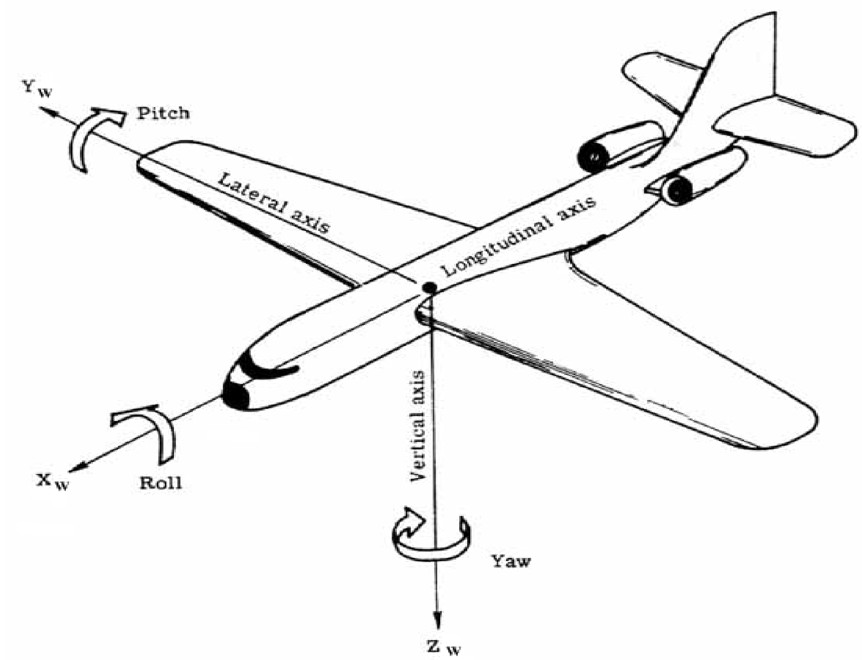
3D Orientation (Attitude) and Coordinate Frames

This section discusses how to define the orientation of a body in 3D.
There are different conventions to define orientation in 3D. We use the following one that is based on 3 angles:



A platform and its rotational degrees of freedom (DoF)
(Image from NASA, "mtp.jpl.nasa.gov/notes/pointing/Aircraft_Attitude2.png")

These three angular coordinates can be understood as a sequence of 3 pure rotations:

a) Rotation about the *z*-axis by the *yaw* angle.
b) Rotation about the transformed (once rotated) *y*-axis by the *pitch* angle.
c) Rotation about the transformed (twice-rotated) *x*-axis by the *roll* angle.

Any orientation in the 3D space can be achieved by a combination of these 3 pure rotations.
The associated rotation matrix (to transform points from the body coordinate frame to the navigation frame) is the composition of a sequence of basic transformations, each one being a pure rotation.

$$R\left(\varphi_x,\varphi_y,\varphi_z\right)=R\left(0,0,\varphi_z\right)\cdot R\left(0,\varphi_y,0\right)\cdot R\left(\varphi_x,0,0\right)$$

Where the pure rotations are expressed by the following rotation matrixes:

$$R(\varphi_x,0,0)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\varphi_x) & -\sin(\varphi_x)\\ 0 & \sin(\varphi_x) & \cos(\varphi_x)\end{bmatrix},\quad R(0,\varphi_y,0)=\begin{bmatrix}\cos(\varphi_y) & 0 & \sin(\varphi_y)\\ 0 & 1 & 0\\ -\sin(\varphi_y) & 0 & \cos(\varphi_y)\end{bmatrix}$$

$$R(0,0,\varphi_z)=\begin{bmatrix}\cos(\varphi_z) & -\sin(\varphi_z) & 0\\ \sin(\varphi_z) & \cos(\varphi_z) & 0\\ 0 & 0 & 1\end{bmatrix}$$

Interpretation:

Suppose we have 2 coordinate frames, A and B, those frames have the same origin but are rotated.
In order to align the coordinate frame A with the coordinate frame B we perform the following sequence of rotations: We are in frame A and we rotate our body respect to the current axis ***Z*** by an angle=φ. Our new orientation is now
defining a new (intermediate) coordinate frame. Now we rotate respect to axis Y in this current coordinate frame (call this axis ***Y'***) for an angle=φ . Now, based on our current orientation, we define in a new intermediate coordinate frame. We finally rotate respect to the axis X (of the current coordinate frame, call this axis ***X''***) by an angular displacement=φ . Now our orientation defines the final coordinate frame, it means we produced 3 pure rotations in order to align both coordinate frames. (transform the initial one to the final one).
Any given point, expressed in frame A can be expressed in frame B by applying a transformation between coordinate frames. We can understand this process by doing it gradually for each intermediate transformation.
In order to express a given 3D point $p^{(A)}$ (originally expressed in frame A) in the first intermediate coordinate frame we just need to apply a proper rotation about axis Z of frame A, i.e.

$$p' = R(0,0,-\varphi_z)\cdot p^{(A)}$$

(note that we are applying the matrix multiplication by left multiplying a column vector).

As we have the point expressed in the first intermediate coordinate frame we can easily express it in the second coordinate frame by considering the pure rotation about the axis Y'.

$$p'' = R(0,-\varphi_y,0)\cdot p' = R(0,-\varphi_y,0)\cdot R(0,0,-\varphi_z)\cdot p^{(A)}$$

Finally we express this point in the final coordinate frame by considering that implies just a pure rotation respect to Z''.

$$p^{(B)} = p''' = R(-\varphi_x,0,0)\cdot p''$$

$$p^{(B)} = R(-\varphi_x,0,0)\cdot R(0,-\varphi_y,0)\cdot R(0,0,-\varphi_z)\cdot p^{(A)}$$

This means that given a point defined in the frame A we can express it in respect to frame B by applying the rotation matrix $R_{A,B} = R(-\varphi_x,0,0)\cdot R(0,-\varphi_y,0)\cdot R(0,0,-\varphi_z)$.
(this matrix operates by *left-multiplying* a 3D column vector that express a 3D point)

What happen if we have a point expressed in frame B and we need to express it in frame A?
→ We need to apply the inverse of $R_{A,B}$

$$R_{B,A} = \left(R_{A,B}\right)^{-1} = \left(R\left(-\varphi_x,0,0\right)\cdot R\left(0,-\varphi_y,0\right)\cdot R\left(0,0,-\varphi_z\right)\right)^{-1}$$

If we manipulate this expression, we get

$$\begin{aligned}
&\left(R\left(-\varphi_x,0,0\right)\cdot R\left(0,-\varphi_y,0\right)\cdot R\left(0,0,-\varphi_z\right)\right)^{-1} = \\
&= R\left(0,0,-\varphi_z\right)^{-1}\cdot R\left(0,-\varphi_y,0\right)^{-1}\cdot R\left(-\varphi_x,0,0\right)^{-1} = \\
&= R\left(0,0,\varphi_z\right)\cdot R\left(0,\varphi_y,0\right)\cdot R\left(\varphi_x,0,0\right) \\
&\Downarrow \\
&R_{B,A} = R\left(0,0,\varphi_z\right)\cdot R\left(0,\varphi_y,0\right)\cdot R\left(\varphi_x,0,0\right)
\end{aligned}$$

Comment: There are other conventions to define coordinate frames. (As exercise you can propose one definition based on the following sequence: rotate x, then y' then z'')

If we want to have a compact expression of $R_{B,A} = R\left(\varphi_x,\varphi_y,\varphi_z\right)$ we need to multiply the pure rotation matrixes, for obtaining the full matrix. The result is:

$$\begin{bmatrix}
\cos(\varphi_y)\cdot\cos(\varphi_z) & -\cos(\varphi_x)\cdot\sin(\varphi_z)+\sin(\varphi_x)\cdot\sin(\varphi_y)\cdot\cos(\varphi_z) & \sin(\varphi_x)\cdot\sin(\varphi_z)+\cos(\varphi_x)\cdot\sin(\varphi_y)\cdot\cos(\varphi_z) \\
\cos(\varphi_y)\cdot\sin(\varphi_z) & \cos(\varphi_x)\cdot\cos(\varphi_z)+\sin(\varphi_x)\cdot\sin(\varphi_y)\cdot\sin(\varphi_z) & -\sin(\varphi_x)\cdot\cos(\varphi_z)+\cos(\varphi_x)\cdot\sin(\varphi_y)\cdot\sin(\varphi_z) \\
-\sin(\varphi_y) & \sin(\varphi_x)\cdot\cos(\varphi_y) & \cos(\varphi_x)\cdot\cos(\varphi_y)
\end{bmatrix}$$

A situation that usually happens is that certain measurements are expressed respect to the coordinate frame defined by the body of a platform. For instance, the angular velocities measured by an IMU are in respect to the current coordinate frame of the sensor (usually rigidly fixed to the platform's body). Similar situation happens with the measurements of on-board radars, laser scanners, cameras and other sensors, which may be attached to the platform. Consequently, it is necessary to process certain measurements to estimate the platform position and attitude and to understand the context where the platform (robot) is operating. Those estimates of the platform's need to be expressed respect to some fixed coordinate frame.

A case of interest for us is about how the Gravity vector is projected (and measured) by the accelerometers according to the orientation of the sensor. The measured accelerations are function of the Pitch and Roll (and of the intensity of the gravity, of course). The projection of the gravity vector, measured by accelerometers, can be used to estimate pitch and roll of a platform, in absolute terms.