Understanding Quaternions

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 prevents them from measuring orientation when the pitch angle approaches +/- 90 degrees.

Quaternions provide an alternative measurement technique that does not suffer from gimbal lock. Quaternions are less intuitive than Euler Angles and the math can be a little more complicated. This application note covers the basic mathematical concepts needed to understand and use the quaternion outputs of CH Robotics orientation sensors.

Sensors from CH Robotics that can provide quaternion orientation outputs include the UM6 Orientation Sensor and the UM6-LT Orientation Sensor.

2. Quaternion Basics

A quaternion is a four-element vector that can be used to encode any rotation in a 3D coordinate system. Technically, a quaternion is composed of one real element and three complex elements, and it can be used for much more than rotations. In this application note we'll be ignoring the theoretical details about quaternions and providing only the information that is needed to use them for representing the attitude of an orientation sensor.

The attitude quaternion estimated by CH Robotics orientation sensors encodes rotation from the "inertial frame" to the sensor "body frame." The inertial frame is an Earth-fixed coordinate frame defined so that the x-axis points north, the y-axis points east, and the z-axis points down as shown in Figure 1. The sensor body-frame is a coordinate frame that remains aligned with the sensor at all times. Unlike Euler Angle estimation, only the body frame and the inertial frame are needed when quaternions are used for estimation (Understanding Euler Angles provides more details about using Euler Angles for attitude estimation).

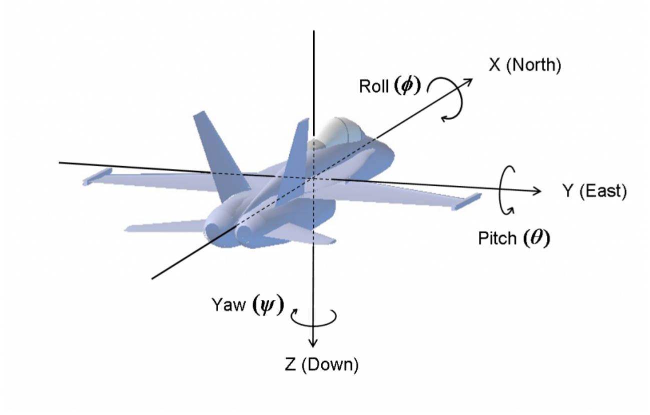

Figure 1 - The Inertial Frame

Let the vector $\mathbf{q}_i^b$ be defined as the unit-vector quaternion encoding rotation from the inertial frame to the body frame of the sensor:

$$\mathbf{q}_i^b = \begin{pmatrix} a & b & c & d \end{pmatrix}^T.$$

where T is the vector transpose operator. The elements *b, c,* and *d* are the "vector part" of the quaternion, and can be thought of as a vector about which rotation should be performed. The element *a* is the "scalar part" that specifies the amount of rotation that should be performed about the vector part. Specifically, if θ is the angle of rotation and the vector $\begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$ is a unit vector representing the axis of rotation, then the quaternion elements are defined as

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \cos(0.5\theta) \\ v_x \sin(0.5\theta) \\ v_y \sin(0.5\theta) \\ v_z \sin(0.5\theta) \end{pmatrix}.$$

In practice, this definition needn't be used explicitly, but it is included here because it provides an intuitive description of what the quaternion represents. CH Robotics sensors output the quaternion $\mathbf{q}_i^b$ when quaternions are used for attitude estimation.

3. Rotating Vectors Using Quaternions

The attitude quaternion $\mathbf{q}_i^b$ can be used to rotate an arbitrary 3-element vector from the inertial frame to the body frame using the operation

$$\mathbf{v}_B = \mathbf{q}_i^b \begin{pmatrix} 0 \\ \mathbf{v}_I \end{pmatrix} (\mathbf{q}_i^b)^{-1}.$$

That is, a vector can rotated by treating it like a quaternion with zero real-part and multiplying it by the attitude quaternion and its inverse. The inverse of a quaternion is equivalent to its conjugate, which means that all the vector elements (the last three elements in the vector) are negated. The rotation also uses quaternion multiplication, which has its own definition.

Define quaternions $\mathbf{q}_1 = \begin{pmatrix} a_1 & b_1 & c_1 & d_1 \end{pmatrix}^T$ and $\mathbf{q}_2 = \begin{pmatrix} a_2 & b_2 & c_2 & d_2 \end{pmatrix}^T$. Then the quaternion product $\mathbf{q}_1 \mathbf{q}_2$ is given by

$$\mathbf{q}_1 \mathbf{q}_2 = \begin{pmatrix} a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 \\ a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2 \\ a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2 \\ a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2 \end{pmatrix}.$$

To rotate a vector from the body frame to the inertial frame, two quaternion multiplies as defined above are required. Alternatively, the attitude quaternion can be used to construct a 3x3 rotation matrix to perform the rotation in a single matrix multiply operation. The rotation matrix from the inertial frame to the body frame using quaternion elements is defined as

$$R_i^b(\mathbf{q}_i^b) = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix}.$$

Then the rotation from the inertial frame to the body frame can be performed using the matrix multiplication

$$\mathbf{v}_B = R_i^b(\mathbf{q}_i^b)\mathbf{v}_I.$$

Regardless of whether quaternion multiplication or matrix multiplication is used to perform the rotation, the rotation can be reversed by simply inverting the attitude quaternion before performing the rotation. By negating the vector part of the quaternion vector, the operation is reversed.

4. Converting Quaternions to Euler Angles

CH Robotics sensors automatically convert the quaternion attitude estimate to Euler Angles even when in quaternion estimation mode. This means that the convenience of Euler Angle estimation is made available even when more robust quaternion estimation is being used.

If the user doesn't want to have the sensor transmit both Euler Angle and Quaternion data (for example, to reduce communication bandwidth requirements), then the quaternion data can be converted to Euler Angles on the receiving end.

The exact equations for converting from quaternions to Euler Angles depends on the order of rotations. CH Robotics sensors move from the inertial frame to the body frame using first yaw, then pitch, and finally roll. This results in the following conversion equations:

$$\phi = \arctan\left(\frac{2(ab+cd)}{a^2-b^2-c^2+d^2}\right),$$

$\theta = -\arcsin(2(bd - ac))$, and

$$\psi = \arctan\left(\frac{2(ad+bc)}{a^2+b^2-c^2-d^2}\right).$$

See the chapter on Understanding Euler Angles for more details about the meaning and application of Euler Angles. When converting from quaternions to Euler Angles, the atan2 function should be used instead of atan so that the output range is correct. Note that when converting from quaternions to Euler Angles, the gimbal lock problem still manifests itself. The difference is that since the estimator is not using Euler Angles, it will continue running without problems even though the Euler Angle output is temporarily unavailable. When the estimator runs on Euler Angles instead of quaternions, gimbal lock can cause the filter to fail entirely if special precautions aren't taken.