Lee and Sug-Joon Yoon, “A comparison study of real-time solution to all attitude angles of an aircraft,” Journal of Mechanical Science and Technology, vol. 20, no. 3, pp.376-381, 2006.

[10] B. Alpern, L. Carter, M. Grayson, and C. Pelkie, “Orientation maps: techniques for visualizing rotations (a consumer’s guide),” IEEE Conference on Visualization, 1993, pp. 183-188.

[11] Andy “Maths - Euler angles,” http://goo.gl/D9Agt.

[12] R. Mukundan, “Quaternions: from classical mechanics to computer graphics, and beyond,” Proceedings of the 7th Asian Technology Conference in Mathematics, 2002, pp. 97-106.

[13] Jack B. Kuipers, ”Quaternions and rotation sequences,” Proc. of the International Conference on Geometry, Integrability and Quantization, September 1-10, 1999, Varna, Bulgaria, pp. 127-143.

[14] Matt Gravelle, “Quaternions and their applications to rotation in 3D space,” http://goo.gl/9bhoU.

[15] Charles F.F. Karney, “Quaternions in molecular modeling,” Journal of Molecular Graphics and Modelling , January 2007, pp. 595-604.

[16] J.D. Gibbon, “The three dimensional euler equations: where do we stand?,” Journal of Physica D, vol. 237, 2008. pp. 1894-1904.

[17] George P. Davailus, Brett A Newman, “The application of quaternion algebra to gyroscopic motion, Navigation, and Guidance,” AIAA, 2005-5945, 2005.

[18] Claus Haetinger, Marcelo Malheiros, Eduardo Dullius,and Márcio Kronbauer,”A quaternion application to control rotation movements in the three dimensional space of an articulate mechanical arm type robot built from low cost material as a supporting tool for teaching at the undergraduate level,” Global Congress on Engineering and Technology Education, March 13-16, 2005, Brazil.

Copyright © TAETI

[19] Kuipers, Jack B., Quaternions and rotation sequences: a primer with applications to orbits, Aerospace, and Virtual Reality. Princeton University Press, 1999.

[20] Pekka T. Laakso, Elements of the quaternion algebra with an application to electrical technology and rotations in 3D, Pekka T. Laakso, 2003.

[21] Yuan J.S.C, “Closed loop manipulator control with quaternion feedback,” IEEE Trans. On Robotics and Automation, vol.4, no.4, 1988, pp. 434-439.

[22] “Quaternion - Definition,” http://goo.gl/M0Xpg.

[23] Y.C. Fang, C.C. Hsieh, M.J. Kim, J.J. Chang and T.C. Woo, “Real time motion fairing with unit quaternions,” Computer-Aided Design, vol. 30, no. 3, 1998, pp. 191-198.

[24] Verena Elisabeth Kremer, “Quaternions and SLERP,” http://goo.gl/Q38wx.

[25] Kuipers, Jack B.,”Quaternions and Rotation Sequences A primer with applications to orbits, aerospace, and virtual reality,” Princeton University Press, Princeton, NJ, pp. 168-183, 1999.

[26] Noel H. Hughes, “Quaternion to Euler angle conversion for arbitrary rotation sequence using geometric methods,” http://goo.gl/s0IrV.

[27] Amy de Buitleir, ”Quaternions,” unpublished.

[28] From, P.J.; Gravdahl, J.T. "Representing attitudes as sets of frames," American Control Conference 2007, ACC ‘07, 9-13 July 2007, pp.2465-2472. URL: http://goo.gl/IdNTB.

[29] Logah Perumal, “Representing rotation in Simulink using quaternion,” The First International Conference on Engineering and Technology Innovation 2011 (ICETI 2011), Kenting, Taiwan, 11 – 15 November 2011, pp. 252.

[30] Logah Perumal, “Representing rotation in Simulink using quaternion,” Journal of Appl. Math. Inf. Sci. vol. 6, no. 6S, 2012, in press.

IJETI

Copyright © TAETI