Our Friend the Quaternion

S. Ziskind

Quaternions are commonly used to represent rotations in three dimensions. One classical engineering use is in strapdown inertial navigation systems, as well as in applications requiring good attitude knowledge (e.g. robotics).

References

[1] F. Klein, "Arithmetic, Algebra, Analysis", Dover, 1925

[2] P. G. Savage, "Strapdown System Algorithms", AGARD Lecture Series No. 95

[3] L. Susanka, http://susanka.org/Notes/quaternions.pdf

1 Strapdown Inertial Navigation

Prior to the introduction of GPS in 1980, precision navigation was done using inertial sensors, namely accelerometers and gyroscopes. A triad of accelerometers provided a 3D acceleration vector, comprised of the combined effects of linear motion and gravity. Subtracting gravity, the acceleration could then be integrated twice to obtain velocity and position. One impediment to doing this was the imperfection of sensors, a large subject that won't be discussed here. Even with perfect sensors, however, there are two problems with mechanizing the integration scheme. First, a very precise knowledge of gravity is needed. Second, the orientation of the accelerometers is needed, so that the accelerometer integration can be performed in a single, consistent coordinate frame.

Two dominant ways to overcome these problems were used. One way was to isolate the motion of the sensor suite from changing vehicle attitude. This entailed using a nested set of very low friction gimbals to (mostly) decouple the sensor platform from vehicle attitude change. In addition, the platform was 'torqued' as the vehicle changed location, so as to keep it locally level. This worked, but required gimbal hardware and torquing motors, adding weight, cost, volume, complexity, and reliability issues.

A second method, only practical with the ability to perform high speed computation, fastened the sensors directly to the vehicle ('strapped down'), and used gyroscopes to establish a calculated/virtual local level frame. Several ways of keeping track of the resulting attitude are used: Euler Angles, direction cosine matrices, and quaternions.

2 The Objects and their representations

A rotation of the plane $\mathbb{R}^2$ may be specified by the angle through which the rotation occurs, but it may also be represented by a complex number of unit magnitude in accordance with the association: $\theta \to e^{i\theta}$. Furthermore, the special multiplication of complex numbers allows one to represent the linear operation of rotation by multiplication. One may hope for a similarly direct representation of 3 dimensional rotations, but the more complicated nature of these rotations forces a more involved representation. The representation presented here, and used in navigation modeling, is the quaternion. The exposition is based on accounts in [1] and [2]. Reference [3] is a more complete discussion, from a mathematically sophisticated perspective.

The basic objects used to represent rotations of $\mathbb{R}^3$ live in $\mathbb{R}^4$ rather than $\mathbb{R}^3$. By analogy to the case of complex numbers, every point in $\mathbb{R}^4$ may be written as a*1+b*i+c*j+d*k. Thus the unit vectors 1, i, j, and k are taken to be the orthonormal basis of $\mathbb{R}^4$. The **quaternions** are the points of $\mathbb{R}^4$, but with the following special multiplication attached. For any quaternion $Q = (a, b, c, d)$ let a be called the <u>scalar part</u>, and let (b, c, d) be called the <u>vector part</u>. The length of a quaternion is, as expected in $\mathbb{R}^4$, $||Q|| = \sqrt{a^2 + b^2 + c^2 + d^2}$. Again by analogy to complex numbers, one defines multiplications by insisting that the distributive law holds, and that:

$$1 = multiplicative\, identity$$

$$i * i = j * j = k * k = -1$$

$$
\begin{array}{ccc}
i * j = +k & j * k = +i & k * i = +j \\
j * i = -k & k * j = -i & i * k = -j
\end{array}
$$

The special multiplication of quaternions is thus a blend of 3D dot and cross products on the vector parts. Clearly quaternion multiplication is not commutative, but that should not be surprising. After all, we aim to mechanize 3D rotations, and such rotations are order dependent.

It is straightforward to verify that the product of two vectors has length equal to the product of their lengths. In particular, multiplication by a vector of unit length is an operation which does not change the length of the argument vector. Consequently, multiplication by a unit vector is a rotation, followed possibly by a reflection. Also by analogy to the complex case, the conjugate of a vector $Q = (a, b, c, d)$ is defined as $Q' = (a, -b, -c, -d)$. It is direct to find that $||Q|| = \sqrt{Q * Q'}$.

It is important to note that the action of quaternion multiplication, in which a quaternion acts on vectors in $\mathbb{R}^4$, is linear, and a straightforward calculation using the special rules of quaternion multiplication, that:

$$
\begin{pmatrix} s \\ x \\ y \\ z \end{pmatrix} * \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} s & -x & -y & -z \\ x & s & -z & y \\ y & z & s & -x \\ z & -y & x & s \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}
$$

so that the 4x4 matrix represents quaternion multiplication. Also, separating the scalar and vector parts of two quaternions, we can write:

$$
(a, U) * (b, V) = (ab - U \cdot V, aV + bU + U \times V)
$$

3 Rotations

An old theorem of Cayley (see [1],[3]) says that the most general rotation of $\mathbb{R}^4$ is obtained by the operation of pre- and post-multiplying by unit quaternions. Specifically, if T is any rotation, then there exist unit quaternions P and R such that for all X in $\mathbb{R}^4$:

$$T[X] = P * X * R$$

To obtain the corresponding representation for rotations of $\mathbb{R}^3$, we embed $\mathbb{R}^3$ into $\mathbb{R}^4$ by the association: $(x, y, z) \to (0, x, y, z)$. It is now claimed that if Q is any quaternion and V is any pure vector quaternion, then $Q * V * Q'$ is also a pure vector quaternion. This fact, which is a simple computation, suggests that every rotation of $\mathbb{R}^3$ can be represented by embedding the vector in $\mathbb{R}^4$ and then pre- and post-multiplying by a unit quaternion and its conjugate.

In order to find the representation of a rotation of $\mathbb{R}^3$, first note that such a rotation is completely specified by the axis of rotation and the size of the rotation. So for a general rotation of $\mathbb{R}^3$ let us attempt to identify the axis and angle. First, let the general unit quaternion be $Q = (a, bU)$, where U is a unit vector and $a^2 + b^2 = 1$. The rotation in question is thus

$$(a, bU) * (0, V) * (a, -bU) = a^2\, V + 2ab\, U \times V + b^2\, (U \cdot V)U - b^2\, (U \times V) \times U$$

In the special case where V = U the result simplifies to $a^2\, V + b^2\, U = V$, so that U is indeed the axis fixed by this rotation. If one chooses, on the other hand, to have V be a unit vector perpendicular to U, then $(U \times V) \times U = V$, and the result is that

$$(a^2 - b^2)\, V + 2ab\, U \times V = cos(\theta)V + sin(\theta)U \times V$$

where θ is the angle of the rotation. Recalling the double angle formula, it follows at once that

$$\begin{aligned} a &= cos(\theta/2) \qquad and \\ b &= sin(\theta/2). \end{aligned}$$

To summarize, suppose that the vector $V = (v_1, v_2, v_3)$ needs to be rotated about the unit vector $U = (u_1, u_2, u_3)$ by the angle θ. Form the quaternion

$$\begin{aligned} Q &= (cos(\theta/2), sin(\theta/2)U) \\ &= (cos(\theta/2), u_1 sin(\theta/2), u_2 sin(\theta/2), u_3 sin(\theta/2)), \end{aligned}$$

and let let Q' be its quaternion conjugate, i.e. $Q' = (cos(\theta/2), -sin(\theta/2)U)$. Next embed V in $\mathbb{R}^4$ as $W = (0, V)$. Form the product $Q * W * Q'$ using the rules of quaternion multiplication. The scalar part of this product will be zero, and the vector part is the position of V after the rotation.

4 Quaternion Differentiation

When an object undergoes rotational motion in space, its orientation at any time is equivalent to some single rotation from its initial orientation, even if the history of the motion consists of a mixture of rotations about various axes. When the angular rate of the object is known as a function of time, it is possible to write the differential equation satisfied by the quaternion representing the cumulative rotation.

To begin with, all vectors will be expressed in the coordinate frame defined at time=0., which will be taken to be an inertial frame. Define

Q_k = the cumulative rotation quaternion at time t=k*dt, and

$Q_{k+1,k}$ = the rotation quaternion which carries the object from its orientation at time k*dt to its orientation at time (k+1)*dt

Now one has $Q_{n+1} = Q_{n+1,n} * Q_n$. Consequently,

$$\dot{Q_n} \doteq (Q_{n+1} - Q_n)/dt = ((Q_{n+1,n} - I)/dt) * Q_n$$

For very small time increments dt, the quaternion $Q_{n+1,n}$ corresponds roughly to a rotation of a given small angle about a fixed axis. If this angle is θ and the fixed axis is U, then we can use small angle approximations to write

$$Q_{n+1,n} = (cos\frac{\theta}{2}, sin\frac{\theta}{2}U_n) \approx (1, \frac{\theta}{2}(U_x, U_y, U_z)) = (1, \frac{\theta_x}{2}, \frac{\theta_y}{2}, \frac{\theta_z}{2})$$

$$\approx \begin{pmatrix} 1 & -\frac{\theta_x}{2} & -\frac{\theta_y}{2} & -\frac{\theta_z}{2} \\ \frac{\theta_x}{2} & 1 & -\frac{\theta_z}{2} & \frac{\theta_y}{2} \\ \frac{\theta_y}{2} & \frac{\theta_z}{2} & 1 & -\frac{\theta_x}{2} \\ \frac{\theta_z}{2} & -\frac{\theta_y}{2} & \frac{\theta_x}{2} & 1 \end{pmatrix}$$

Using the matrix representation for the operator $(Q_{n+1,n} - I)/dt$, the differential equation for Q becomes

$$\dot{\mathbf{Q}} = \frac{1}{2} \begin{pmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & -w_z & w_y \\ w_y & w_z & 0 & -w_x \\ w_z & -w_y & w_x & 0 \end{pmatrix} \mathbf{Q}$$

where w_x, w_y, and w_z are the components of the angular rate vector.

This last equation is very useful for several reasons. First, it is linear, which gives it very stable and straightforward numerical integration properties. There are no quotients, roots, or trig functions. Second, the matrix entries are those that are supplied by inertial rate sensors.