A Concise Note on Coordinate Frames Transformation and Quaternion Conventions

Mohammad Hussein Yoosefian Nooshabadi

December 27, 2021

1 Introduction

There are several factors that play role in successfully developing algorithms for robotic and aerospace systems. Two basic, thus often underestimated, topics are ‘Coordinate Frames Transformation’ and ‘Quaternion Conventions’. These two topics can be too tricky, especially since different authors have various notations [1]. This note tries to explain the topics in a straightforward and short but comprehensive manner. There are many sources that have thoroughly covered these topics such as [2] and [3]. This note is mostly a brief summary of the two latter references.

2 Rotation and Translation Between Coordinate Frames

In this note, ${}^{I}\mathbf{R}_C$ denotes the rotation matrix which transfers vectors from frame C to frame I. Its corresponding quaternion is ${}^{I}_{C}\mathbf{q}$. So if ${}^{C}\mathbf{v}$ is a vector expressed in C, one can write:

$$ {}^{I}\mathbf{v} = {}^{I}\mathbf{R}_C \, {}^{C}\mathbf{v} $$

in which ${}^{I}\mathbf{v}$ is the vector $\mathbf{v}$ expressed in I.

Furthermore, ${}^{I}\mathbf{p}_C$ denotes the position of the origin of the frame C with respect to the origin of the frame I expressed in frame I. When writing the translation and rotation between two frames, where there is more than one intermediate frame between them, we use *pre-multiplication* rule. As an example, consider Fig. 1, where a visual-inertial setup (a camera and an IMU that are rigidly attached to a body) is shown in three consecutive instants, along with a global frame. The blue frame is the IMU frame and the red frame is the frame attached to the camera. It is desired to calculate the rotation matrix and the translation vector between frame G and frame I_2. We do this using two distinct paths (in Fig. 1, carefully pay attention to the directions of the curves (rotation matrices) and the lines (translation vectors)):

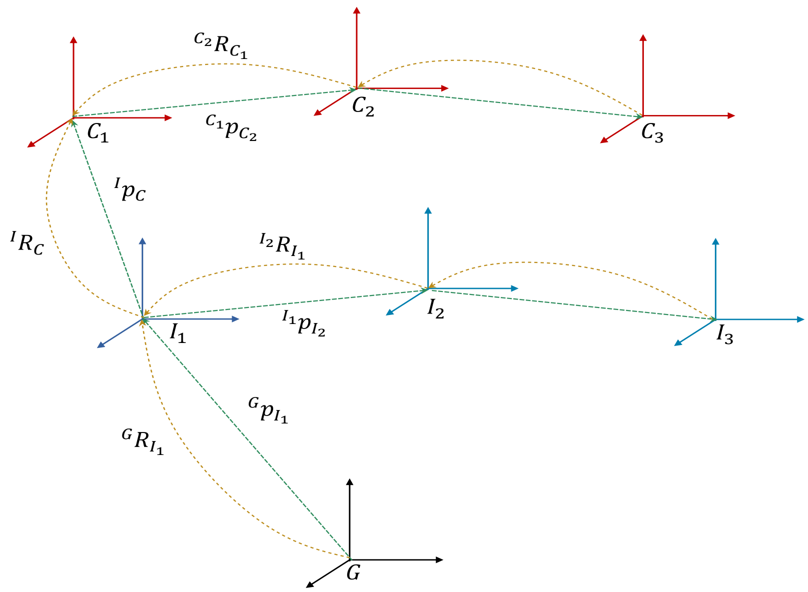

Figure 1: Coordinate frames of a VIO platform in three consecutive instants.

- From G to I_1 to I_2:

$$
\begin{aligned}
{}^{I_2}\mathbf{R}_G &= {}^{I_2}\mathbf{R}_{I_1}\,{}^{G}\mathbf{R}_{I_1}^{-1} \\
{}^{G}\mathbf{p}_{I_2} &= {}^{G}\mathbf{p}_{I_1} + {}^{G}\mathbf{R}_{I_1}\,{}^{I_1}\mathbf{p}_{I_2}
\end{aligned}
$$

- From G to I_1 to C_1 to C_2 to I_2:

$$
\begin{aligned}
{}^{I_2}\mathbf{R}_G &= {}^{I}\mathbf{R}_C\,{}^{C_2}\mathbf{R}_{C_1}\,{}^{I}\mathbf{R}_C^{-1}\,{}^{G}\mathbf{R}_{I_1}^{-1} \\
{}^{G}\mathbf{p}_{I_2} &= {}^{G}\mathbf{p}_{I_1} + {}^{G}\mathbf{R}_{I_1}\,{}^{I}\mathbf{p}_C + {}^{G}\mathbf{R}_{C_1}\,{}^{C_1}\mathbf{p}_{C_2} - {}^{G}\mathbf{R}_{C_2}\,{}^{I}\mathbf{p}_C
\end{aligned}
$$

  where:

$$
\begin{aligned}
{}^{G}\mathbf{R}_{C_1} &= {}^{G}\mathbf{R}_{I_1}\,{}^{I}\mathbf{R}_C \\
{}^{G}\mathbf{R}_{C_2} &= {}^{G}\mathbf{R}_{I_1}\,{}^{I}\mathbf{R}_C\,{}^{C_2}\mathbf{R}_{C_1}^{-1}
\end{aligned}
$$

3 Quaternion Conventions

A quaternion set comprises a vector and a scalar. Depending on being active or passive, being right-handed or left-handed, etc., one can define 12 different quaternions. Two most common unit quaternions are *Hamiltonian* and *JPL*. The *Hamiltonian* quaternion is mostly used in Robotics, while the *JPL* convention, as its name suggests, is mostly used in Aerospace applications. In this section, the most useful properties of these two quaternion conventions are discussed.

3.1 Hamiltonian Quaternion

This quaternion convention was introduced by Sir William Rowan Hamilton and is mostly used in applied sciences (mathematics and physics) and robotics (is used in Eigen library, ROS, Google Ceres, and MATLAB). This quaternion is right-handed and passive. In this quaternion the vector part comes *after* the scalar part:

$$\mathbf{q} = \begin{pmatrix} q_w & \mathbf{q}_v^\mathsf{T} \end{pmatrix}^\mathsf{T} \tag{1}$$

where $\mathbf{q}_v = \begin{pmatrix} q_x & q_y & q_z \end{pmatrix}^\mathsf{T}$ is the vector part and q_w is the scalar part. The rotation matrix corresponding to this quaternion is:

$$\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_xq_y - q_wq_z) & 2(q_xq_z + q_wq_y) \\ 2(q_xq_y + q_wq_z) & q_w^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_yq_z - q_wq_x) \\ 2(q_xq_z - q_wq_y) & 2(q_yq_z + q_wq_x) & q_w^2 + q_z^2 - q_x^2 - q_y^2 \end{bmatrix} \tag{2}$$

A random vector such as $\boldsymbol{\eta} = \begin{pmatrix} \eta_1 & \eta_2 & \eta_3 \end{pmatrix}^\mathsf{T}$ can be transformed into quaternion space using:

$$\mathbf{q} = \exp_q(\boldsymbol{\eta}) = \begin{pmatrix} \cos\left(\|\boldsymbol{\eta}\|_2\right) \\ \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2} \sin\left(\|\boldsymbol{\eta}\|_2\right) \end{pmatrix} \tag{3}$$

in which $\|\cdot\|_2$ is Euclidean norm. In some cases where $\|\boldsymbol{\eta}\|_2 \ll 1$ one can approximate equation (3) to:

$$\mathbf{q} = \exp_q(\boldsymbol{\eta}) \approx \begin{pmatrix} 1 \\ \boldsymbol{\eta} \end{pmatrix} \tag{4}$$

Let $\mathbf{p}$ and $\mathbf{q}$ be two Hamiltonian quaternions. the quaternion multiplication is defined as:

$$\mathbf{p} \odot \mathbf{q} = \begin{pmatrix} p_wq_w - \mathbf{p}_v \cdot \mathbf{q}_v \\ p_w\mathbf{q}_v + q_w\mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{pmatrix} = \begin{bmatrix} p_wq_w - p_xq_x - p_yq_y - p_zq_z \\ p_wq_x + p_xq_w + p_yq_z - p_zq_y \\ p_wq_y - p_xq_z + p_yq_w + p_zq_x \\ p_wq_z + p_xq_y - p_yq_x + p_zq_w \end{bmatrix} \tag{5}$$

Note that quaternion multiplication is non-commutative, i.e.:

$$\mathbf{p} \odot \mathbf{q} \neq \mathbf{q} \odot \mathbf{p}$$

Also, two $(\cdot)^R$ and $(\cdot)^L$ operators on a quaternion like $\mathbf{p}$ are defined as:

$$\mathbf{p}^R = \begin{pmatrix} p_w & -\mathbf{p}_v^\mathsf{T} \\ \mathbf{p}_v & p_w\mathbf{I}_3 - [\mathbf{p}_v \times] \end{pmatrix} \tag{6}$$

$$\mathbf{p}^L = \begin{pmatrix} p_w & -\mathbf{p}_v^\mathsf{T} \\ \mathbf{p}_v & p_w\mathbf{I}_3 + [\mathbf{p}_v \times] \end{pmatrix} \tag{7}$$

where the $[\cdot\times]$ operator for a random vector like $\boldsymbol{\eta}$ is defined as:

$$[\boldsymbol{\eta}\times] = \begin{bmatrix} 0 & -\eta_3 & \eta_2 \\ \eta_3 & 0 & -\eta_1 \\ -\eta_2 & \eta_1 & 0 \end{bmatrix} \tag{8}$$

If $\boldsymbol{\omega} = (\omega_x \quad \omega_y \quad \omega_z)^\mathsf{T}$ is a random vector and $\mathbf{q}$ is a quaternion, one can define:

$$\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} = \mathbf{q} \otimes \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \tag{9}$$

in which

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{10}$$

3.2 JPL Quaternion

This quaternion was introduced by the Navigation and Ancillary Information Facility, also known as NAIF, which is a subsection of the Jet Propulsion Laboratory at NASA. This quaternion is also passive, but unlike Hamiltonian quaternion, is left-handed. This quaternion is mostly used by NASA for aerospace applications. In this quaternion the vector part comes *before* the scalar part:

$$\mathbf{q} = (\mathbf{q}_v^\mathsf{T} \quad q_w)^\mathsf{T} \tag{11}$$

in which $\mathbf{q}_v = (q_x \quad q_y \quad q_z)^\mathsf{T}$ is the vector part and q_w is the scalar part. Let $\mathbf{p}$ and $\mathbf{q}$ be two JPL quaternions. The quaternion multiplication is defined as:

$$\mathbf{q} \otimes \mathbf{p} = \begin{pmatrix} q_w \mathbf{p}_v + p_w \mathbf{q}_v - \mathbf{q}_v \times \mathbf{p}_v \\ p_w q_w - \mathbf{q}_v \cdot \mathbf{p}_v \end{pmatrix} = \begin{bmatrix} q_w p_x + q_z p_y - q_y p_z + q_x p_w \\ -q_z p_x + q_w p_y + q_x p_z + q_y p_w \\ q_y p_x - q_x p_y + q_w p_z + q_z p_w \\ -q_x p_x - q_y p_y - q_z p_z + q_w p_w \end{bmatrix} \tag{12}$$

Moreover, the two $(\cdot)^R$ and $(\cdot)^L$ operators for a JPL quaternion like $\mathbf{p}$ are defined as:

$$\mathbf{p}^R = \begin{pmatrix} p_w \mathbf{I}_3 + [\mathbf{p}_v \times] & \mathbf{p}_v \\ -\mathbf{p}_v^\mathsf{T} & p_w \end{pmatrix} \tag{13}$$

$$\mathbf{p}^L = \begin{pmatrix} p_w \mathbf{I}_3 - [\mathbf{p}_v \times] & \mathbf{p}_v \\ -\mathbf{p}_v^\mathsf{T} & p_w \end{pmatrix} \tag{14}$$

in which the operator $[\cdot\times]$ is defined as in equation (8). Also, for the quaternion multiplication:

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} &= \mathbf{q}^L \mathbf{p} \\ &= \mathbf{p}^R \mathbf{q} \end{aligned} \tag{15}$$

If $\boldsymbol{\omega} = (\omega_x \quad \omega_y \quad \omega_z)^\mathsf{T}$ is a random vector and $\mathbf{q}$ is a quaternion, one can define:

$$\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} = \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \mathbf{q} \tag{16}$$

in which

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{17}$$

4 Relation Between the Two Quaternion Conventions

Let $\mathbf{q}_H$ be a Hamiltonian quaternion:

$$\mathbf{q}_H = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}$$

Its equivalent JPL quaternion will be:

$$\mathbf{q}_J = \begin{bmatrix} -q_x & -q_y & -q_z & q_w \end{bmatrix} \tag{18}$$

Also, note that **all quaternion conventions correspond to a single rotation matrix**, as they are just different ways to describe a **single** rotation:

$$\mathbf{R}_H = \mathbf{R}_J \tag{19}$$

As a result, to avoid confusion, it is recommended to use the rotation matrices and convert the rotation matrices to their corresponding quaternion format, only whenever needed. As mentioned earlier, all MATLAB commands regarding quaternions, assume the Hamiltonian convention. For instance, the

```
rotm2quat()
```

command converts a rotation matrix to its Hamiltonian quaternion form. If one works with the JPL convention, they can convert the quaternion resulted from this command to the JPL format using equations (18).

References

[1] H. Sommer, I. Gilitschenski, M. Bloesch, S. Weiss, R. Siegwart, and J. Nieto, "Why and how to avoid the flipped quaternion multiplication," *Aerospace*, vol. 5, no. 3, p. 72, 2018.

[2] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[3] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.