Vehicle-IMU orientation calibration

The following method is used to rotate the axii of an IMU arbitrarily placed on a vehicle. The rotation aims in matching the axii of the IMU and the vehicle frame (see picture) in terms of orientation, **not** in terms of frame origin.

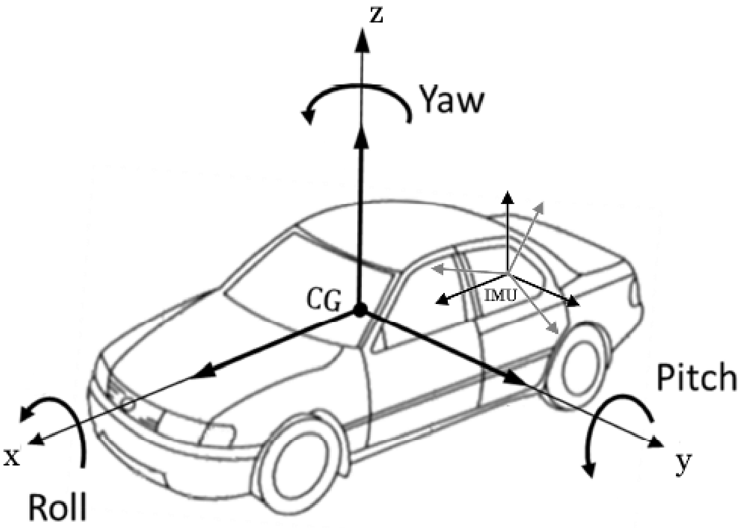

Figure 1: Vehicle coordinate frame and IMU coordinate frames. The original IMU orientation (grey) is rotated to match the orientation of the vehicle coordinate frame (black).

The arbitrary orientation of the IMU can be described by rotating the vehicle's coordinate frame around its axii 3 times (one for each axis), as shown below:

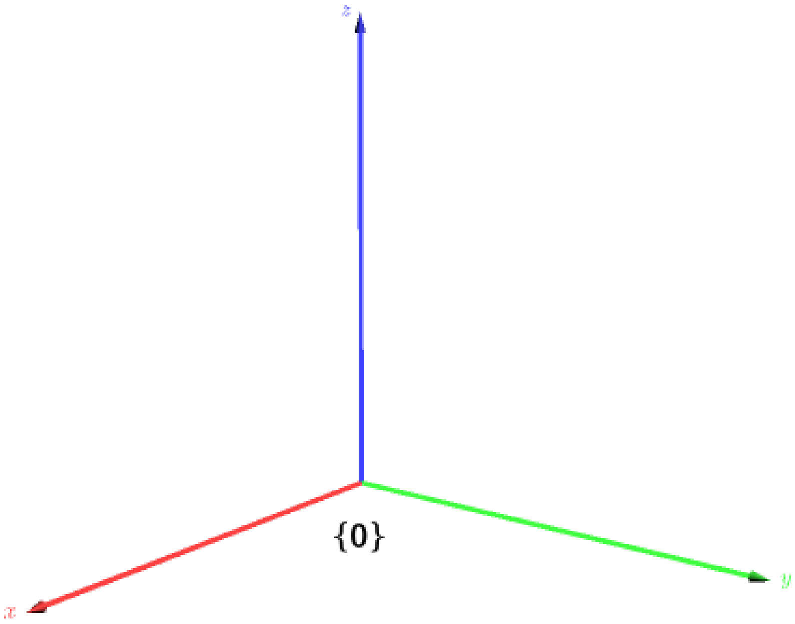

Figure 2: Original vehicle frame orientation.

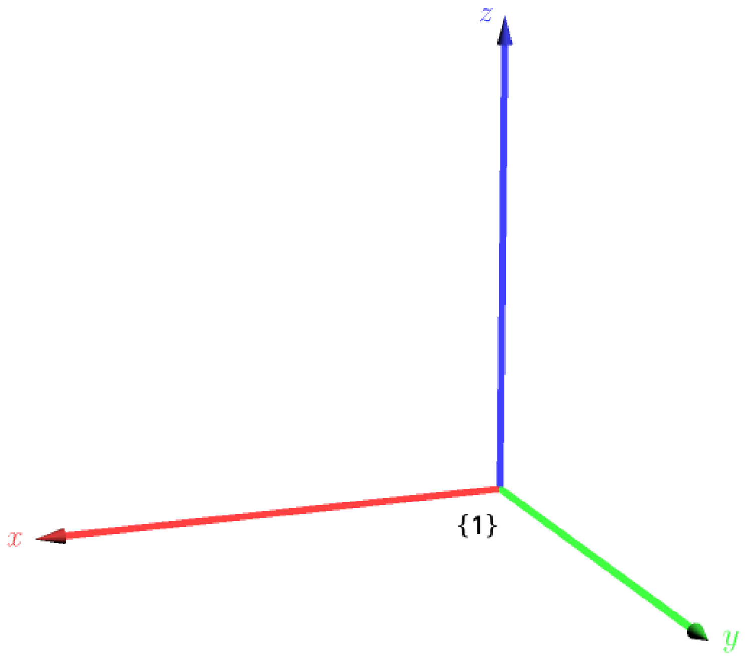

Figure 3: Result of rotating original vehicle frame around z-axis by -30°. This frame and the original one share the same z-axis.

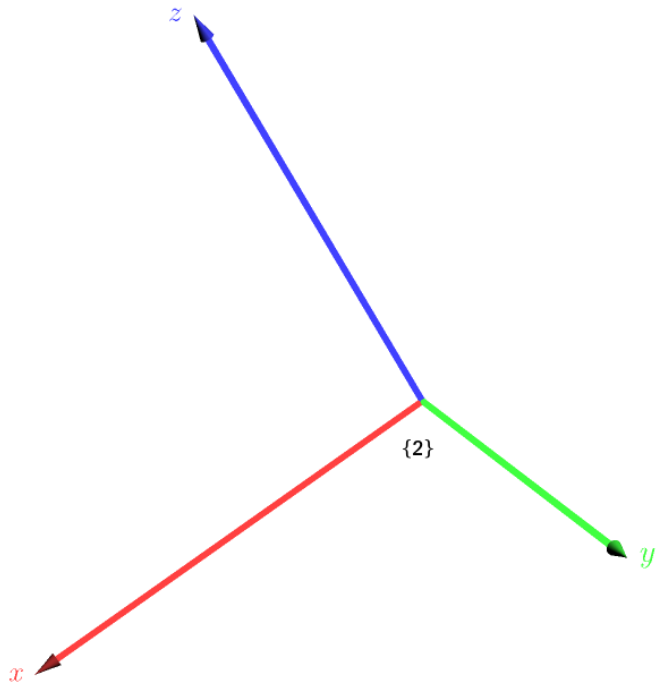

Figure 4: Result of rotating original vehicle frame around z-axis by -30°, followed by rotating around y-axis by 30°.

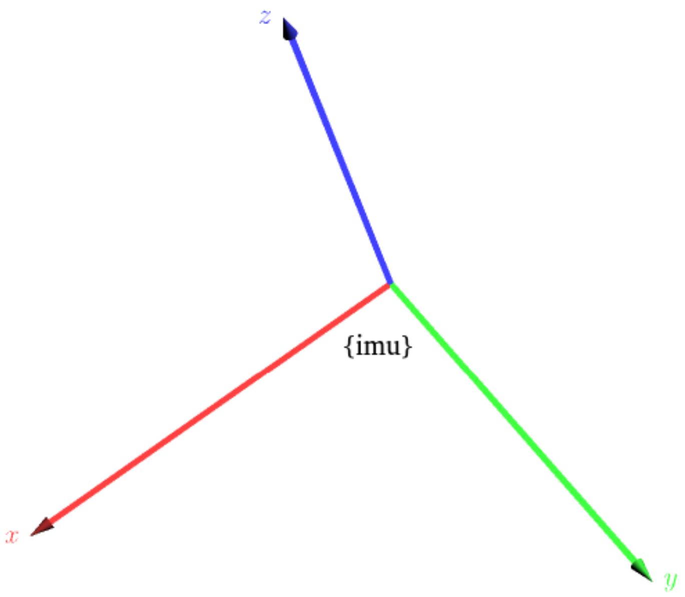

Figure 5: Result of rotating original vehicle frame around z-axis by -30°, followed by rotating around y-axis by 30°, followed by rotating around x-axis by -30°.

If we assume the acceleration at each frame is denoted by $\vec{\boldsymbol{a}}_n$ where n is the name of the frame (see pictures), then the following holds:

$$\vec{\boldsymbol{a}}_0 = R_z \cdot \vec{\boldsymbol{a}}_1 \tag{1}$$

$$\vec{\boldsymbol{a}}_1 = R_y \cdot \vec{\boldsymbol{a}}_2 \tag{2}$$

$$\vec{\boldsymbol{a}}_2 = R_x \cdot \vec{\boldsymbol{a}}_{imu} \tag{3}$$

where:

$$R_z = rodrigues\Big(\begin{bmatrix}0 & 0 & 1\end{bmatrix},\ c\Big)$$

$$R_y = rodrigues\Big(\begin{bmatrix}0 & 1 & 0\end{bmatrix},\ b\Big)$$

$$R_x = rodrigues\Big(\begin{bmatrix}1 & 0 & 0\end{bmatrix},\ a\Big)$$

This means that the acceleration in the calibrated frame $\{0\}$ is connected to the measured acceleration of the IMU through:

$$\vec{\boldsymbol{a}}_0 = R_z \cdot R_y \cdot R_x \cdot \vec{\boldsymbol{a}}_{imu} \tag{4}$$

$$\Leftrightarrow$$

$$\vec{\boldsymbol{a}}_{imu} = R_x^T \cdot R_y^T \cdot R_z^T \cdot \vec{\boldsymbol{a}}_0 \tag{5}$$

Expanding these equations results in:

$$\boxed{\vec{\boldsymbol{a}}_0 = \underbrace{\begin{bmatrix} \cos(b)\cos(c) & \cos(c)\sin(a)\sin(b) - \cos(a)\sin(c) & \sin(a)\sin(c) + \cos(a)\cos(c)\sin(b) \\ \cos(b)\sin(c) & \cos(a)\cos(c) + \sin(a)\sin(b)\sin(c) & \cos(a)\sin(b)\sin(c) - \cos(c)\sin(a) \\ -\sin(b) & \cos(b)\sin(a) & \cos(a)\cos(b) \end{bmatrix}}_{R_{total}} \vec{\boldsymbol{a}}_{imu}} \tag{6}$$

The point of the calibration is to find the angles a, b, c in order to rotate the measured IMU acceleration $\vec{\boldsymbol{a}}_{imu}$ to $\vec{\boldsymbol{a}}_0$. As it will be explained below, the calibration has to be calculated in two stages.

Solving for angles

Static calibration

The first step calculates the angles a and b. These have to be calculated when the gravitational vector is constant (does not change orientation, because its norm should be constant). Furthermore, the vehicle must be parallel to the sea level (flat road) and not stand on a slope, otherwise the calculated angles will contain a component related to the slope. These two requirements are satisfied when the vehicle is in a standstill on a flat road, therefore the calibration can be done while vehicle is **static**.

The result of this first stage matches the z-axis of the two frames, thus rotating the vector $\vec{\boldsymbol{a}}_{imu} \rightarrow \vec{\boldsymbol{a}}_1$ (frame $\{1\}$). The $\vec{\boldsymbol{a}}_1$ is related to $\vec{\boldsymbol{a}}_{imu}$:

$$\vec{\boldsymbol{a}}_1 = R_y \cdot R_x \cdot \vec{\boldsymbol{a}}_{imu} \Leftrightarrow$$

$$\boxed{\vec{\boldsymbol{a}}_1 = \underbrace{\begin{bmatrix} \cos(b) & \sin(a)\sin(b) & \cos(a)\sin(b) \\ 0 & \cos(a) & -\sin(a) \\ -\sin(b) & \cos(b)\sin(a) & \cos(a)\cos(b) \end{bmatrix}}_{R_{1,imu}} \vec{\boldsymbol{a}}_{imu}} \tag{7}$$

If the vehicle is not in a slope and is not moving, then only the gravitational force is exerted on the IMU sensor. Therefore, the x and y component of the $\vec{\boldsymbol{a}}_1$ should be zero. Using the above equation:

$$\begin{gathered} a_{1_y} = 0 \Leftrightarrow \\ \cos(a) \cdot a_{imu_x} = \sin(a) \cdot a_{imu_z} \end{gathered} \tag{8}$$

and

$$\begin{gathered} a_{1_x} = 0 \Leftrightarrow \\ \cos(b) \cdot a_{imu_x} + \sin(a)\sin(b) \cdot a_{imu_y} + \cos(a)\sin(b) \cdot a_{imu_z} = 0 \xLeftrightarrow{\cdot \sin(a)} \\ \sin(a)\cos(b) \cdot a_{imu_x} + \sin^2(a)\sin(b) \cdot a_{imu_y} + \cos(a)\sin(b)\underbrace{\sin(a) \cdot a_{imu_z}}_{(8)} = 0 \Leftrightarrow \\ \sin(a)\cos(b) \cdot a_{imu_x} + \sin(b)\Big(\sin^2(a) \cdot a_{imu_y} + \cos^2(a) \cdot a_{imu_y}\Big) = 0 \Leftrightarrow \\ \sin(a)\cos(b) \cdot a_{imu_x} + \sin(b) \cdot a_{imu_y} = 0 \end{gathered} \tag{9}$$

Assuming $a \neq k\dfrac{\pi}{2}$ and $a_{imu_z} \neq 0$, we get using (8):

$$\boxed{a = \arctan 2\big(a_{imu_y}, a_{imu_z}\big)} \tag{10}$$

While, assuming $b \neq k\dfrac{\pi}{2}$ and $a_{imu_y} \neq 0$, we get using (9):

$$\boxed{b = \arctan 2\big(- \sin(a) \cdot a_{imu_x}, a_{imu_y}\big)} \tag{11}$$

Dynamic calibration

After rotating $\{imu\}$ frame so its z-axis orientation matches the one with the vehicle's frame ($\{1\}$ frame), we need to calculate the angle c which matches the x and y axii of $\{1\}$ frame to that of the vehicle's frame ($\{0\}$ frame).

In order to proceed with the second stage of the calibration, the angles a and b must have already been found by the first calibration stage. These are necessary since the second stage relies on the $\vec{\boldsymbol{a}}_1$ vector, as described below.

The acceleration in the vehicle frame ($\vec{\boldsymbol{a}}_0$) is connected to $\vec{\boldsymbol{a}}_1$ as seen in (1):

$$\vec{\boldsymbol{a}}_0 = R_z \cdot \vec{\boldsymbol{a}}_1 \Leftrightarrow$$
$$\vec{\boldsymbol{a}}_0 = \begin{bmatrix} \cos(c) & -\sin(c) & 0 \\ \sin(c) & \cos(c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{\boldsymbol{a}}_1$$

Assuming the vehicle is **accelerating** in a straight line, the y component of $\vec{\boldsymbol{a}}_0$ should be zero. This hypothesis gives the solution for angle c, assuming $c \neq k\frac{\pi}{2}$ and $a_{1_x} \neq 0$:

$$\boxed{c = \arctan 2\big(- a_{1_y}, a_{1_x}\big)} \tag{12}$$

Real-world scenario problems

In practice, acceleration readings tend to be noisy. This means that calibration (both static and dynamic) needs to account for errors produced due to the noisy signals. Using only one reading for $\vec{\boldsymbol{a}}_{imu}$ should be avoided. Instead, it is encouraged that the conclusion should be drawn from multiple values, spanning a period of time. Some form of low-pass filtering should also prove effective in increasing algorithm robustness.

When it comes to **static** calibration, the vehicle, as we described above, is in a standstill. Therefore, the actual acceleration of the vehicle should be constant (only gravity applies). However, due to noise, the readings present with some form of oscillations around the actual acceleration values. We can average out each acceleration axis component to remove the noise and only keep the DC values.

Dynamic calibration on the other hand is more complex. Since the vehicle can not be in a standstill, actual acceleration changes over time through actual physical world phenomena and noise at the same time. We need to clear the noise from the signal, without affecting the effect of these real world phenomena. Therefore, moving average filtering of a time window should be used, designed not to skew the effect of acceleration-changing phenomena.

Furthermore, we need to detect when these real world phenomena affecting acceleration take place and only use these data for dynamic calibration. Braking seems to be easier to detect than throttling, since commercial vehicles in general have greater braking capabilities than throttling. A threshold method based on the speed provided by the GPS (along with a low-pass filter) should prove effective in detecting braking/throttling events.