# Introduction to Frames

Ziv Yaniv

School of Engineering and Computer Science
The Hebrew University, Jerusalem, Israel.

The subject of this lecture is *coordinate frame transformations* in short frames. The lecture is based on the following sources [1, 2, 3].

Frames are rigid transformations an abstract representation of a rotation and a translation.

$$^{A}_{B}F = [R, t]$$

The notation $^{A}_{B}F$ means that this frame describes the location and orientation of $B$ relative to $A$ where $A, B$ are cartesian coordinate systems as shown in Figure 1.

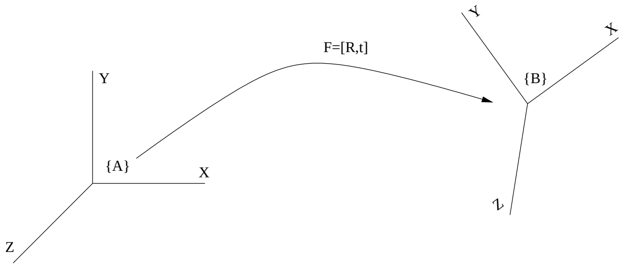


Figure 1: Transformation between two frames A and B.

The representation of translations is straightforward, 3D vectors. The representation of rotations is usually done in one of two ways, as a matrix or as a unit quaternion. The following two sections describe different aspects of these two representations. They are followed by two sections which describe small angle rotation matrices and rotation interpolation.

## 1 Rotation Matrices

If a matrix $R$ has the following properties it is a rotation matrix:

$$RR^T = I \quad (1)$$

$$det(R) = 1 \quad (2)$$

Property 2 is necessary to remove reflection matrices which comply with property 1 but have $det(R) = -1$.
What is immediately evident from properties 1 and 2 is that multiplying any vector by a rotation matrix does not change its norm.

$$||Rx|| = < Rx, Rx > = x^T R^T R x = x^T x = ||x||$$

We now take a look at the composition and inverse of frames described using rotation matrices.
Given $R_i$ a rotation matrix and $t_i$ a translation we have the following:

$$\begin{aligned} F_i &= [R_i, t_i] \\ F_i^{-1} &= [R_i^T, -R_i^T t_i] \\ F_i * F_j &= [R_i * R_j, R_i t_j + t_i] \end{aligned}$$

There are many ways to specify a rotation matrix. The following section describes several of them.

## 1.1 Rotation matrix specification

### 1.1.1 Fixed axis

Given frame $B$ we describe its orientation relative to frame $A$ as follows:
Start with a frame coincident with $A$. First rotate it about $X_A$ then rotate about $Y_A$ and finally rotate it about $Z_A$.

Let $\omega$ be a three vector $(\omega_x, \omega_y, \omega_z)$ representing the fixed axis rotation with respect to the x, y, z axes. This defines the following rotation matrix, where $c$ is cosine and $s$ sine:

$$\begin{aligned} {}^A_B R_{xyz} &= R_z(\omega_z) R_y(\omega_y) R_x(\omega_x) \\ &= \begin{bmatrix} c\omega_z & -s\omega_z & 0 \\ s\omega_z & c\omega_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\omega_y & 0 & s\omega_y \\ 0 & 1 & 0 \\ -s\omega_y & 0 & c\omega_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\omega_x & -s\omega_x \\ 0 & s\omega_x & c\omega_x \end{bmatrix} \quad (3) \\ &= \begin{bmatrix} c\omega_z c\omega_y & c\omega_z s\omega_y s\omega_x - s\omega_z c\omega_x & c\omega_z s\omega_y c\omega_x + s\omega_z s\omega_x \\ s\omega_z c\omega_y & s\omega_z s\omega_y s\omega_x + c\omega_z c\omega_x & s\omega_z s\omega_y c\omega_x - c\omega_z s\omega_x \\ -s\omega_y & c\omega_y s\omega_x & c\omega_y c\omega_x \end{bmatrix} \end{aligned}$$

Extracting $(\omega_x, \omega_y, \omega_z)$ from such a rotation matrix is according to the following formulas:

$$\begin{aligned} \omega_y &= atan2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \omega_z &= atan2(r_{21}/c\omega_y, r_{11}/c\omega_y) \\ \omega_x &= atan2(r_{32}/c\omega_y, r_{33}/c\omega_y) \end{aligned}$$

We chose the positive square root so that $-\pi \leq \omega_y \leq \pi$. Looking at the above equations one should ask, what happens if $\omega_y = \pm\pi$. We have an ambiguity in the interpretation of the rotation matrix. This situation is known as *gimbal lock* and is described in appendix A.

### 1.1.2 Z,Y,X Euler angles

According to Euler's rotation theorem, any rotation can be described using three angles. This section describes one of the options (Z,Y,X), for other specifications see [1].

Given frame $B$ we describe its orientation relative to frame $A$ as follows:
Start with a frame coincident with $A$. First rotate it about $Z_B$ then rotate about $Y_B$ and finally rotate it about $X_B$.

We build the rotation matrix incrementally according to the above description.

$$^A_BR = {}^A_{B'}R \; {}^{B'}_{B''}R \; {}^{B''}_{B}R$$

The result is the same as was described for fixed axis rotation and extracting the angles is the same too. Unfortuantly again we have gimbal lock (appendix A).

### 1.1.3 Axis Angle, Rodrigues' Formula

Given frame $B$ we describe its orientation relative to frame $A$ as follows:
Start with a frame coincident with $A$. Rotate the frame about the vector $^An$ by an angle $\theta$ according to the right hand rule.

Let $n = (n_x, n_y, n_z)$ be the rotation axis (unit vector), $\theta$ the rotation angle. The following matrix is the rotation we seek, where $c$ is cosine $s$ sine and $v\theta = 1 - cos\theta$:

$$R_n(\theta) = \begin{bmatrix} n_xn_xv\theta + c\theta & n_xn_yv\theta - n_zs\theta & n_xn_zv\theta + n_ys\theta \\ n_xn_yv\theta + n_zs\theta & n_yn_yv\theta + c\theta & n_yn_zv\theta - n_xs\theta \\ n_xn_zv\theta - n_ys\theta & n_yn_zv\theta + n_xs\theta & n_zn_zv\theta + c\theta \end{bmatrix} \tag{4}$$

<u>How did we get this matrix?</u>
We are looking for a matrix $R(n, \theta)$, a rotation in the plane defined by $n$ by $\theta$ radians. We will now show two constructive proofs, geometric and differential.

We start with the geometric proof. Let $p$ be a point we want to rotate and $a$ is the vector from the origin to $p$. Every vector $a$ can be decomposed into two vectors, parallel and perpendicular to $n$. The rotation only affects the perpendicular part and leaves the parallel component unchanged.

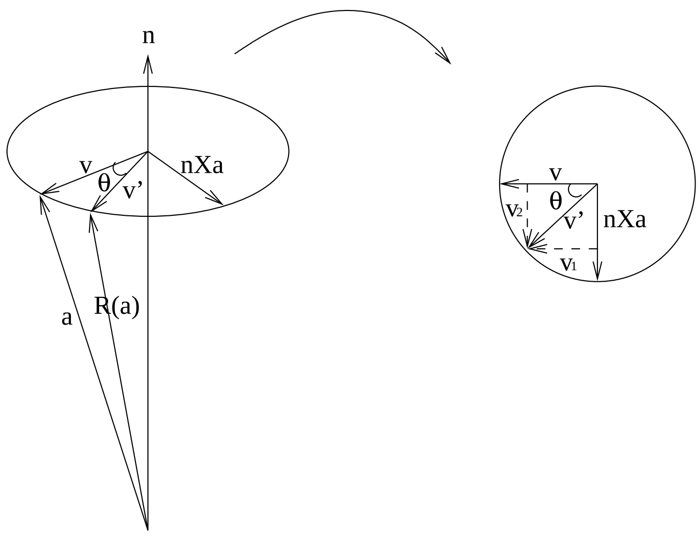


Figure 2: Axis-Angle specification. Projection of vector $a$ onto rotation plane and the result of rotation by $\theta$.

Given $n, \theta$ and $a$ we have the following:

- $v = [I - nn^t]a$ - Projection of $a$ onto the plane defined by $n$.
- $(n^t a)n = n(n^t a) = nn^t a$ - Part of $a$ which is parallel to $n$ (projection of $a$ onto $n$) .

We now look at the plane defined by $n$, Figure 2. The result of rotating $v$ in the plane is given by two perpendicular vectors:

$$\begin{aligned} v_1 &= |v|cos(\theta) * \tfrac{v}{|v|} \\ v_2 &= |v|sin(\theta) * \tfrac{n \times a}{|n||a|sin(\alpha)} = |v|sin(\theta) * \tfrac{n \times a}{|v|} \end{aligned} \tag{5}$$

According to Equations 5 and Figure 2 it is easy to see that the result of the rotation in the plane is given by:

$$v' = v_1 + v_2 = cos(\theta)[I - nn^t]a + sin(\theta)(n \times a) \tag{6}$$

Given Equation 6 we can build the rotation matrix which is:

$$R = nn^t + cos(\theta)[I - nn^t] + sin(\theta)N$$

Where

$$N = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}$$

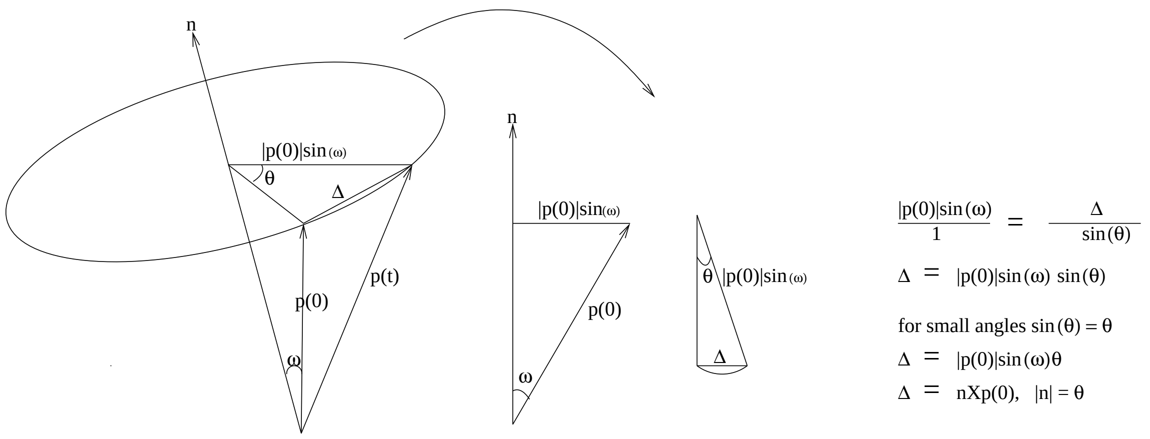


Figure 3: Axis-Angle specification. The tangential velocity at $p(0)$ is perpendicular to $n$ and to $p(0)$ and its size is $\Delta = n \times p(0)$

We now show the differential proof.
Consider a point $p(0)$ rotating with constant angular velocity around an axis $n$, $|n| = \theta$, Figure 3.

The points tangential velocity at time $t$ is given by (see Figure 3):

$$\begin{aligned} \dot{p}(t) &= n \times p(t) \\ &= Np(t) \end{aligned} \tag{7}$$

where again we represent the cross product by a skew symmetric matrix

$$N = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}$$

Equation 7 is a differential equation who's solution at time $t = \theta$ is given by:

$$p(\theta) = e^{N\theta} p(0)$$

We now evaluate $e^{N\theta}$ [1]:

$$e^{N\theta} = I + N\theta + \frac{(N\theta)^2}{2!} + \frac{(N\theta)^3}{3!} + \cdots$$

It is easy to verify that $N^3 = -N$. Substituting this into the previous equation yields:

$$\begin{aligned} e^{N\theta} &= I + \left(N\theta - \frac{N\theta^3}{3!} + \frac{N\theta^5}{5!} + \cdots\right) + \left(\frac{N^2\theta^2}{2!} - \frac{N^2\theta^4}{4!} + \cdots\right) \\ &= I + N\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \cdots\right) + N^2\left(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \cdots\right) \end{aligned}$$

[1] Matrix exponentiation is defined as follows: $e^{At} = \sum_{n=0}^{\infty} \frac{(At)^n}{n!} = \sum_{n=0}^{\infty} \frac{A^n t^n}{n!}$ with $A^0 \equiv I$

We identify the Taylor expansion of sin and cos in the previous equation which simplifies to:

$$e^{N\theta} = I + N\sin\theta + N^2(1-\cos\theta)$$

Extracting $n$ and $\theta$ from the matrix given in Equation 4 is done using the following formulas:

$$\begin{aligned} \theta &= acos(\frac{r_{11} + r22 + r_{33} - 1}{2}) \\ n &= \frac{1}{2sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \end{aligned}$$

**Note**:
The rotation axis is ill defined when the rotation angle approaches zero or $\pi$ and it is not defined when these angles are reached.

# 2 Quaternions

Quaternions are a mathematical object of the form

$$Q = s + ix + jy + kz = s + v$$

where $s, x, y, z \in R$, and $i, j, k$ are mutually orthogonal imaginary units with the following composition rule

| | i | j | k |
|---|---|---|---|
| i | -1 | k | -j |
| j | -k | -1 | i |
| k | j | -i | -1 |

We now define several basic operations on quaternions:

**Addition/Subtraction**
Given two quaternions $q_1 = [s_1, v_1]$ and $q_2 = [s_2, v_2]$ their addition/subtraction is defined as

$$q = q_1 \pm q_2 = [s_1 \pm s_2, v_1 \pm v_2]$$

**Multiplication**
Given two quaternions $q_1 = [s_1, v_1]$ and $q_2 = [s_2, v_2]$ their multiplication is defined as

$$\begin{aligned} q_1 * q_2 &= [s_1 + v_1] * [s_2 + v_2] \\ &= s_1 s_2 + s_1 v_2 + s_2 v_1 + v_1 * v_2 \end{aligned} \quad (8)$$

We now simplify Equation 8 by computing the multiplication of two 3-D vectors according to the above composition rules:

$$\begin{aligned}
v_1 * v_2 &= (iv_{1x} + jv_{1y} + kv_{1z}) * (iv_{2x} + jv_{2y} + kv_{2z}) \\
&= (-v_{1x}v_{2x} - v_{1y}v_{2y} - v_{1z}v_{2z}) + \\
&\quad i(v_{1y}v_{2z} - v_{1z}v_{2y}) + \\
&\quad j(v_{1z}v_{2x} - v_{1x}v_{2z}) + \\
&\quad k(v_{1x}v_{2y} - v_{1y}v_{2x}) \\
&= -(v1 \cdot v2) + (v_1 \times v_2)
\end{aligned} \tag{9}$$

Substituting Equation 9 into Equation 8 yields:

$$q = q_1 * q_2 = [(s_1 s_2 - v_1 \cdot v_2), (s_1 v_2 + s_2 v_1 + v_1 \times v_2)] \tag{10}$$

Quaternion multiplication can also be done in matrix form as follows:

$$q_1 * q_2 = \begin{bmatrix} s_1 & -v_{1x} & -v_{1y} & -v_{1z} \\ v_{1x} & s_1 & -v_{1z} & v_{1y} \\ v_{1y} & v_{1z} & s_1 & -v_{1x} \\ v_{1z} & -v_{1y} & v_{1x} & s_1 \end{bmatrix} q_2 = \Re q_2$$

or

$$q_2 * q_1 = \begin{bmatrix} s_1 & -v_{1x} & -v_{1y} & -v_{1z} \\ v_{1x} & s_1 & v_{1z} & -v_{1y} \\ v_{1y} & -v_{1z} & s_1 & v_{1x} \\ v_{1z} & v_{1y} & -v_{1x} & s_1 \end{bmatrix} q_2 = \overline{\Re} q_2$$

Note that $\overline{\Re}$ differs from $\Re$ in that the lower-right-hand $3 \times 3$ submatrix is transposed.

**Conjugate, Norm and Inverse**

The conjugate norm and inverse are given by the following three equations:

$$\overline{q} = [s, -v] \tag{11}$$

$$N(q) = ||q|| = \sqrt{q * \overline{q}} = \sqrt{s^2 + |v|^2} \tag{12}$$

$$q^{-1} = \frac{1}{q} = \frac{1}{q} * \frac{\overline{q}}{\overline{q}} = \frac{\overline{q}}{N(q)^2} \tag{13}$$

If $N(q) = 1$ then the quaternion is referred to as a unit quaternion and $\overline{q} = q^{-1}$.

## 2.1 Quaternions as rotational operators

<u>**Claim**</u>:
Given two vectors $a$ and $b$ and a unit quaternion of the form $q = [cos\frac{\theta}{2}, n\sin\frac{\theta}{2}]$ the general rotation of $a$ into $b$ about an arbitrary unit axis $n$ by $\theta$ radians is given by the equation:

$$b = q * a * q^{-1} \tag{14}$$

<u>**Proof**</u>:
We will show that Equation 14 is equivalent to the application of the matrix 4 to the vector $a$.

We start by deriving the general product form given by $q * v * q^{-1}$:

$$\begin{aligned}
&[s,u]*[0,v]*[s,-u] = \\
&= [s,u]*[v\cdot u, sv - v\times u] \\
&= [s(v\cdot u) - u\cdot(sv - v\times u), s^2v - s(v\times u) + (v\cdot u)u + u\times(sv - v\times u)] \\
&= [s(v\cdot u) - s(v\cdot u) + u\cdot(v\times u), s^2v + (v\cdot u)u - s(v\times u) + s(u\times v) - u\times(v\times u)] \\
&= [0, s^2v + (v\cdot u)u + 2s(u\times v) + (u\cdot v)u - (u\cdot u)v] \\
&= [0, 2(u\cdot v)u + (s^2 - u\cdot u)v + 2s(u\times v)]
\end{aligned} \tag{15}$$

Where the vector equality $a\times(b\times c) = (a\cdot c)b - (a\cdot b)c$ was used in line 4 of Equation 15.

Expanding Equation 14 using Equation 15 yields:

$$\begin{aligned}
b &= [0, 2(n\sin\frac{\theta}{2}\cdot a)n\sin\frac{\theta}{2} + (2cos^2\frac{\theta}{2} - 1)a + 2cos\frac{\theta}{2}(n\sin\frac{\theta}{2}\times a)] \\
&= [0, 2\sin^2\frac{\theta}{2}(n_xa_x + n_ya_y + n_za_z)n + (2cos^2\frac{\theta}{2} - 1)a + \\
&\quad 2\sin\frac{\theta}{2}cos\frac{\theta}{2}(n_ya_z - n_za_y, n_za_x - n_xa_z, n_xa_y - n_ya_x)]
\end{aligned}$$

We now look at $b_x$ and employ the trigonometric equalities $\sin^2\frac{\theta}{2} = \frac{1}{2}(1-\cos\theta),\ cos^2\frac{\theta}{2} = \frac{1}{2}(1+\cos\theta),\ and \sin\frac{\theta}{2}cos\frac{\theta}{2} = \frac{1}{2}\sin\theta$:

$$\begin{aligned}
b_x &= 2n_xn_x\sin^2\frac{\theta}{2}a_x + 2n_yn_x\sin^2\frac{\theta}{2}a_y + \\
&\quad 2n_zn_x\sin^2\frac{\theta}{2}a_z + (2cos^2\frac{\theta}{2} - 1)a_x + \\
&\quad 2n_y\sin\frac{\theta}{2}cos\frac{\theta}{2}a_z - 2n_z\sin\frac{\theta}{2}cos\frac{\theta}{2}a_y \\
&= n_xn_x(1-\cos\theta)a_x + n_yn_x(1-\cos\theta)a_y + n_zn_x(1-\cos\theta)a_z + \\
&\quad \cos\theta a_x + n_y\sin\theta a_z - n_z\sin\theta a_y
\end{aligned} \tag{16}$$

Looking at Equation 16 we see that it is exactly the result that is obtained by applying the axis angle rotation matrix (Equation 4) to the vector $a$ which proves the desired equivalence (a similar derivation for $b_y$ and $b_z$ completes the proof).

Note that rotation by $q$ is equivalent to rotation by $-q$ (easily seen from Equation 15).

Finally, we rewrite Equation 15 in a more computationally efficient way. Observing that for a unit quaternion [s,u] we have $s^2 - u \cdot u = 1 - 2(u \cdot u)$ and $2(u \cdot v)u - 2(u \cdot u)v = 2u \times (u \times v)$ we rewrite Equation 15 as:

$$[s, u] * [0, v] * [s, -u] = [0, v + 2s(u \times v) + 2u \times (u \times v)$$

## 2.2 Conversions between Representations

**Quaternion to Matrix:**
Given the quaternion $[s, (x, y, z)]$ the rotation matrix it defines is given by:

$$R = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2sz & 2xz + 2sy \\ 2xy + 2sz & 1 - 2x^2 - 2z^2 & 2yz - 2sx \\ 2xz - 2sy & 2yz + 2sx & 1 - 2x^2 - 2y^2 \end{bmatrix}$$

**Matrix to Quaternion:**
Given the previous matrix $R$ the corresponding quaternion is computed as follows: First compute $s$:

$$r_{11} + r_{22} + r_{33} = 3 - 4(x^2 + y^2 + z^2) = 3 - 4(1 - s^2) = 4s^2 - 1$$

This yields $s^2$. Now the quaternion entries are given by:

$$\begin{aligned} s &= \pm\frac{1}{2}\sqrt{r_{1,1} + r_{2,2} + r_{3,3} + 1} \\ x &= \frac{r_{3,2} - r_{2,3}}{4s} \\ y &= \frac{r_{1,3} - r_{3,1}}{4s} \\ z &= \frac{r_{2,1} - r_{1,2}}{4s} \end{aligned}$$

The sign chosen for $s$ is not important, yields same rotation. We have a problem if $s = 0$, this can be solved as follows:

1. Find the maximal element on the diagonal, element $(i, i)$ and set $j = (i+1) mod 3$ and $k = (j + 1) mod 3$.

2. Compute the following:

$$w = \sqrt{r_{i,i} - r_{j,j} - r_{k,k} + 1}$$

3. Given $q[1] \equiv x,\ q[2] \equiv y,\ q[3] \equiv z$

$$\begin{aligned} q[i] &= \frac{w}{2} \\ q[j] &= \frac{r_{i,j} + r_{j,i}}{2w} \\ q[k] &= \frac{r_{i,k} + r_{k,i}}{2w} \end{aligned}$$

# 3 Rotation matrices with small angles

A general rotation matrix in fixed axis specification with the rotation angles $(\omega_x, \omega_y, \omega_z)$ looks like this:

$$\begin{bmatrix} c\omega_z c\omega_y & c\omega_z s\omega_y s\omega_x - s\omega_z c\omega_x & c\omega_z s\omega_y c\omega_x + s\omega_z s\omega_x \\ s\omega_z c\omega_y & s\omega_z s\omega_y s\omega_x + c\omega_z c\omega_x & s\omega_z s\omega_y c\omega_x - c\omega_z s\omega_x \\ -s\omega_y & c\omega_y s\omega_x & c\omega_y c\omega_x \end{bmatrix}$$

Given that the angles are small we can approximate this matrix using the following standard approximations: $sin(\theta) \simeq \theta$, $cos(\theta) \simeq 1$ and $sin(\theta_1)sin(\theta_2) \simeq 0$. These approximations come from observing that the Taylor series around $x = 0$ ($f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}}{n!}(x - x_0)^n$) for $sin(x)$ and $cos(x)$ are:

$$\begin{aligned} sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \ldots &\simeq& \; x + O(x^3) \\ cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \ldots &\simeq& \; 1 + O(x^2) \end{aligned}$$

Finally the resulting approximation is:

$$\begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix}$$

# 4 Rotation Interpolation

Given two orientations we would like to smoothly interpolate between them.
Suggested solutions:

1. Take the two rotation matrices $R_0, R_1$ and linearly interpolate them by interpolating the matrix entries.

   $$Interp(R_0, R_1, t) = (1 - t)R_0 + tR_1$$

   The result is not a rotation matrix and must be projected onto the space of rotation matrices, renormalized.

2. Extract the rotation angles from the two matrices and linearly interpolate between them.

   $$Interp(R_0, R_1, t) = R_{xyz}((1 - t)\theta_{x0} + t\theta_{x1}, (1 - t)\theta_{y0} + t\theta_{y1}, (1 - t)\theta_{z0} + t\theta_{z1})$$

   The result will not be a smooth interpolation due to the fact that we are simultaneously interpolating three different angles.

3. Linearly interpolate between the two quaternions representing the rotation.

$$Interp(q_0, q_1, t) = (1 - t) * q_0 + t * q_1$$

The result gives a non-uniform change during changes of the interpolation parameter (see Figure 4).

4. Interpolate between the quaternions using spherical linear interpolation.

$$\begin{aligned} Interp(q_0, q_1, t) &= \frac{\sin((1-t)\Omega)}{\sin(\Omega)} q_0 + \frac{\sin(t\Omega)}{\sin(\Omega)} q_1 \\ cos(\Omega) &= q_0 \cdot q_1 \end{aligned}$$

The result is a smooth interpolation, equal changes of the interpolation parameter give equal changes in rotation space(see Figure 4). A problem with interpolations on the sphere is which direction to choose, through $\Omega$ or through $2\pi - \Omega$. A simple resolution of this ambiguity is to choose the interpolation along the shortest path. Naively we would choose the smaller angle as corresponding to the shortest path. This is not true on the unit quaternion sphere due to the feature that the rotation associated with $q$ is equivalent to the rotation associated with $-q$. So, to choose the shortest path between two rotations $q_0$ and $q_1$ check the differences $|q_0 - q_1|$ and $|q_0 + q_1|$ and choose the quaternion $q_1$ or $-q_1$ corresponding to the smaller of the two. Figure 5 illustrates this situation.

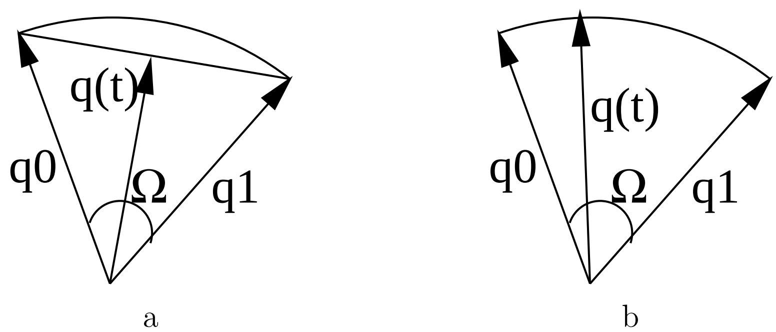


Figure 4: Linear interpolation of quaternions (Lerp). Spherical linear interpolation of quaternions (Slerp).

# A Gimbal Lock

The first question I asked myself when I heard the term *gimbal lock* is what is a gimbal? The following definition is taken from [4]:

> Gimbal - A contrivance, consisting of a ring or base on an axis, which permits an object, as a ships compass, mounted in or on it to tilt freely in any direction, in effect, suspending the object so that it will remain horizontal even when its support is tipped. Also called gimbal ring.

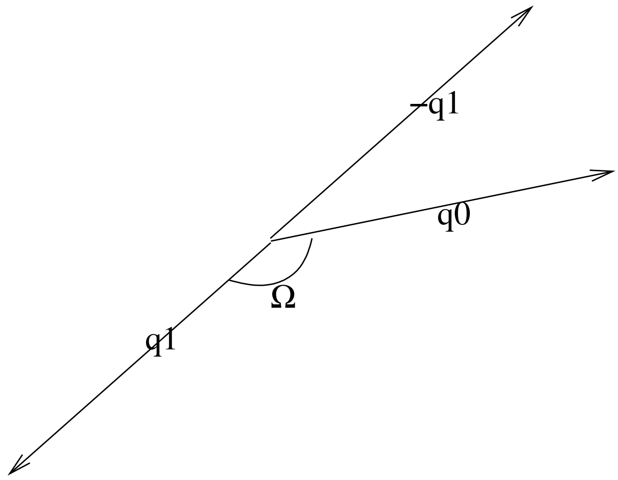


Figure 5: Choice of shortest path on quaternion unit sphere.

Gimbal lock derives its name from a mechanical problem in the gimbal mechanism, loss of a degree of freedom when certain rotations are performed. When considering rotation matrices stating that there is a loss of a degree of freedom is a bit misleading. When specifieng the matrix entries using Euler angles there is no loss of degrees of freedom. Given three angles $(\omega_x\omega_y\omega_z)$ there is a one to one mapping to a matrix $R$. Unfortunatly when given a specific matrix $R$ there isn't always a one to one mapping back to rotation angles. Here we have our loss of degree of freedom. To illustrate this let us look at our rotation matrix

$$\begin{bmatrix} c\omega_z c\omega_y & c\omega_z s\omega_y s\omega_x - s\omega_z c\omega_x & c\omega_z s\omega_y c\omega_x + s\omega_z s\omega_x \\ s\omega_z c\omega_y & s\omega_z s\omega_y s\omega_x + c\omega_z c\omega_x & s\omega_z s\omega_y c\omega_x - c\omega_z s\omega_x \\ -s\omega_y & c\omega_y s\omega_x & c\omega_y c\omega_x \end{bmatrix}$$

setting $\omega_y = \frac{\pi}{2}$ yields the following matrix:

$$\begin{bmatrix} 0 & c\omega_z s\omega_x - s\omega_z c\omega_x & c\omega_z c\omega_x + s\omega_z s\omega_x \\ 0 & s\omega_z s\omega_x + c\omega_z c\omega_x & s\omega_z c\omega_x - c\omega_z s\omega_x \\ -1 & 0 & 0 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 0 & s(\omega_x - \omega_z) & c(\omega_x - \omega_z) \\ 0 & c(\omega_x - \omega_z) & -s(\omega_x - \omega_z) \\ -1 & 0 & 0 \end{bmatrix}$$

Given a matrix with this form we cannot extract $\omega_x$ and $\omega_z$ we can only compute $(\omega_x - \omega_z)$. We have a loss of degree of freedom, if we set $\omega_x$ to be a certain value this determines the value of $\omega_z$.

Note that if $\omega_y = -\frac{\pi}{2}$ we can only compute $(\omega_x + \omega_z)$.

# References


[1] Craig J.J., “Introduction to robotics: mechanics and control”, Addison-Wesley, 2nd ed., chp. 2, 1989.

[2] Funda Janez, ”Quaternions and Homogeneous Transforms in Robotics”, M.Sc thesis, School of Engineering and Applied Science, University of Pennsylvania, 1988.

[3] Watt A. and Watt M., “Advanced Animation and Rendering Techniques Theory and Practice”, Addison-Wesley, chp. 15, 1992.

[4] Webster’s Encyclopedic Unabridged Dictionary of the English Language, Gramercy, 1996.