THE FEASIBILITY OF TAIT-BRYAN EULER ANGLES IN MODERN TECHNOLOGY

Mathematics Extended Essay

Abstract

In this extended essay, I compared the feasibility of using Tait-Bryan Euler Angles in performing rotations in modern day devices. I first displayed the formula and derived it to understand its fundamental basis. Then I examined present applications of Tait-Bryan Euler Angles. I then investigated a problem in the use of the formula called gimbal lock. After investigating the pros and cons of the formula, I understood that while Tait-Bryan Euler Angles' simplistic nature visually appeal to even the general public gimbal lock is too large a problem to permit its use. Thus, in modern technology other rotational formulas should be used to perform rotations while Tait-Bryan Euler Angles should be used to show the net rotations on user interfaces.

Word Count: 120

Table of Contents

Introduction

Tait-Bryan Euler Angles are one of the earliest rotational formulas. Succeeding the formula, several other rotational methods have been then developed and used instead of Tait-Bryan Euler Angles. In response to this shift, I wanted to see why Tait-Bryan Euler Angles have lost their ground in the realm of technology. Even more I want to see if the shift is truly just and if the future of the formula is actually as bleak as it seems.

Tait-Bryan Euler Angles

Tait-Bryan Euler Angles are a counterintuitive method of rotation in that to represent an orientation, the coordinate system axes themselves are rotated and the values of the fixed vectors within the system are calculated – this is called intrinsic rotation. For example, if a position vector wanted to be rotated clockwise by π about the Z-axis, the X-axis and Y-axis are themselves rotated counter-clockwise by π (while the position vector remains stationary) and the location of the new position vector is calculated. The formula is derived from Euler's Rotation Theorem which states that an arbitrary rotation can be represented by three consecutive rotation about three independent axes (Euler Angles n.d.).

The three axes used are the X, Y, and Z axes. With the formula, a rotation about a defined axis is broken down into three rotations. To rotate a vector, the Z, Y, and X axes are consecutively rotated and the new value of the vector is calculated (Euler Angles n.d.). The convention used throughout this paper is that rotation is first about the Z-axis, then the Y-axis, then the X-axis; this is important as 3-dimentional rotation is not communitive.

The formula for this method of rotation is:

$$\begin{bmatrix} x_{rotated} \\ y_{rotated} \\ z_{rotated} \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\Psi)\cos(\Theta) & \cos(\Theta)\sin(\Psi) & -\sin(\Theta) \\ \cos(\Psi)\sin(\phi)\sin(\Theta) - \cos(\phi)\sin(\Psi) & \cos(\phi)\cos(\Psi) + \sin(\phi)\sin(\Psi)\sin(\Theta) & \cos(\Theta)\sin(\phi) \\ \sin(\phi)\sin(\Psi) + \cos(\phi)\cos(\Psi)\sin(\Theta) & \cos(\phi)\sin(\Psi)\sin(\Theta) - \cos(\Psi)\sin(\phi) & \cos(\phi)\cos(\Theta) \end{bmatrix} \begin{bmatrix} x_{initial} \\ y_{initial} \\ z_{initial} \end{bmatrix}$$

("Understanding Euler Angles", n.d.)

In the formula, ϕ (phi), Θ (theta), Ψ (psi) represents the angle of rotation **of the axes** about the X, Y, and Z axis, respectively. The diagram below shows the axes of rotation.

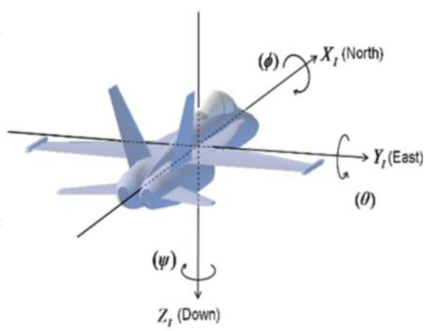


Figure 1: The three types of rotations an orientation can undergo ("Understanding Euler Angles", n.d.)

Deriving Tait-Bryan Euler Angles

Tait-Bryan Euler Angles are used because of their basis in fundamental trigonometry. By demonstrating this basis, an in-depth cost-benefit analysis of using this formula can be shown.

In Tait-Bryan Euler Angles rotation about the Z-axis will be called yaw rotation, rotation about the Y-axis will be called pitch rotation, and rotation about the X-axis will be called roll rotation (Understanding Euler Angles, n.d).

Figure 2 shows rotation yaw counter clockwise by angle Ψ about the Z-axis, rotation pitch counter clockwise by angle Θ about the Y-axis, and rotation roll counter clockwise by angle ϕ about the X-axis (Understanding Euler Angles, n.d.). It is important to note that after rotation yaw, the orientation's X and Y axes will be unaligned with the earth-fixed[1] X and Y axes. There are many conventions of the axis directions for the earth-fixed axes. In this essay the north-east-down (NED) convention will be used: for the earth-fixed axes the X-axis points north, the Y-axis points east, and the Z-axis points vertically downward.

The initial orientation of an object is called the "Inertial Frame" (I.F) which is aligned with the earth-fixed axes (Understanding Euler Angles, n.d.). After the orientation has undergone yaw, it is known as the "Vehicle-1 Frame" (V1); after the orientation has undergone pitch, it is known as the "Vehicle-2 Frame" (V2); and after the orientation has undergone roll (the last rotation), the final orientation is known as the "Body Frame" (B.F) (Understanding Euler Angles, n.d.).

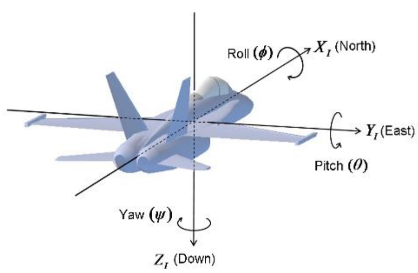


Figure 2: The three types of rotations an orientation can undergo ("Understanding Euler Angles", n.d.)

Yaw – from the Inertial Frame to the Vehicle 1 Frame

The first rotation, yaw, is about the Z-axis by the angle Ψ. The X and Y axes will rotate about the Z-axis by angle Ψ and the fixed vectors will be post-multiplied with the rotation matrix $\begin{bmatrix} \cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$ to find the new coordinates ("Understanding Euler Angles, n.d.). Note that since the axes were rotated around the Z-axis, the values of the x and y coordinates of the fixed vectors change while the values of the z coordinates remain the same.

[1] The untransformed coordinate system axes

Mathematically, $\begin{bmatrix} x_{V1} \\ y_{V1} \\ z_{V1} \end{bmatrix} = \begin{bmatrix} \cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{I.F} \\ y_{I.F} \\ z_{I.F} \end{bmatrix}$ ("Understanding Euler Angles, n.d.).

In figure 3, the grey axes represent the I.F. axes while the red axes represent the V.1. frame axes after yaw rotation has been performed. Notice how the X and Y axes rotates, but the Z-axis remains fixed as rotation was about it.

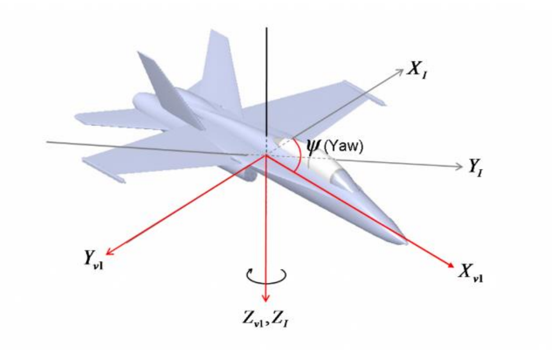


Figure 3: Yaw rotation been applied to the Inertial Frame ("Understanding Euler Angles", n.d.)

Pitch – from the Vehicle 1 Frame to Vehicle 2 Frame

The second rotation, pitch, is about the Y-axis by the angle Θ. The X and Z axes will rotate about the Y-axis by angle Θ and the fixed vectors will be post-multiplied with the rotation matrix $\begin{bmatrix} \cos\Theta & 0 & -sin\Theta \\ 0 & 1 & 0 \\ sin\Theta & 0 & cos\Theta \end{bmatrix}$ to find the new coordinates ("Understanding Euler Angles, n.d.). Note that since the axes were rotated around the Y-axis, the values of the x and z coordinates of the fixed vectors change while the values of the y coordinates remain the same.

Mathematically, $\begin{bmatrix} x_{V2} \\ y_{V2} \\ z_{V2} \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & -sin\Theta \\ 0 & 1 & 0 \\ sin\Theta & 0 & cos\Theta \end{bmatrix} \begin{bmatrix} x_{V1} \\ y_{V1} \\ z_{V1} \end{bmatrix}$ ("Understanding Euler Angles, n.d.).

In figure 4, the grey axes represent the Vehicle 1 Frame axes while the red axes represent the Vehicle 2 Frame axes after pitch rotation has been performed. Notice how the Y-axis is fixed as rotation was performed about it.

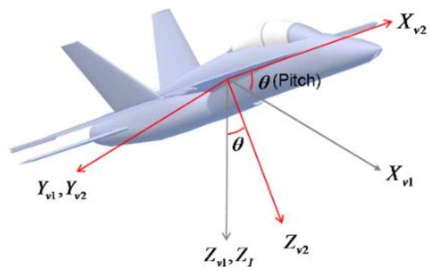


Figure 4: Pitch rotation being applied to the vehicle 1 frame ("Understanding Euler Angles", n.d.)

Roll – from the vehicle 2 frame to the body frame

The last rotation, roll, is about the X-axis by angle ϕ. The Y and Z axes will rotate about the X-axis by angle ϕ and the fixed vectors will be post-multiplied with the rotation matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix}$ to find the new coordinates ("Understanding Euler Angles, n.d.). Note that since the axes were rotated around the Z-axis, the values of the x and y coordinates of the fixed vectors change while the values of the z coordinates remain the same.

Mathematically, $\begin{bmatrix} x_{b.f} \\ y_{b.f} \\ z_{b.f} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x_{V2} \\ y_{V2} \\ z_{V2} \end{bmatrix}$ ("Understanding Euler Angles, n.d.).

In figure 5, the grey axes represent the Vehicle 2 Frame axes while the red axes represent the Body Frame axes after roll rotation has been performed. Notice how the X-axis is fixed as rotation was performed about it.

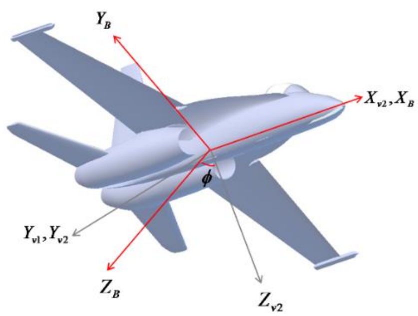


Figure 5: Roll rotation being applied to the Vehicle 2 Frame ("Understanding Euler Angles", n.d.)

The complete rotation

The summative matrix multiplication steps from the Inertial Frame to the body frame is:

$$\begin{bmatrix} x_{B.F} \\ y_{B.F} \\ z_{B.F} \end{bmatrix} = [R_{Roll}][R_{Pitch}][R_{Yaw}] \begin{bmatrix} x_{I.F} \\ y_{I.F} \\ z_{I.F} \end{bmatrix} = [R_{combined}] \begin{bmatrix} x_{I.F} \\ y_{I.F} \\ z_{I.F} \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\Psi)\cos(\Theta) & \cos(\Theta)\sin(\Psi) & -\sin(\Theta) \\ \cos(\Psi)\sin(\phi)\sin(\Theta) - \cos(\phi)\sin(\Psi) & \cos(\phi)\cos(\Psi) + \sin(\phi)\sin(\Psi)\sin(\Theta) & \cos(\Theta)\sin(\phi) \\ \sin(\phi)\sin(\Psi) + \cos(\phi)\cos(\Psi)\sin(\Theta) & \cos(\phi)\sin(\Psi)\sin(\Theta) - \cos(\Psi)\sin(\phi) & \cos(\phi)\cos(\Theta) \end{bmatrix} \begin{bmatrix} x_{I.F} \\ y_{I.F} \\ z_{I.F} \end{bmatrix}$$

("Understanding Euler Angles", n.d.)

Proof of the Constituent Matrices

Proof of Yaw Rotation

Consider yaw rotatation in figure 6 where the X and Y axes rotate counterclockwise by angle Ψ to X_0 and Y_0 (the Z-axis is not shown for clarity). A given fixed vector V with coordinates (a,b,c) will have the coordinates (a_0,b_0,c) after the rotation. Note that the z-coordinate of V before and after the rotation are the same since rotation was about the Z-axis.

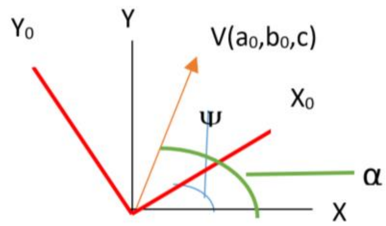


Figure 6: Yaw rotation

cos α = $\frac{a}{|V|}$ **Eq. 1:** a= |V|cos α

cos $(\alpha - \Psi)$ = $\frac{a_0}{|V|}$

a_0 = $|V|(cos(\alpha - \Psi))$

= $|V| \cos(\alpha)\,(cos(\Psi) + |\mathrm{V}| \sin(\alpha) \sin(\Psi))$

(Substitute variables using Eq.1 and Eq 2)

= a $cos(\Psi)$ + b $\sin(\Psi)$

Sin α = $\frac{b}{|V|}$ **Eq. 2:** b= |V|sin α

Sin $(\alpha - \Psi)$ = $\frac{b_0}{|V|}$

b_0 = $|V|(sin(\alpha - \Psi))$

= $|V|(sin(\alpha) \cos(\Psi) - |\mathrm{V}|\sin(\Psi) \cos(\alpha)$

(Substitute variables using Eq.1 and Eq 2)

= a $cos(\Psi)$ – b sin (Ψ)

∴ V(a $cos(\Psi)$ + b $\sin(\Psi)$, a $cos(\Psi)$ – b sin (Ψ),c) = Yaw_Ψ(a,b,c) where Yaw_Ψ is the rotation function by angle Ψ (“A short derivation…”, 2011).

In matrix form, $Yaw_\Psi \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \mathrm{a}\, cos(\Psi) + \mathrm{b} \sin(\Psi) \\ \mathrm{a}\, cos(\Psi) - \mathrm{b} \sin(\Psi) \\ c \end{bmatrix}$

To convert the transformation function Yaw_Ψ into matrix form, the transformation is applied to the identity matrix which is then post multiplied with the vector.

$$Yaw_\Psi \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \left[Yaw_\Psi \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} Yaw_\Psi \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} Yaw_\Psi \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Therefore we have reached the original yaw transformation equation in matrix form:

$$\begin{bmatrix} x_{V1} \\ y_{V1} \\ z_{V1} \end{bmatrix} = \begin{bmatrix} \cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{I.F} \\ y_{I.F} \\ z_{I.F} \end{bmatrix}$$

Proof of Pitch Rotation

Consider pitch rotatation in figure 7 where the X and Z axes rotate counterclockwise by angle Θ to X_0 and Z_0 (the Y-axis is not shown for clarity). A given fixed vector V with coordinates (a,b,c) will have the coordinates (a_0,b,c_0) after the rotation. Note that the y-coordinate of V before and after the rotation are the same since rotation was about the Y-axis.

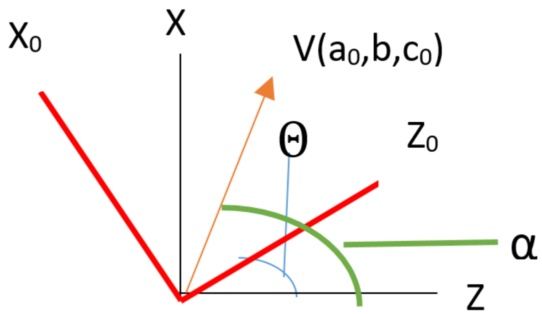


Figure 7: Pitch rotation

$\cos \alpha = \frac{c}{|V|}$ **Eq. 3:** c= |V|cos α

$\cos (\alpha - \Theta) = \frac{c_0}{|V|}$

$c_0 = |V|(cos(\alpha - \Theta))$

$= |V| \cos(\alpha) (cos(\Theta) + |V| \sin(\alpha) \sin(\Theta))$

(Substitute variables using Eq.3 and Eq 4)

$= c \, cos(\Theta) + a \sin(\Theta)$

$\text{Sin } \alpha = \frac{a}{|V|}$ **Eq. 4:** a= |V|sin α

$\text{Sin } (\alpha - \Theta) = \frac{a_0}{|V|}$

$b_0 = |V|(sin(\alpha - \Theta))$

$= |V|(sin(\alpha) \cos(\Theta) - |V| \sin(\Theta) \cos(\alpha)$

(Substitute variables using Eq.3 and Eq 4)

$= a \, cos(\Theta) - c \sin(\Theta)$

∴ V(c $cos(\Theta)$ + a $\sin(\Theta)$, b, a $cos(\Theta)$ – c $\sin(\Theta)$) = $Pitch_\Theta$(a,b,c) where $Pitch_\Theta$ is the rotation function by angle Θ (“A short derivation…”, 2011).

In matrix form, $Pitch_\Theta \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} c \, cos(\Theta) + a \sin(\Theta) \\ b \\ a \, cos(\Theta) - c \sin(\Theta) \end{bmatrix}$

To convert the transformation function $Pitch_\Theta$ into matrix form, the transformation is applied to the identity matrix which is then post multiplied with the vector.

$$Pitch_{\Theta}\begin{bmatrix}a\\b\\c\end{bmatrix}=\left[Pitch_{\Theta}\begin{bmatrix}1\\0\\0\end{bmatrix}Pitch_{\Theta}\begin{bmatrix}0\\1\\0\end{bmatrix}Pitch_{\Theta}\begin{bmatrix}0\\0\\1\end{bmatrix}\right]\begin{bmatrix}a\\b\\c\end{bmatrix}=\begin{bmatrix}\cos\Theta & 0 & -sin\Theta\\0 & 1 & 0\\sin\Theta & 0 & cos\Theta\end{bmatrix}\begin{bmatrix}a\\b\\c\end{bmatrix}$$

Therefore we have reached the original pitch transformation equation in matrix form:

$$\begin{bmatrix}x_{V2}\\y_{V2}\\z_{V2}\end{bmatrix}=\begin{bmatrix}\cos\Theta & 0 & -sin\Theta\\0 & 1 & 0\\sin\Theta & 0 & cos\Theta\end{bmatrix}\begin{bmatrix}x_{V1}\\y_{V1}\\z_{V1}\end{bmatrix}$$

Proof of Roll Rotation

Consider roll rotatation in figure 8 where the Y and Z axes rotate counterclockwise by angle ϕ to Y_0 and Z_0 (the X-axis is not shown for clarity). A given fixed vector V with coordinates (a,b,c) will have the coordinates (a,b_0,c_0) after the rotation. Note that the x-coordinate of V before and after the rotation are the same since rotation was about the X-axis.

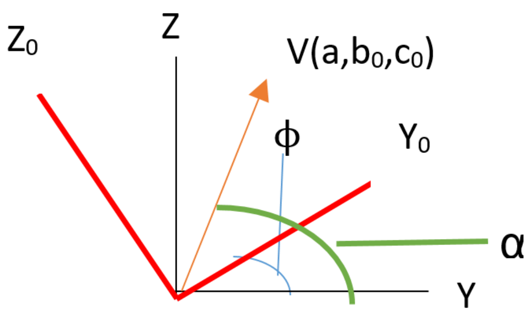


Figure 8: roll rotation

cos α = $\frac{b}{|V|}$ **Eq. 5:** b= |V|cos α

cos $(\alpha - \Theta)$ = $\frac{b_0}{|V|}$

b_0 = $|V|(cos(\alpha - \phi))$

= $|V|\cos(\alpha)\,(cos(\phi) + |V|\sin(\alpha)\sin(\phi))$

(Substitute variables using Eq.5 and Eq 6)

= b $cos(\phi)$ + c sin(ϕ)

Sin α = $\frac{c}{|V|}$ **Eq. 6:** c= |V|sin α

Sin $(\alpha - \Theta)$ = $\frac{c_0}{|V|}$

c_0 = $|V|(sin(\alpha - \phi))$

= $|V|(sin(\alpha)\cos(\phi) - |V|\sin(\phi)\cos(\alpha)$

(Substitute variables using Eq.5 and Eq 6)

= c $cos(\phi)$ – b sin (ϕ)

∴ V(a, b $cos(\phi)$ + c sin(ϕ), c $cos(\phi)$ – b sin (ϕ)) = $Roll_{\phi}$(a,b,c)   where $Roll_{\phi}$ is the rotation function by angle ϕ (“A short derivation…”, 2011).

In matrix form, $Roll_{\phi}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix}$

To convert the transformation function $Roll_{\phi}$ into matrix form, the transformation is applied to the identity matrix which is then post multiplied with the vector.

$$Roll_{\phi}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \left[Roll_{\phi}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} Roll_{\phi}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} Roll_{\phi}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Therefore we have reached the original roll transformation equation in matrix form:

$$\begin{bmatrix} x_{b.f} \\ y_{b.f} \\ z_{b.f} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x_{V2} \\ y_{V2} \\ z_{V2} \end{bmatrix}$$

Applications of Tait-Bryan Euler Angles

Tait-Bryan Euler Angles are generally used in devices for younger audiences because of its ease of use; the derivation above demonstrated the basis in trigonometry of Euler Angles. Additionally, the decomposition of a rotation about an arbitrary axis into three rotation about familiar axes proves very valuable to the every day person. Below is an image of an online user-friendly 3-D model of a plane. In the simulation, users can rotate the plane around it’s three coordinate axes. This access to the Euler Angles users have allow them to easily inspect the plane from all angles.

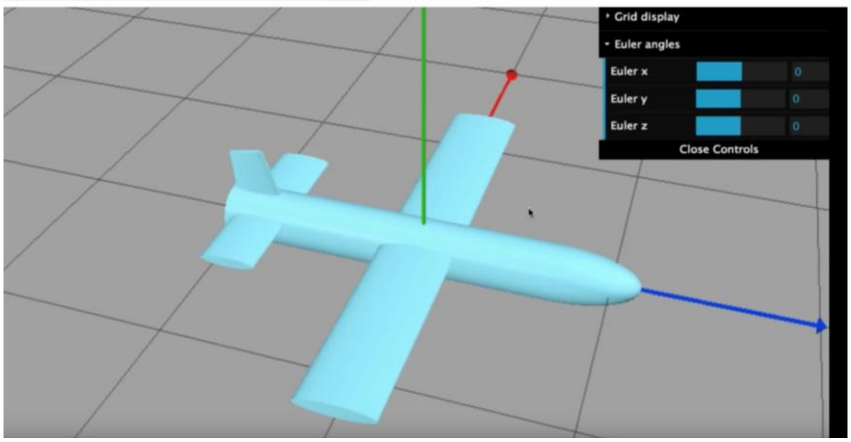


Figure 9: A 3-D model employing Euler Angles (“Euler Angles…”, 2015)

Tait-Bryan Euler Angles are used in a wide variety of fields: aeronautics, virtual reality, robotics and more. Due to its nature of intrinsic rotation, the formula is commonly used to rotate stationary vectors due to changing orientations such as in first-person technology.

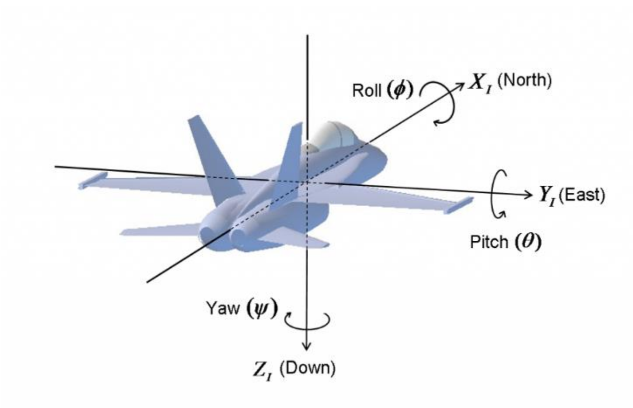
The trigonometric basis of Tait-Bryan Euler Angles encourages it to be the first type of rotation taught in engineering classes. Even more, their simple nature makes it the attractive choice for rudimentary programmers in their rotation of objects.

Tait-Bryan Euler Angles are also used in some sophisticated devices. Its nature of intrinsic rotation encourages its use in rotating stationary vectors due to changing orientations such as in first-person technology. Redshift Labs is a company that produces robotic technology. One of the items they produce is the UM7 Orientation Sensor which utilizes Tait-Bryan Euler Angles. They describe it as “an Attitude and Heading Reference System (AHRS) that contains a three-axis accelerometer, rate gyro, and magnetometer…[that] produce[s] attitude and heading estimates” (UM7 Orientation Sensor, n.d.). This gadgets’ use in actual robots to track its location and attitude shows the utility of Tait-Bryan Euler Angles.

More commonly seen, however, are devices that display the orientation of a device using Euler Angles but perform the rotations using other methods. Such is seen in CH Robotic orientation sensors which perform the rotations using quaternions. (“Understanding Euler Angles…”, n.d.). This suggests that while Euler Angles are user-friendly, they are not feasible as a method of rotation. This is largely attributed to the problem of gimbal lock.

Gimbal Lock

Gimbal Lock is one of the greatest problems when using Euler Angles. This problem occurs when $\Theta = \pi/2 + k\pi$ where k is an integer (Jia, 2016).



("Understanding Euler Angles", n.d.)

First, I will explain the problem logically.

Scenario One: Suppose the orientation of a plane as shown above. Suppose the coordinate axes rotate yaw by $\Psi = \pi/3$ (an arbitrary value), then pitch by $\Theta = \pi/2$.

Scenario Two: Then suppose another instance from the same starting position where the coordinate axes turns pitch by $\Theta = \pi/2$ then perform roll rotation by $\phi = -\pi/3$.

With some visualization, you can see that the location of the plane in scenario one and two are the same; i.e the net rotation of the coordinate axes are the same. The arbitrary net rotation is no longer uniquely represented but can be represented by first yawing then pitching or first pitching then rolling. This problematic case only occurs when $\Theta = \pi/2 + k\pi$ (Jia, 2016).

Now I will explain gimbal lock mathematically.

Suppose scenario one that has yaw rotation by $\Psi = \pi/3$ and no roll rotation. The rotation matrix looks like:

$$\begin{bmatrix} 0 & 0 & -1 \\ \sin(-\pi/3) & \cos(-\pi/3) & 0 \\ \cos(-\pi/3) & \sin(\pi/3) & 0 \end{bmatrix}$$

Suppose scenario two that has roll rotation by $\phi = -\pi/3$. The rotation matrix looks like:

$$\begin{bmatrix} 0 & 0 & -1 \\ \sin(-\pi/3) & \cos(-\pi/3) & 0 \\ \cos(-\pi/3) & \sin(\pi/3) & 0 \end{bmatrix}$$

One can see that the rotation matrices are the same. This scenario can be generalized for $\Psi = -\phi$.

When $\Theta = \pi/2$, the rotation matrix, S, looks like:

$$S = \begin{bmatrix} 0 & 0 & -1 \\ \cos(\Psi)\sin(\phi) - \cos(\phi)\sin(\Psi) & \cos(\phi)\cos(\Psi) + \sin(\phi)\sin(\Psi) & 0 \\ \sin(\phi)\sin(\Psi) + \cos(\phi)\cos(\Psi) & \cos(\phi)\sin(\Psi) - \cos(\Psi)\sin(\phi) & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & -1 \\ \sin(\phi - \Psi) & \cos(\phi - \Psi) & 0 \\ \cos(\phi - \Psi) & \sin(\Psi - \phi) & 0 \end{bmatrix} \text{(Jia, 2016)}$$

From S_{12}, S_{13}, S_2, S_{23} it is observable that ϕ and $(-\Psi)$ are interchangeable to produce the same rotation matrix (i.e. the same net rotation).

The fact that yaw + pitch can be achieved by pitch + roll is problematic: there is a loss of a degree of freedom. When $\Theta = \pi/2 + k\pi$, yaw and roll are interchangeable because pitching by $\pi/2 + k\pi$ aligns the object's z-axis with the earth-fixed X-axis. Thus, any roll rotations are useless as they can be done by yaw rotations. This is can be shown in figure below.

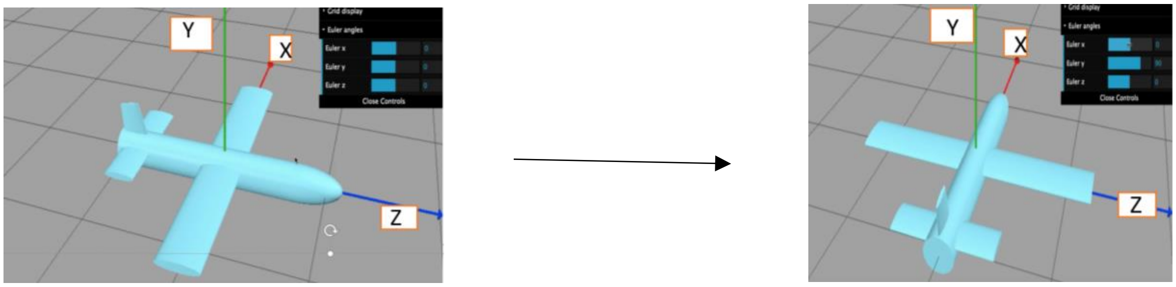


Figure 10: The object’s X-axis becoming aligned with Earth-fixed Z-axis (“Euler Angles…”, 2015)

The problem of gimbal lock becomes observable during the interpolation of orientations. In interpolation, the Tait-Bryan Euler Angles maintain that the order of rotation is still yaw, pitch, roll (Jia, 2016). In the diagram above where $\Psi = 0$, $\Theta = \pi/2$, $\Theta = 0$, if yaw rotation was applied, the body frame would be disregarded and the new rotation matrix including yaw and pitch would be reapplied to the initial frame.

The diagram below demonstrates the problem of gimbal lock during interpolation. When adjusting the yaw and the roll angles while $\Theta = \pi/2$, they both produce the same rotations as shown by the blue arrow.

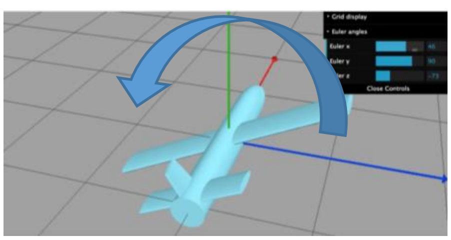


Figure 11: Gimbal Lock (“Euler Angles…”, 2015)

The problem of gimbal lock is computationally expensive because computers have to explicitly go to great lengths to prevent the issue. In robots, sensors are programmed to warm users when its orientation is nearing towards gimbal lock conditions. This single problem is the reason Tait-Bryan Euler Angles are substituted for rotational formulas that don’t have this issue such as quaternions.

Conclusion

Tait-Bryan Euler Angles are useful in their ability to rotate vectors. The formula's property of intrinsic rotation is definitely applicable in modern devices. Even more, its basis in trigonometry and matrix multiplication allows even rudimentary programmers to utilize the formula. Additionally, the disintegration of an abstract rotation into consecutive defined rotations about familiar axes enable public users to perform rotations on models using Euler angles without knowing the mathematical basis.

On the other hand, the problem of gimbal lock provides trouble for Tait-Bryan Euler Angles. This computationally expensive problem dissuades its use to perform rotations.

In observance of gimbal lock, Tait-Bryan Euler Angles should not be used to perform rotations; however, programs should display net rotations in Tait-Bryan Euler Angles due to its user-friendly, simplistic nature. This use of Tait-Bryan Euler Angles reaps the advantages of the formula without bearing its disadvantages.

Word Count: 2771

Works Cited:

A short derivation to basic rotation around the x-, y- or z-axis [PDF]. (2011, September).

Euler Angles - Interactive 3D Graphics [Video file]. (2015, February 15). Retrieved February 7, 2017, from https://www.youtube.com/watch?v=q0jgqeS_ACM&t=76s

Euler Angles. (n.d.). Retrieved February 04, 2017, from http://mathworld.wolfram.com/EulerAngles.html

Understanding Euler Angles. (n.d.). CH Robotics. Retrieved October 31, 2016, from http://www.chrobotics.com/library/understanding-euler-angles

Rowland, Todd and Weisstein, Eric W. "Linear Transformation." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/LinearTransformation.html

UM7 Orientation Sensor. (n.d.). Retrieved February 07, 2017, from https://www.pololu.com/product/2764

Jia, Y. (2016, September 6). Rotation in the Space∗ [PDF].