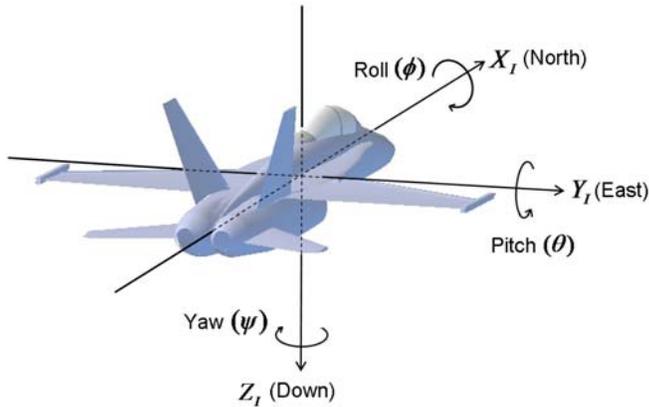


## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---

### 1. Introduction



Attitude and Heading Sensors from CH Robotics can provide orientation information using both Euler Angles and Quaternions. Compared to quaternions, Euler Angles are simple and intuitive and they lend themselves well to simple analysis and control. On the other hand, Euler Angles are limited by a phenomenon called "Gimbal Lock," which we will

investigate in more detail later. In applications where the sensor will never operate near pitch angles of  $\pm 90$  degrees, Euler Angles are a good choice.

Euler angles provide a way to represent the 3D orientation of an object using a combination of three rotations about different axes. For convenience, we use multiple coordinate frames to describe the orientation of the sensor, including the "inertial frame," the "vehicle-1 frame," the "vehicle-2 frame," and the "body frame." The inertial frame axes are Earth-fixed, and the body frame axes are aligned with the sensor. The vehicle-1 and vehicle-2 are intermediary frames used for convenience when illustrating the sequence of operations that take us from the inertial frame to the body frame of the sensor.

It may seem unnecessarily complicated to use four different coordinate frames to describe the orientation of the sensor, but the motivation for doing so will become clear as we proceed.

For clarity, this application note assumes that the sensor is mounted to an aircraft. All examples and figures are given showing the changing orientation of the aircraft.

**Table of Contents**

**1. INTRODUCTION ..... 1**

**2. THE INERTIAL FRAME..... 3**

**3. THE VEHICLE-1 FRAME ..... 4**

**4. THE VEHICLE-2 FRAME ..... 5**

**5. THE BODY FRAME..... 6**

**6. GIMBAL LOCK..... 7**

**7. USING THE EULER ANGLE OUTPUTS OF THE SENSOR..... 9**

**7.1. Converting Body-Frame Data to the Inertial-Frame..... 9**

**DISCLAIMER..... 12**

**List of Figures**

Figure 1 - Inertial Frame ..... 3

Figure 2 - The Vehicle-1 Frame..... 4

Figure 3 - The Vehicle-2 Frame..... 5

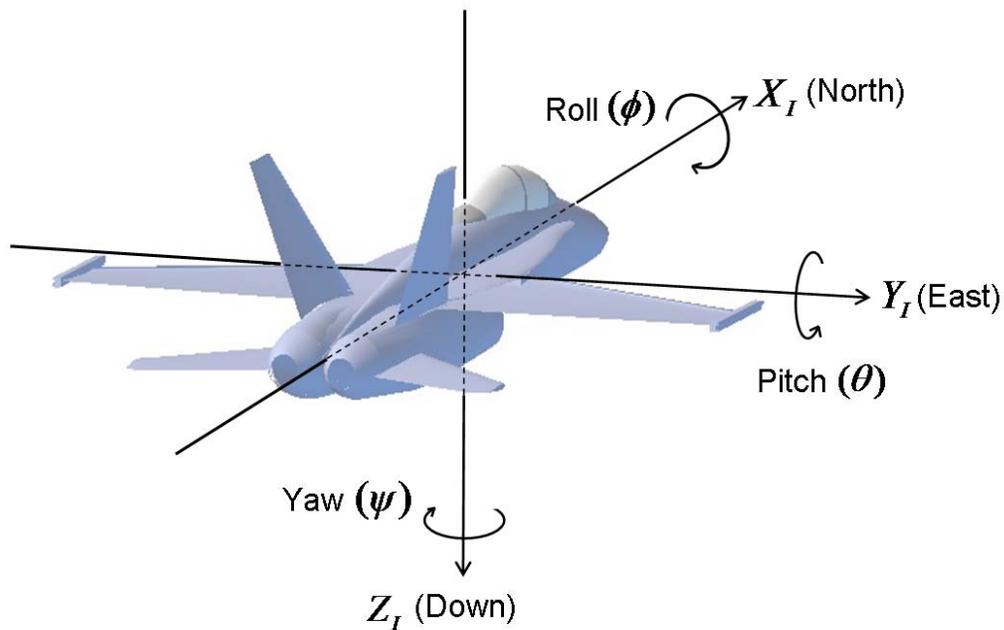
Figure 4 - The Body Frame..... 6

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

### 2. The Inertial Frame

The "inertial frame" is an Earth-fixed set of axes that is used as an unmoving reference. CH Robotics' sensors use a common aeronautical inertial frame where the x-axis points north, the y-axis points east, and the z-axis points down as shown below. We will call this a North-East-Down (NED) reference frame. Note that because the z-axis points down, altitude above ground is actually a negative quantity.

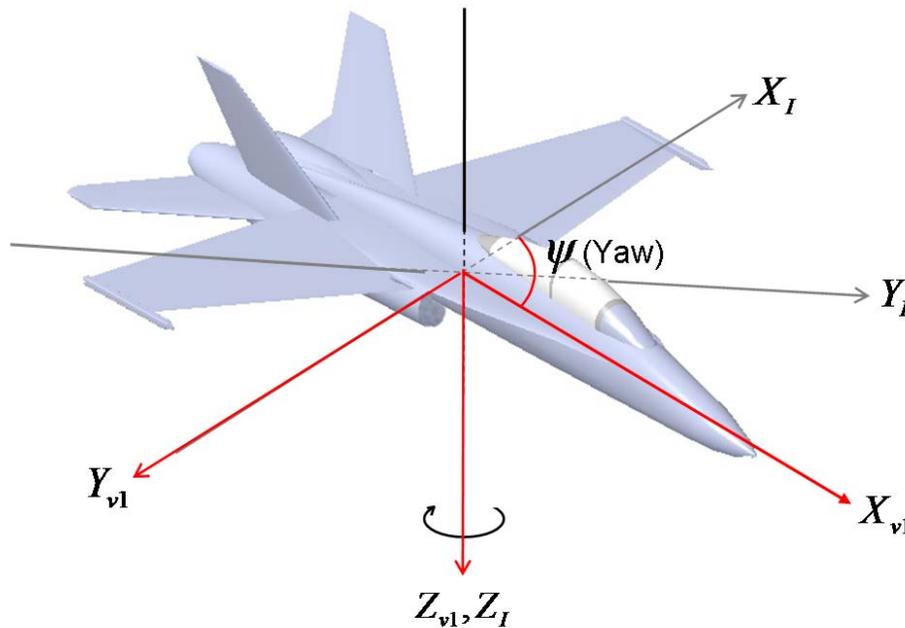


**Figure 1 - Inertial Frame**

The sequence of rotations used to represent a given orientation is first yaw, then pitch, and finally roll.

### 3. The Vehicle-1 Frame

As shown in Figure 1, yaw represents rotation about the inertial-frame z-axis by an angle  $\psi$ . The yaw rotation produces a new coordinate frame where the z-axis is aligned with the inertial frame and the x and y axes are rotated by the yaw angle  $\psi$ . We call this new coordinate frame the vehicle-1 frame. The orientation of the vehicle-1 frame after yaw rotation is show in Figure 2. The vehicle-1 frame axes are colored red, while the inertial frame axes are gray.



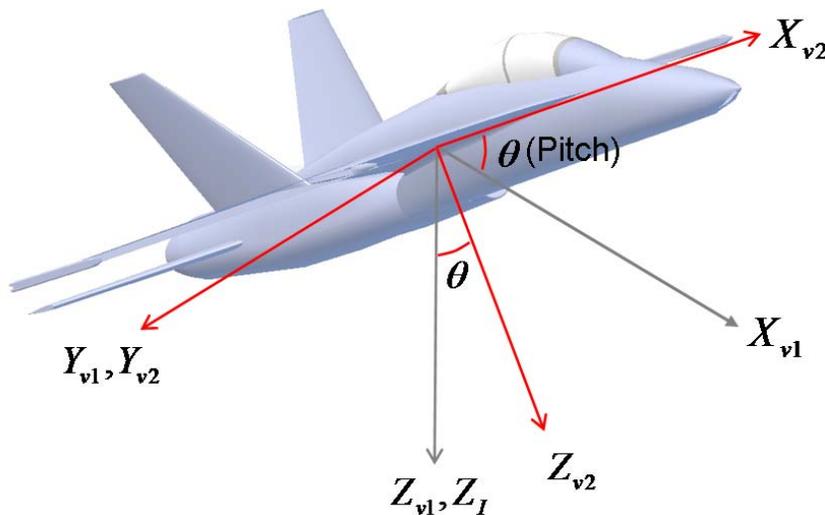
**Figure 2 - The Vehicle-1 Frame.**

Rotation of a vector from the Inertial Frame to the Vehicle-1 Frame can be performed by multiplying the vector by the rotation matrix

$$R_I^{v1}(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

## 4. The Vehicle-2 Frame

Pitch represents rotation about the vehicle-1 Y-axis by an angle  $\theta$  as shown in Figure 3. For clarity, the inertial-frame axes are not shown. The vehicle-1 frame axes are shown in gray, and the vehicle-2 axes are shown in red. It is important to note that pitch is NOT rotation about the inertial-frame Y-axis.



**Figure 3 - The Vehicle-2 Frame**

The rotation matrix for moving from the vehicle-1 frame to the vehicle-2 frame is given by

$$R_{v1}^{v2}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}.$$

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

The rotation matrix for moving from the inertial frame to the vehicle-2 frame consists simply of the yaw matrix multiplied by the pitch matrix:

$$R_I^{v2}(\theta, \psi) = R_{v1}^{v2}(\theta)R_I^{v1}(\psi).$$

### 5. The Body Frame

The body frame is the coordinate system that is aligned with the body of the sensor. On an aircraft, the body frame x-axis typically points out the nose, the y-axis points out the right side of the fuselage, and the z-axis points out the bottom of the fuselage.

The body frame is obtained by performing a rotation by the angle  $\phi$  around the vehicle-2 frame x-axis as shown in Figure 4. For clarity, the body frame and vehicle-1 frame axes are not shown. The vehicle-2 frame axes are shown in gray, while the body-frame axes are shown in red.

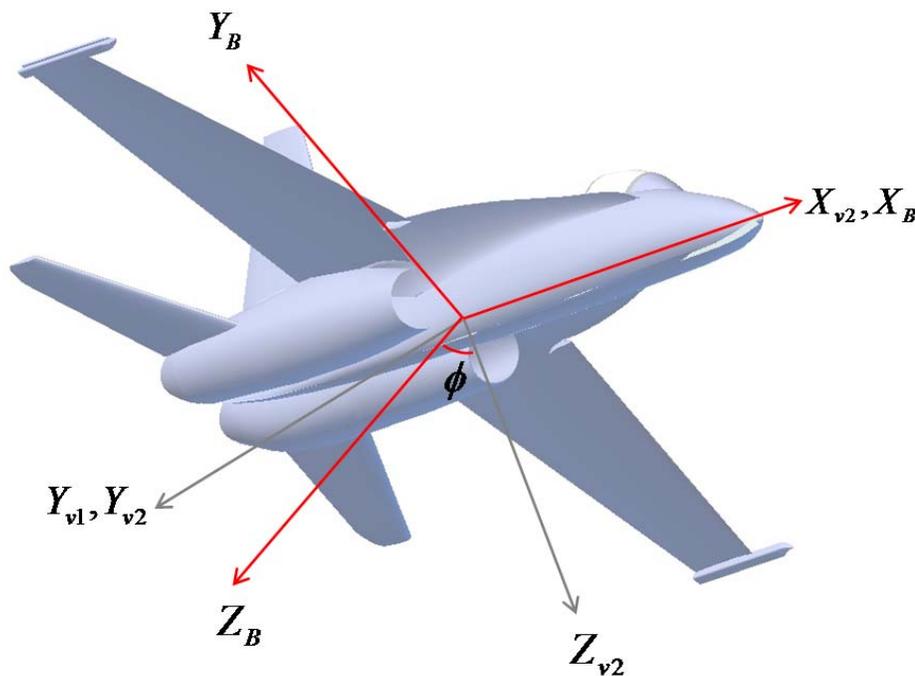


Figure 4 - The Body Frame

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---

The rotation matrix for moving from the vehicle-2 frame to the body frame is given by

$$R_{v_2}^B(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}.$$

The complete rotation matrix for moving from the inertial frame to the body frame is given by

$$R_I^B(\phi, \theta, \psi) = R_{v_2}^B(\phi)R_{v_1}^{v_2}(\theta)R_I^{v_1}(\psi).$$

The rotation matrix for moving the opposite direction - from the body frame to the inertial frame - is given by

$$R_B^I(\phi, \theta, \psi) = R_I^{v_1}(-\psi)R_{v_1}^{v_2}(-\theta)R_{v_2}^B(-\phi).$$

Note that all this does is reverse the order of operations and reverse the direction of rotation.

## 6. Gimbal Lock

Gimbal lock occurs when the orientation of the sensor cannot be uniquely represented using Euler Angles. The exact orientation at which gimbal lock occurs depends on the order of rotations used. On CH Robotics' sensors, the order of operations results in gimbal lock when the pitch angle is 90 degrees.

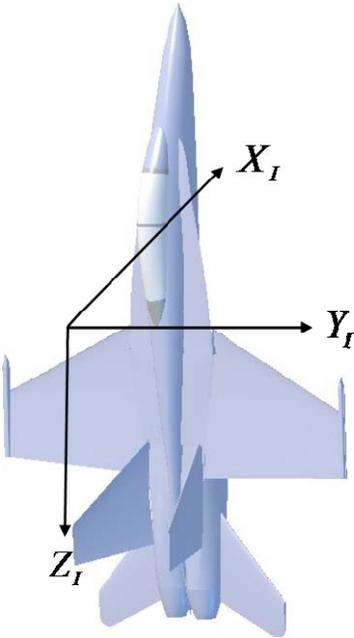
Intuitively, the cause of gimbal lock is that when the pitch angle is 90 degrees, yaw and roll cause the sensor to move in exactly the same fashion. Consider Figure 5 for an illustration of the gimbal lock condition. By following the sequence of rotations discussed in this paper, it should be easy to see that the orientation in Figure 5 can be obtained by yawing and then pitching, OR by pitching and then rolling.

More details about the mathematical reasons for gimbal lock are given in Section 7.1

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---



An orientation sensor or AHRS that uses Euler Angles will always fail to produce reliable estimates when the pitch angle approaches 90 degrees. This is a fundamental problem of Euler Angles and can only be solved by switching to a different representation method. All CH Robotics attitude sensors use quaternions so that the output is always valid even when Euler Angles are not.

For details about quaternions, please refer to Application Note AN-1006 - Understanding Quaternions.

## 7. Using the Euler Angle Outputs of the Sensor

The methodology needed to use the Euler Angle outputs of CH Robotics sensors depends on the specific application. A number of common applications is described in this section.

### 7.1. Converting Body-Frame Data to the Inertial-Frame

The rate gyros, accelerometers, and magnetometers on CH Robotics orientation sensors are aligned with the body frame of the sensor, so that if inertial frame data is needed, the sensor outputs must be converted from the body frame to the inertial frame. This can be accomplished by performing a simple matrix multiplication using the matrices described in 5.

For example, suppose that we want to obtain inertial frame accelerometer data so that we can integrate acceleration to obtain velocity estimates in the north, east, and down directions. Let  $\mathbf{v}_B$  be the measured body-frame acceleration vector reported by the sensor. Then the inertial frame acceleration is given by

$$\mathbf{v}_I = R_B^I(\phi, \theta, \psi) * \mathbf{v}_B .$$

The vector  $\mathbf{v}_I$  gives us the measured acceleration with respect to the inertial frame. Note that this gives us the *measured* inertial-frame acceleration, not the actual acceleration. A little more work is required before we can extract the physical acceleration of the sensor, and even then, the obtainable velocity accuracy using low-cost sensors is extremely poor. For more details, see Application Note AN-1007 - Estimating Velocity and Position with Accelerometers.

Magnetometer data can also be converted to the inertial frame in exactly the same fashion as the accelerometers if desired.

Converting rate gyro data to the inertial frame is a little more complicated. Like the accelerometer and magnetometer data, the rate gyro data is reported with respect to the body frame of the sensor. This means that the derivative of your Euler Angles is NOT what is being reported by the

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---

rate gyros. If you want angular rates with respect to the inertial frame, the rate gyro data must be converted to the proper coordinate frames. This is a little more complicated than it was for the accelerometers and magnetic sensors because each gyro angular rate must be converted to a different coordinate frame. Recall that yaw represents rotation about the inertial frame z-axis, pitch represents rotation about the vehicle-1 frame y-axis, and roll represents rotation about the vehicle-2 frame x-axis. Then to get the angular rates in the proper frames, the z-axis gyro output must be rotated into the inertial frame, the y-axis gyro output must be rotated into the vehicle-1 frame, and the x-axis gyro output must be rotated into the vehicle-2 frame.

The resulting transformation matrix for converting body-frame angular rates to Euler angular rates is given by

$$D(\phi, \theta, \psi) = \begin{bmatrix} 1 & \cos(\phi) \tan(\theta) & \sin(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) / \cos(\theta) & \cos(\phi) / \cos(\theta) \end{bmatrix}.$$

Let  $p$  represent the body-frame x-axis gyro output,  $q$  represent the body-frame y-axis gyro output, and  $r$  represent the body-frame z-axis output. Then the Euler Angle rate data is computed as

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = D(\phi, \theta, \psi) * \begin{pmatrix} p \\ q \\ r \end{pmatrix}.$$

This operation illustrates mathematically why gimbal lock becomes a problem when using Euler Angles. To estimate yaw, pitch, and roll rates, gyro data must be converted to their proper coordinate frames using the matrix  $D$ . But notice that there is a division by  $\cos(\theta)$  in two places on the last row of the matrix. When the pitch angle approaches +/- 90 degrees, the denominator goes to zero and the matrix elements diverge to infinity, causing the filter to fail.

The conversion from body-frame gyro data to Euler Angle rates happens internally on all CH Robotics sensors, but the converted data is not made available on all CH Robotics products. Refer

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---

to specific device datasheets for details on what data is available. For devices where Euler Angle rates are not reported, the body-frame angular rate data can be converted as described above.

## AN-1005 Understanding Euler Angles

Document rev. 1.0, updated 10/19/2012

---

### Disclaimer

This document is provided as reference only; "typical" device specifications must be evaluated by the end-user. CH Robotics reserves the right to modify this document and the products it describes without notice.

**CH Robotics products are not intended for use in weapons systems, aircraft, life-saving or life-sustaining systems, automobiles, or any other application where failure could result in injury, death, property damage, or environmental damage.**