

MIT 16.S897

Lecture 3

QUATERNIONS, PART 2 & RIGID BODIES

Lecture by Zachary Manchester
Typesetting by Joseph Hobbs

February 10, 2026

Contents

1	Quaternions, Part 2	1
1.1	Quaternion Product	2
1.2	Identity Quaternion	3
1.3	Quaternion Inverse	3
1.4	Double Cover of SO(3)	3
1.5	Quaternions as Rotations	4
1.6	Quaternions and Rotation Matrices	4
1.7	Quaternion Kinematics	4
1.8	Exponential Map	5
2	Rigid-Body Dynamics	6
2.1	Conserved Quantities	6
2.2	Moment of Inertia	7
2.3	Composing Moments of Inertia	7

1 Quaternions, Part 2

Extending the “planar quaternion” from the previous lecture, we introduce the **quaternion**

$$q = \begin{bmatrix} \cos(\theta/2) \\ r \sin(\theta/2) \end{bmatrix} \quad (1)$$

where r is a unit vector representing the axis of rotation, and θ is the angle of rotation about that axis. In a similar way that complex numbers are written as the sum of *real* and *imaginary* parts, quaternions are the sum of the **scalar part** and the **vector part**. In the above notation, the first row is the scalar part and the second row is the vector part. In

this course, the scalar part will always be written first, though other conventions call for the opposite order. For brevity, we will write

$$q = \begin{bmatrix} s \\ v \end{bmatrix}$$

where s is the scalar part and v is the vector part. The corresponding axis-angle vector to this quaternion is clearly

$$\phi = \theta r.$$

It is important to note that every rotation is described by a **unit quaternion**.

$$q^T q = \cos^2\left(\frac{\theta}{2}\right) + r^T r \sin^2\left(\frac{\theta}{2}\right) = 1.$$

Given any quaternion, it is extremely easy to find the “closest” unit quaternion; simply determine the norm of the quaternion according to

$$\|q\|^2 = q^T q \quad (2)$$

and divide every component by that value. This is far simpler than finding the “closest” orthogonal matrix to a given square matrix, which is typically done by singular value decomposition (SVD).

1.1 Quaternion Product

The definition of a Lie group requires that its elements are closed under a multiplication rule. If we wish to use unit quaternions to parameterize $\text{SO}(3)$, we must define the **quaternion product** like so.

$$q_1 \star q_2 = \begin{bmatrix} s_1 \\ v_1 \end{bmatrix} \star \begin{bmatrix} s_2 \\ v_2 \end{bmatrix} := \begin{bmatrix} s_1 s_2 - v_1^T v_2 \\ s_1 v_2 + s_2 v_1 + v_1 \times v_2 \end{bmatrix} \quad (3)$$

Notice that, due to the vector cross product, the quaternion product is **non-commutative** in general ($q_1 \star q_2 \neq q_2 \star q_1$). It is also important to note that the quaternion product is linear in each q_1 and q_2 , meaning we can write it as a matrix multiplication, recalling the *hat map* for 3D vectors.

$$q_1 \star q_2 = \begin{bmatrix} s_1 & -v_1^T \\ v_1 & s_1 I + \hat{v}_1 \end{bmatrix} \begin{bmatrix} s_2 \\ v_2 \end{bmatrix} = L(q_1) q_2$$

We introduce the notation $L(q_1)$ for this matrix, because left-multiplication by $L(q_1)$ is equivalent to quaternion *left*-multiplication by q_1 . We can also write a similar matrix, $R(q_2)$, like so.

$$q_1 \star q_2 = \begin{bmatrix} s_2 & -v_2^T \\ v_2 & s_2 I - \hat{v}_2 \end{bmatrix} \begin{bmatrix} s_1 \\ v_1 \end{bmatrix} = R(q_2) q_1$$

We call this matrix $R(q_2)$ because left-multiplication by $R(q_2)$ is equivalent to quaternion *right*-multiplication by q_2 . Finally, notice that, because quaternion inverse involves only flipping the sign on the vector part, the matrices L and R obey the property

$$L(q^\dagger) = L(q)^T \quad \text{and} \quad R(q^\dagger) = R(q)^T.$$

1.2 Identity Quaternion

The **identity quaternion** q_0 has the property that

$$q_0 \star q = q \star q_0 = q$$

for any quaternion q . It is not hard to see that the identity quaternion is uniquely given by a unit scalar part and zero vector part.

$$q_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4}$$

1.3 Quaternion Inverse

The **quaternion inverse** q^\dagger is defined for every quaternion q . For a unit quaternion q , its inverse is defined such that q^\dagger describes the inverse rotation as q . In other words, q^\dagger corresponds to rotation about the same axis as q , but by the negative angle. For a unit quaternion,

$$q^\dagger = \begin{bmatrix} \cos(-\theta/2) \\ r \sin(-\theta/2) \end{bmatrix}.$$

Using the properties of the cosine and sine functions,

$$q^\dagger = \begin{bmatrix} \cos(\theta/2) \\ -r \sin(\theta/2) \end{bmatrix}. \tag{5}$$

In other words, the quaternion inverse of a unit quaternion q is easily computed by negating the vector part of q .

1.4 Double Cover of SO(3)

In the same way as last lecture, it is not hard to show that

$$q(\theta + 2\pi) = -q \tag{6}$$

and therefore every quaternion and its negation correspond to the same rotation. Therefore, the set of unit quaternions are a **double cover** of $\text{SO}(3)$. It is an unfortunate result of group theory that we cannot write quaternions differently such that they do not doubly cover $\text{SO}(3)$.

1.5 Quaternions as Rotations

To rotate a vector Bx using a quaternion q , we must compute the product

$$\boxed{\begin{bmatrix} 0 \\ {}^Nx \end{bmatrix} = q \star \begin{bmatrix} 0 \\ {}^Bx \end{bmatrix} \star q^\dagger}. \quad (7)$$

Quaternion multiplication is associative, so we may compute these two products in either order. Notice that, by our definitions and properties of $L(q)$ and $R(q)$, we have

$$\begin{bmatrix} 0 \\ {}^Nx \end{bmatrix} = L(q)R(q^\dagger) \begin{bmatrix} 0 \\ {}^Bx \end{bmatrix} = L(q)R(q)^T \begin{bmatrix} 0 \\ {}^Bx \end{bmatrix}$$

and, because the quaternion product is associative, we can reverse the order of multiplications to get

$$\begin{bmatrix} 0 \\ {}^Nx \end{bmatrix} = R(q^\dagger)L(q) \begin{bmatrix} 0 \\ {}^Bx \end{bmatrix} = R(q)^T L(q) \begin{bmatrix} 0 \\ {}^Bx \end{bmatrix}.$$

To make our notation slightly cleaner, we will introduce the linear operator H , which converts a 3D vector into a quaternion with zero scalar part.

$$Hx = \begin{bmatrix} 0 \\ x \end{bmatrix} \Rightarrow H := \begin{bmatrix} 0 \\ I \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using H , we write

$$\boxed{{}^Nx = H^T L(q) R(q)^T H {}^Bx = H^T R(q)^T L(q) H {}^Bx}. \quad (8)$$

1.6 Quaternions and Rotation Matrices

It is interesting to note that quaternions q and rotation matrices Q have many similar properties.

$$Q^T Q = I \leftrightarrow q^T q = 1 \quad (9)$$

$$Q_3 = Q_2 Q_1 \leftrightarrow q_3 = q_2 \star q_1 \quad (10)$$

$$Q_1 = Q_2^T Q_3 \leftrightarrow q_1 = q_2^\dagger \star q_3 \quad (11)$$

$${}^N\hat{x} = Q^B \hat{x} Q^T \leftrightarrow H^N x = q \star H^B x \star q^\dagger \quad (12)$$

1.7 Quaternion Kinematics

Let us remember our example of integrating an angular velocity $\omega(t)$ to obtain a spacecraft attitude. Suppose that, over a very small timestep Δt , the quaternion q_1 rotates by Δq to become q_2 .

$$q_2 = q_1 \star \Delta q = q_1 \star \begin{bmatrix} \cos(\Delta\theta/2) \\ r \sin(\Delta\theta/2) \end{bmatrix}$$

Here, we take r to be the unit vector aligned with the axis of rotation (by the right-hand rule), and $\Delta\theta$ to be the small angle rotated through in Δt . Mathematically, $\omega(t) \cdot \Delta t = r \cdot \Delta\theta$. Taking the small-angle approximation of the cosine and sine functions,

$$q_2 = q_1 \star \Delta q \approx q_1 \star \begin{bmatrix} 1 \\ r \cdot \Delta\theta/2 \end{bmatrix}.$$

Evaluating the quaternion product and applying $\omega(t) \cdot \Delta t = r \cdot \Delta\theta$,

$$\begin{aligned} q_2 &\approx q_1 \star \left(q_0 + \frac{\Delta t}{2} H\omega(t) \right) \\ &= q_1 + \frac{\Delta t}{2} \cdot q_1 \star H\omega(t). \end{aligned}$$

Rearranging and taking the limit as $\Delta t \rightarrow 0$,

$$\lim_{\Delta t \rightarrow 0} \frac{q_2 - q_1}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{1}{2} L(q_1) H\omega(t).$$

From this, we conclude that

$$\boxed{\dot{q} = \frac{1}{2} L(q) H\omega(t)} \quad (13)$$

and we can integrate this numerically (for example, using Runge-Kutta) to obtain a quaternion result. It will be useful for us later to define the **attitude Jacobian** $G(q)$.

$$\boxed{G(q) := L(q)H} \Rightarrow \dot{q} = \frac{1}{2} G(q)\omega \quad (14)$$

1.8 Exponential Map

Because the attitude Jacobian $G(q)$ is linear in q , for constant angular velocity ω , we again have a linear first-order ODE for q . This suggests that we can again use the **exponential map** to obtain closed-form solutions for q . We define the **quaternion exponential map** $\exp_q(\cdot)$ such that

$$\exp_q(\phi) = \begin{bmatrix} \cos(\|\phi\|) \\ \phi / \|\phi\| \cdot \sin(\|\phi\|) \end{bmatrix}.$$

The closed-form solution to our linear ODE is therefore

$$\boxed{q(0) = q_0 \Rightarrow q(t) = \exp_q \left(\frac{1}{2} \omega t \right)}. \quad (15)$$

It is important to note that, at $\phi = 0$, the exponential map experiences a point singularity, because the vector part is divided by zero. For numerical stability, it is more preferable to define $\text{expq}(\cdot)$ equivalently as

$$\boxed{\text{expq}(\phi) = \begin{bmatrix} \cos(\|\phi\|) \\ \phi \cdot \text{sinc}(\|\phi\|) \end{bmatrix}}. \quad (16)$$

where $\text{sinc}(x) := \sin(x)/x$ when $x \neq 0$ and $\text{sinc}(0) := 1$. This definition is continuous and differentiable everywhere, ensuring numerical stability. Some computational tools define $\text{sinc}(x) := \sin(\pi x)/\pi x$. When implementing the exponential map, check corresponding documentation to determine which definition your chosen tool uses for $\text{sinc}(\cdot)$.

2 Rigid-Body Dynamics

The study of rigid-body dynamics focuses on studying the relative motion of bodies on which **Euclidean distances between pairs of points are preserved**. Practically, a body is considered “rigid” if its structural modal frequencies are much faster (order of magnitude or more) than both its rigid-body frequencies and its controller’s bandwidth. Because we consider each rigid body to have its own reference frame, we will typically speak of relative motion of corresponding reference frames, rather than the bodies themselves.

2.1 Conserved Quantities

Under torque-free conditions, **angular momentum** is conserved in all *inertial* reference frames.

$${}^N h = \text{constant} \quad (17)$$

This is *not true* in non-inertial (rotating) reference frames. However, it is true that the *magnitudes* of angular momentum vectors are preserved in rotating frames (assuming no torque). With no unbalanced forces or torques, **kinetic energy** is also preserved in the inertial frame. Imagine a rigid body is rotating about its center of mass, but its center of mass is not moving. We can write the kinetic energy of the rigid body T as the integral

$$T = \int \frac{1}{2} ({}^N \dot{x})^T ({}^N \dot{x}) dm. \quad (18)$$

Recalling our expression for ${}^N \dot{x}$, we can rewrite this as

$$T = \int \frac{1}{2} (Q \omega \times {}^B x)^T (Q \omega \times {}^B x) dm,$$

recalling that ω expresses the angular velocity vector in the *body* frame. Distributing the transpose operator and rearranging,

$$T = \frac{1}{2} \int (\omega \times {}^B x)^T Q^T Q (\omega \times {}^B x) dm.$$

We can cancel $Q^T Q$ because all rotation matrices are orthogonal. Using our previously defined “hat map”,

$$T = -\frac{1}{2} \int \omega^T \hat{x}^B \hat{x} \omega dm.$$

Because ω is independent of position on the body, we can factor it out from the integral.

$$T = \frac{1}{2} \omega^T \left[- \int {}^B \hat{x}^B \hat{x} dm \right] \omega = \frac{1}{2} \omega^T J \omega. \quad (19)$$

In the next section, we will discuss the properties of the integral J .

2.2 Moment of Inertia

The integral J is called the **moment of inertia** (or sometimes, the *inertia tensor*) and is a function only of rigid-body geometry (it is not dependent on attitude because it is computed in the body frame). It is also important to remember that ω here is the *body frame* angular velocity!

$$J := - \int {}^B \hat{x}^B \hat{x} dm = \int ({}^B x^T {}^B x I - {}^B x^B x^T) dm$$

It is important to note that the moment of inertia is always symmetric ($J = J^T$). Additionally, the moment of inertia is **positive semidefinite**, which means that it has non-negative eigenvalues and therefore the rotational kinetic energy defined above is always non-negative. The eigenvectors of J are always orthogonal, and its eigenvectors are considered the **principal axes** of the rigid body in question. Finally, the diagonal values of J obey the triangle inequality: $J_{ii} + J_{jj} \geq J_{kk}$.

2.3 Composing Moments of Inertia

When two rigid bodies are combined, their moments of inertia J_1, J_2 can be composed using the following process.

1. Find the center of mass of the combined body.
2. Sum the inertias J_1 and J_2 .
3. Add to the sum of inertias two terms considering each constituent body as a point mass, located at its own center of mass.

Formally, the inertia of a combination of rigid bodies is computed as

$$J_{1+2} = J_1 + J_2 + m_1 (r_1^T r_1 I - r_1 r_1^T) + m_2 (r_2^T r_2 I - r_2 r_2^T) \quad (20)$$

where m_1, m_2 are the respective masses of the two bodies and r_1, r_2 are the respective centers of mass of the two bodies, relative to their combined center of mass. This result is known as the **parallel axis theorem**.