Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L34

Representing Rotation in Simulink using Quaternion

*Logah Perumal**

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

Received: 10 May. 2013, Revised: 4 Sep. 2013, Accepted: 5 Sep. 2013
Published online: 1 Apr. 2014

Abstract: Euler angles were commonly used to represent rotation of a body, but it leads to an undesired phenomenon known as gimbal lock. This undesired phenomenon can be overcome by using quaternion, which was founded by Sir William Rowan Hamilton. Since its introduction, conversion between Euler angles and quaternion became essential; to enable quaternion to be compatible with applications developed using Euler angles and vice versa. In this work, a Simulink program is developed to demonstrate use of quaternion in representing rotation of a body in 3-dimensional space. Simulink program developed in this paper utilizes six degree of freedom animation block (employs Euler rotation sequence of XYZ), which enables users to graphically see and maneuver a missile in random orientation as it flies in a 3-dimensional Euclidean space. Random quaternion sequence is converted to Euler angles with XYZ sequence (in accordance to Euler rotation sequence utilized by the animation block) using new method known as sets of regions. Use of sets of regions enables quaternion to be applied in random sequence and gimbal lock phenomenon is avoided entirely.

Keywords: Euler angles, Quaternion, Rotation Sequence, Matlab Simulink, Orientation, Gimbal Lock, Sets of Region

1 Introduction

Rotation of a body in 2 - dimensional or 3 - dimensional spaces is often accomplished using Euler angles which were formulated by Leonhard Euler. One of drawbacks of using Euler angles to represent rotation is a phenomenon known as gimbal lock [1]. Gimbal lock is a phenomenon in which one of the rotation axes realigns with the other axis and eventually loses one degree of freedom. This phenomenon imposes restriction in representing rotation of a body, when Euler angles are used. In 1843, Sir William Rowan Hamilton introduced a new theory known as quaternion [2] and [3] to describe orientation of a body in space. Quaternion uses vector of four dimensions and quaternion products to represent rotation. Quaternion has many advantages over Euler angles and one of them is that the gimbal lock phenomenon is avoided. Upon realizing advantages of quaternion, many researchers showed interest and carried in-depth research on quaternion. One of fields which benefitted mostly from introduction of quaternion is in computer graphics and animation [4,5,6].

Practitioners tend to combine sequence of three rotations about the coordinate axes to accomplish any rotation [7]. 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. 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. 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].

Conversion between quaternion and Euler angles are later studied, in order to enable quaternion to be compatible with existing application developed using Euler angles. Nevertheless, conversion between quaternion and Euler angles are limited to certain rotation sequences, as described in upcoming section 4. In this work, a Simulink program is developed to demonstrate use of quaternion in representing rotation of a body in 3 dimensional space. Simulink program developed in this paper utilizes six degree of freedom animation block (employs Euler rotation sequence of XYZ), which enables viewers to graphically see and maneuver a missile in random orientation as it flies in a 3-dimensional

* Corresponding author e-mail: logah_z@yahoo.com, logah.perumal@mmu.edu.my

Euclidean space. Random quaternion sequence is converted to Euler angles with XYZ sequence (in accordance to Euler rotation sequence utilized by the animation block) using new method (sets of region)[9].

This paper is arranged as follows. Euler angles and quaternion are briefly introduced in sections 2 and 3, respectively. Various methods used in conversion between quaternion and Euler angles are described in section 4. Section 5 is used to test and validate proposed method while section 6 shows development of a simulink program utilizing the proposed method. The paper is finally concluded in section 7.

2 Euler Angles

Euler angles are used to rotate a body in mixed axis of rotation system in 3 dimensional Euclidean spaces. The three components of Euler angles are given as:

$$e = [\alpha, \beta, \gamma]^T \tag{1}$$

where e is Euler angle vector, α is rotation angle about x-axis, β is rotation angle about y-axis and γ is rotation angle about z-axis.

Consider $t \in \Re^3$, a vector in inertial coordinate system. The vector t, when subjected to rotation about a single axis (known as coordinate rotation) using one of the Euler angles would produce new vector which can be described in body coordinate system through the relation:

$$t^{'} = R_m(\theta)t \tag{2}$$

where $t^{'}$ is rotated vector described in body coordinate system, $R_m(\theta)$ is coordinate rotation matrix, t is vector in inertial coordinate system, m is rotation sequence and

$$\theta = \begin{cases} \alpha \; if \; m = x \\ \beta \; if \; m = y \\ \gamma \; if \; m = z \end{cases} \tag{3}$$

Coordinate rotation matrix for each axis is given by:

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

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

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

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:

$$R_{abc} = R_a(\theta)R_b(\theta)R_c(\theta) \tag{5}$$

where R_{abc} is rotation matrix with rotation sequence of abc, in which

$$a, b, c = \begin{cases} X \\ Y \\ Z \end{cases} \tag{6}$$

and

$$\theta = \begin{cases} \alpha \; if \; rotation \; axis = x \\ \beta \; if \; rotation \; axis = y \\ \gamma \; if \; rotation \; axis = z \end{cases} \tag{7}$$

3 Quaternion

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:

$$Q = [q_0, q]^T \tag{8}$$
$$q_0 = cos(\frac{\theta}{2})$$
$$q = sin(\frac{\theta}{2})n$$

where

$$n = \begin{cases} i \\ j \\ k \end{cases}$$

$$\theta = \begin{cases} \alpha \; if \; n = i \\ \beta \; if \; n = j \\ \gamma \; if \; n = k \end{cases}$$

Consider coordinate rotation of $t \in \Re^3$, a vector in inertial coordinate system. The vector t, when rotated by an angle about a single axis would produce a new vector which can be described in inertial coordinate system through the relation:

$$t^{'} = Q_n \times t \times Q_n^* \tag{9}$$

where $t^{'}$ is rotated vector described in inertial coordinate system, t is vector in inertial coordinate system, Q is quaternion vector with 4 elements, Q^* is conjugate of Q and n is rotation sequence.

Quaternion for rotation about individual axis is given by:

$$\begin{aligned} Q_i &= Q_X = cos(\tfrac{\alpha}{2}) + sin(\tfrac{\alpha}{2})i + sin(\tfrac{0}{2})j + sin(\tfrac{0}{2})k \\ &= cos(\tfrac{\alpha}{2}) + sin(\tfrac{\alpha}{2})i \\ Q_j &= Q_Y = cos(\tfrac{\beta}{2}) + sin(\tfrac{0}{2})i + sin(\tfrac{\beta}{2})j + sin(\tfrac{0}{2})k \\ &= cos(\tfrac{\beta}{2}) + sin(\tfrac{\beta}{2})j \\ Q_k &= Q_Z = cos(\tfrac{\gamma}{2}) + sin(\tfrac{0}{2})i + sin(\tfrac{0}{2})j + sin(\tfrac{\gamma}{2})k \\ &= cos(\tfrac{\gamma}{2}) + sin(\tfrac{\gamma}{2})k \end{aligned} \tag{10}$$

Eq. 10 can be rewritten in general form as:

$$\begin{aligned} t^{'} &= Q_n \times t \times Q_n^* \\ t^{'} &= R^Q t \end{aligned} \tag{11}$$

where $t^{'}$ is rotated vector described in inertial coordinate system, t is vector in inertial coordinate system and R^Q is quaternion rotation matrix.

The general quaternion rotation matrix (quaternion DCM) is given by:

$$[R^Q = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_1 & 2q_1q_2+2q_0q_2 \\ 2q_0q_3+2q_1q_2 & q_0^2-q_1^2+q_2^2-q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{12}$$

4 Conversion Between Quaternion and Euler Angles

Conversion of given Euler angles sequence to equivalent quaternion can be simply achieved by using appropriate formulae, but conversion of a given quaternion to Euler angles sequence is more complex and can be done in several ways.

Most commonly used method to convert given quaternion to equivalent Euler angles sequence is by first generating rotation matrix (also known as direction cosine 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 [5, 10, 11, 12, 13]. Author in [14] proposed a geometric method to convert quaternion to equivalent Euler angles sequence. Advantage of the geometric method is that conversion is done without involving any matrices, which causes reduction in computational time and complexity. In [15], 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. 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 [15]. A new method is proposed in [9] 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 new method uses sets of regions to convert quaternion to Euler angles and it is also proved to be valid for all rotation sequences, including rotations that are taking place around the same axis twice. Since quaternion rotates a body within inertial coordinate system, thus Cartesian frame can be used to represent attitude of a rotated body, by using Cartesian coordinates. A set of regions can be defined by imposing restrictions onto the Cartesian coordinates of **t'** in the Cartesian frame. There are three types of regions; octants, quadrants and axes. Total of 26 regions (consisting of 8 octants, 12 quadrants and 6 axes) can be defined in the Cartesian frame using relation:

$$x(n) \ for \ X = \begin{cases} <0 \\ >0 \\ =0 \end{cases}; \tag{13}$$

$$Y = \begin{cases} <0 \\ >0 \\ =0 \end{cases};$$

$$Z = \begin{cases} <0 \\ >0 \\ =0 \end{cases}$$

where x is region type given by:

$$x = \begin{cases} \text{Oct for Octant} \\ \text{Quad for Quadrant} \\ \text{Axe for Axis} \end{cases}$$

, n is region numbering and X, Y, Z are cartesian coordinates of $t^{'}$.

The 8 Octants are defined by:

Oct(1) for	$X>0$		Oct(5) for	$X<0$	
	$Y<0$			$Y<0$	
	$Z<0$			$Z>0$	
Oct(2) for	$X>0$		Oct(6) for	$X>0$	
	$Y>0$			$Y<0$	
	$Z<0$			$Z>0$	
Oct(3) for	$X<0$		Oct(7) for	$X>0$	
	$Y>0$			$Y>0$	
	$Z<0$			$Z>0$	
Oct(4) for	$X<0$		Oct(8) for	$X<0$	
	$Y<0$			$Y>0$	
	$Z<0$			$Z>0$	

The 12 Quadrants are defined by:

$Quad(9)$	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	$X>0$, $Y=0$, $Z>0$
$Quad(13)$	for	$X=0$, $Y<0$, $Z>0$
$Quad(14)$	for	$X=0$, $Y<0$, $Z<0$
$Quad(15)$	for	$X=0$, $Y>0$, $Z<0$
$Quad(16)$	for	$X=0$, $Y>0$, $Z>0$
$Quad(17)$	for	$X<0$, $Y>0$, $Z=0$
$Quad(18)$	for	$X>0$, $Y>0$, $Z=0$
$Quad(19)$	for	$X>0$, $Y<0$, $Z=0$
$Quad(20)$	for	$X<0$, $Y<0$, $Z=0$

The 6 Axes are defined by:

$Axe(21)$	for	$X=0$, $Y=0$, $Z>0$	$Axe(24)$	for	$X<0$, $Y=0$, $Z=0$
$Axe(22)$	for	$X=0$, $Y>0$, $Z=0$	$Axe(25)$	for	$X=0$, $Y<0$, $Z=0$
$Axe(23)$	for	$X>0$, $Y=0$, $Z=0$	$Axe(26)$	for	$X=0$, $Y=0$, $Z<0$

Euler angles can then be calculated based on the region in which the rotated vector t' lies. The vector t' is also known as visualizing quaternion, since it is used as a visualizing tool to specify the regions. Euler angles with XYZ sequence can then be computed using corresponding formulas for the respective region by using the following arrangements shown in Figure 1.

5 Testing of the Proposed Method

The method proposed in section 4 is tested in Matlab-simulink environment. Simulink model developed for testing is as shown in Figure 2. A vector $t' = [2 \;\; 0 \;\; 0]^T$ is rotated using three Euler angles. The Euler angles are first converted to quaternion with random rotation sequence using Rotation Angles to Quaternion Rotation Order: ijk block which employs equation (10). Visualizing quaternion is then computed using Quaternion Rotation block, which employs equations (11) and (12). 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.

The Euler angles produced by the proposed method (with XYZ rotation sequence) are then sent to another Rotation Angles to Quaternion Rotation Order: XYZ block whereby it is converted to quaternion with sequence XYZ and visualizing quaternion t' is computed using another Quaternion Rotation block. Visualizing quaternions t' computed using the two individual Quaternion Rotation 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 I. From Table I, it can be seen that visualizing quaternions computed using the two individual Quaternion Rotation blocks match each other without significant difference for all the cases. Small variations are caused by accumulation of rounding errors due to the trigonometric functions.

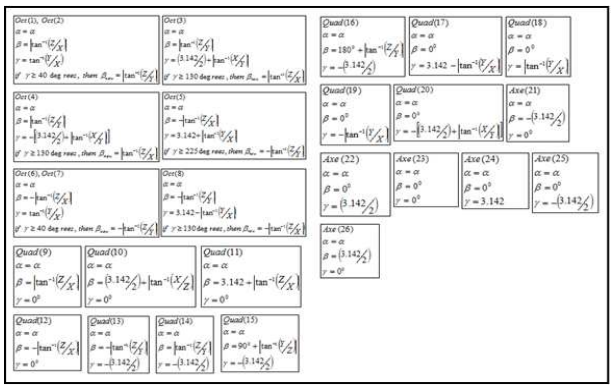

Figure 1: Sets of regions used in calculating Euler angles

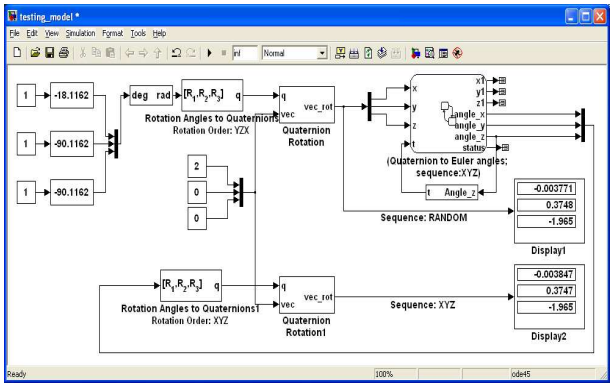

Figure 2: Simulink model developed for testing of the proposed method

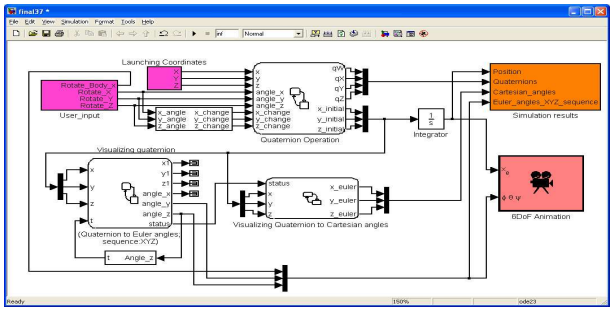

Figure 3: Simulink program developed to visualize rotation of a body in 3-dimensional Euclidean space using Quaternion

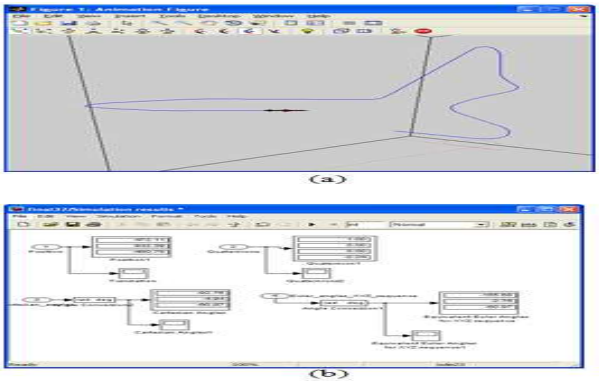

Figure 4: Sample program outputs (a) Missiles orientation can be viewed in 3-dimensional Euclidean space and (b) Numerical results

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

Random rotation sequence	Random rotation angles, in degrees	Visualizing quaternion produced from random rotation sequence (Display1)	Visualizing quaternion produced from XYZ rotation sequence (Display2)	Region	% Difference
	[-78.3 -33.3 85.5]	[0.339 -0.06833 -1.97]	[0.3325 -0.06702 -1.971]	1	[1.92 1.92 -0.05]
ZYX	[-20.7 -18.9 35.1]	[1.77 0.2299 -0.9023]	[1.767 0.2295 -0.9083]	2	[0.11 0.17 -0.66]
	[-36 266.4 -0.9]	[-0.12 1.174 -1.615]	[-0.1199 1.17 -1.618]	3	[0 0.34 -0.19]
ZYZ	[-78.3 -18.9 108.9]	[-0.4016 -1.85 -0.6439]	[-0.401 -1.846 -0.6573]	4	[0.15 0.22 -2.08]
	[-6.3 125.1 107.1]	[-0.4129 -0.1262 1.953]	[-0.3956 -0.1211 1.957]	5	[0.04 4.04 -0.20]
ZXY	[-43.2 259.2 13.5]	[1.104 -0.2565 1.648]	[1.084 -0.252 1.662]	6	[0.02 1.75 -0.85]
	[-7.2 -46.5 -67.5]	[0.5999 1.899 0.1818]	[0.5997 1.898 0.1906]	7	[0 0.05 -4.84]
ZXZ	[-302.4 144 -43.2]	[-0.154 1.729 0.9926]	[-0.1545 1.728 0.9955]	8	[0 0.06 -0.29]
	[-50.4 0 0]	[1.275 0 -1.541]	[1.275 0 -1.541]	9	[0 0 0]
YXZ	[-136.8 0 0]	[-1.458 0 -1.369]	[-1.458 0 -1.369]	10	[0 0 0]
	[151.2 0 7.2]	[-1.618 0 -1.176]	[-1.618 0 -1.175]	11	[0 0 0.09]
YXY	[144 0 223.2]	[1.984 0 0.2507]	[1.984 0 0.2507]	12	[0 0 0]
	[0 90 45]	[0 -1.414 1.414]	[0 -1.414 1.414]	13	[0 0 0]
YZX	[0 90 -77.4]	[0 -0.4363 -1.952]	[0 -0.4363 -1.952]	14	[0 0 0]
	[-7.2 -90 0]	[0 1.984 -0.2507]	[0 1.984 -0.2503]	15	[0 0 0.16]
YZY	[7.2 -90 0]	[0 1.984 0.2507]	[0 1.984 0.2515]	16	[0 0 -0.32]
	[0 0 -122.4]	[-1.072 1.689 0]	[-1.072 1.688 0]	17	[0 0.06 0]
XYZ	[0 0 -7.2]	[1.984 0.2507 0]	[1.984 0.2507 0]	18	[0, 0, 0]
	[0 76.25 -90]	[0.4754 -1.943 0]	[0.4754 -1.943 0]	19	[0, 0, 0]
XYX	[0 119.447 -90]	[-0.9832 -1.742 0]	[-0.9825 -1.742 0]	20	[0, 0, 0]
	[0 0 90]	[0 0 2]	[0 0 2]	21	[0, 0, 0]
XZY	[0 -90 0]	[0 2 0]	[0 2 0]	22	[0, 0, 0]
	[0 360 0]	[2 0 0]	[2 0 0]	23	[0, 0, 0]
	[0 -180 0]	[-2 0 0]	[-2 0 0]	24	[0, 0, 0]
XZX	[0 90 0]	[0 -2 0]	[0 -2 0]	25	[0, 0, 0]
	[0 90 -90]	[0 0 -2]	[0 0 -2]	26	[0, 0, 0]

6 Simulink Program

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. Figure 3 shows the program developed in Matlab-Simulink environment. State flow is utilized to program the 26 regions with respective formulas.

Quaternion Operation block employs equation (10) to calculate quaternion for each coordinate rotation, whenever user keys in new commands. Equations (11) and (12) 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 sets of region method. The visualizing quaternion is also sent to Visualizing Quaternion to Cartesian angles block, developed to convert the visualizing quaternion to Cartesian angles. Missiles 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 missiles 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 missiles 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 figure 4.

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.

7 Conclusion

A sample program has been successfully developed to demonstrate application of quaternion in representing rotation of a body in 3-dimensional space. Simulink program developed in this paper utilizes six degree of freedom animation block (employs Euler rotation sequence of XYZ), which enables viewers to graphically see and maneuver a missile in random orientation as it flies in a 3-dimensional Euclidean space. Use of sets of regions enables quaternion to be applied in random sequence and gimbal lock phenomenon is avoided entirely. The program can also be used as an education tool for undergraduate first year Physics and Mathematics classes.

References

[1] Eric, M. J. and Paul, F. Apollo Lunar Surface Journal. Gimbal Angles, Gimbal Lock, and a Fourth Gimbal for Christmas. Accessed online at: http://www.hq.nasa.gov/alsj/gimbals.html.

[2] Hamilton, W. R. On a new species of imaginary quantities connected with a theory of quaternions. Proceedings of the Royal Irish Academy 2. November 13th, Dublin, Ireland, (1843).

[3] Hamilton, W. R. On quaternions. Proceedings of the Royal Irish Academy 3. November 11th, Dublin, Ireland, (1844).

[4] Mukundan, R. Quaternions - From Classical Mechanics to Computer Graphics and Beyond. Proceedings of Asian Technology Conference in Mathematics, Malaysia, (2002) .

[5] Kuipers, J. B. Quaternions and Rotation Sequences A Primer with Applications to Orbits, Aerospace, and Virtual Reality, New Jersey: Princeton Publication, (1999).

[6] Dam, E. B., Martin, K. Quaternions, Interpolation and Animation. (Department of Computer Science, University of Copenhagen, Denmark), (1998).

[7] Alpern, B., Carter, L., Grayson, M., Pelkie, C. Orientation maps: Techniques for visualizing rotations (a consumers guide). IEEE Conference on Visualization, San Jose, California, USA, 25-29 (1993).
[8] Diebel, J, Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. (Stanford University, USA), (2006).
[9] Logah Perumal Conversion of Quaternion to Euler Angles using Sets of Regions, INTI Journal, **2011**, 109-124. (2011).
[10] 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, **35**, (1996).
[11] Dam, Eric B., Koch, Martin, Quaternions, Interpolation and Animation. (Department of Computer Science, University of Copenhagen, Denmark), **7**, 93 (1998).
[12] James Diebel, Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Accessed online at www.astro.rug.nl/software/kapteyn/_downloads/attitude.pdf
[13] Martin John Baker, Maths - Euler Angles. Assessed online at http://www.euclideanspace.com/maths/geometry/rotations/euler/andygoldstein.htm
[14] Noel H. Hughes, Quaternion to Euler Angle Conversion for Arbitrary Rotation Sequence Using Geometric Methods. Accessed online at noelhughes.net/uploads/quat_2_euler_paper_ver2-1.pdf.
[15] Amy de Buitleir. Quaternions. Accessed online at http://www.euclideanspace.com/index.html.

Logah Perumal received his Bachelor degree in Mechanical Engineering and Master in Mechanical Engineering from Universiti Tenaga Nasional, Malaysia in 2006 and 2009, respectively. Currently working as a lecturer with Faculty of Engineering and Technology at Multimedia University, Malaysia. His research interests include numerical methods, finite element method, computation and fuzzy logic.