Attitude and Heading Reference System

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

Robotic Systems
Control of **avionic systems** implies to determine the **attitude** of an aircraft in terms of its **Euler angles**:

- Roll
- Pitch
- Yaw

Inertial sensors normally used are **not able** to directly sense these angles.

The solution is to adopt special **complementary filters** or **Kalman filters** that are able to perform a "**sensor fusion**" among the various inertial data in order to estimate the attitude of the aircraft.
Gyroscopes

- They measure the **angular speeds (rates)** along the roll, pitch and yaw axis.

- By means of **numeric integration** we can approximately compute the roll, pitch and yaw angles.
Accelerometers

- They measure the **linear acceleration** along the roll, pitch and yaw axis
- They measure mechanical solicitations but also the **gravity vector**
- If we compute the **inclination** of the $g$ vector measured, we can estimate **roll** and **pitch**
Magnetometers

- They measure the **Earth magnetic field** along roll, pitch and yaw axis.
- By computing the **inclination** of the measured vector w.r.t. the magnetic north, we can estimate the **yaw**.
Approaches used for AHRS

- **PI Controller**: Complementary Filter
- **Kalman Gain**: Kalman Filter
AHRS: the Complementary Filter

1. Acceleration Data
2. Normalize
3. Gravity Vector from Accelerometers
4. Cross Product
5. Error on "G" compensation
6. Apply "G" compensation
7. PI Controller
8. Corrected Error Rates
9. Sum
10. Angular rates from gyroscope
11. Integrate Euler Angles
12. Euler Angles
13. Rotate Vector (0, 0, 1)
\textbf{AHRS: Attitude and Heading Reference System (caso Roll e Pitch)}

\texttt{while True do}
\begin{itemize}
\item On each $\Delta T$;
\item \{ax, ay, az\} $\leftarrow$ \texttt{read_accelerometers()};
\item \{gx, gy, gz\} $\leftarrow$ \texttt{read_gyroscopes()};
\item / \* Rotate Gravity Vector using $\phi, \theta, \psi$ \* /
\item \{gravX, gravY, gravZ\} $\leftarrow$ \texttt{rotate_vector(\{0, 0, 1\})};
\item / \* Normalize Acceleration Data \* /
\item \{ax, ay, az\} $\leftarrow \frac{\{ax, ay, az\}}{||\{ax, ay, az\}||}$;
\item / \* Compute Error using Cross Product \* /
\item \{ex, ey, ez\} $\leftarrow \{ax, ay, az\} \times \{gravX, gravY, gravZ\}$;
\item / \* Apply PI Controller to each element of the error vector \* /
\item \{corrX, corrY, corrZ\} $\leftarrow$ \texttt{PI_control(\{ex, ey, ez\})};
\item / \* Correct Gyro Measures \* /
\item \{gx, gy, gz\} $\leftarrow$ \{gx, gy, gz\} + \{corrX, corrY, corrZ\};
\item / \* Update angles using integration \* /
\item \{$\phi, \theta, \psi$\} $\leftarrow$ \{$\phi, \theta, \psi$\} + \{gx, gy, gz\}$\Delta T$;
\end{itemize}
\texttt{end}
When a rigid body is subject to mechanical solicitations, accelerometers will measure both the $g$ vector and such solicitations.

But they are “noise” with respect to the measurement algorithm.

We note that:

- In static conditions, we have $\sqrt{ax^2 + ay^2 + az^2} = g$
- In case of solicitations, we have $\sqrt{ax^2 + ay^2 + az^2} \neq g$

The $g$-compensation is applied by weighting the gyroscope correction on the basis of the difference:

$$\sqrt{ax^2 + ay^2 + az^2} - g$$
while True do
    On each $\Delta T$;
    ...
    $\{gravX, gravY, gravZ\} \leftarrow rotate\_vector(\{0, 0, 1\});$
    ...
end

The most critical part is the rotation of the vector $\{0, 0, 1\}$
Rotation of a Vector in $R^3$

Rotation along $x$ axis, roll

\[
\begin{bmatrix}
  x' \\
  y' \\
  z'
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \phi & -\sin \phi \\
  0 & \sin \phi & \cos \phi
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = R_\phi \begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
\]
Rotation along y axis, \textbf{pitch}

\[
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix}
= \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
= R_\theta
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]
Rotation of a Vector in $\mathbb{R}^3$

Rotation along $z$ axis, \textit{yaw}

$$
\begin{bmatrix}
  x' \\
  y' \\
  z'
\end{bmatrix} =
\begin{bmatrix}
  \cos \psi & -\sin \psi & 0 \\
  -\sin \psi & \cos \psi & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = R_\psi
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
$$
Rotazione di un Vettore in $R^3$

Rotazione intorno tutti gli assi usando **roll, pitch e yaw**

$$
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix} = R_{\phi} R_{\theta} R_{\psi}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = DCM
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
$$
The DCM is the **rotation matrix** of a rigid body whose attitude is expressed with Euler angles $\theta, \phi, \psi$

\[
\begin{bmatrix}
\cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\
\cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\
-\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \\
\end{bmatrix}
\]

For each step of the algorithm, we should compute the DCM
A **quaternion** is a **complex number** with four components, a real part and three imaginary parts:

\[ q = \{q_0, q_1, q_2, q_3\} = q_0 + q_1i + q_2j + q_3k \]

\( i, j, k \) are **imaginary units** and are characterised by the following properties:

\[ i^2 = -1, \quad j^2 = -1, \quad k^2 = -1 \]

\[ -ij = ij = k, \quad -jk = kj = i, \quad -ki = ik = j \]

Quaternions obey to an algebra in with the classical operations are defined: (algebraic) sum, products, ration, norm, etc.
Any vector $v = \{v_x, y, v_z\}$ in $R^3$, and rotated (on itself) of an angle $\alpha$ can be represented by a quaternion:

$$q = \{\cos\frac{\alpha}{2}, v_x\sin\frac{\alpha}{2}, v_y\sin\frac{\alpha}{2}, v_z\sin\frac{\alpha}{2}\}$$
Any vector $v = \{v_x, y_y, v_z\}$ in $\mathbb{R}^3$, and rotated (on itself) of an angle $\alpha$ can be represented as the vector $\{||v||, 0, 0\}$ to which two rotations (two angles) plus the angle $\alpha$ are applied, → the Euler angles.
If we consider a unit vector, a quaternion can represent that vector rotated according the Euler angles → a quaternion is an alternative representation of Euler angles $\phi, \theta, \psi$
Quaternions

Quaternions e DCM

\[ q = \{ q_0, q_1, q_2, q_3 \} \]

The DCM can be computed by a quaternion as::

\[
\begin{bmatrix}
\cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\
\cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\
-\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta
\end{bmatrix}
\]

\[
\begin{bmatrix}
q_0^2 - q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\
2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_3 q_2 + q_0 q_1) \\
2(q_1 q_3 + q_0 q_2) & 2(q_2 q_2 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2
\end{bmatrix}
\]
The gravity vector \( \{0, 0, 1\} \) can be rotated by means of a multiplication with the rotation matrix:

The rotated vector is the **third column** of the rotation matrix:

\[
\begin{bmatrix}
\cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\
\cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\
-\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \\
\end{bmatrix}
\]

By using a quaternion, trigonometric functions are not needed.
Integration of a quaternion

When Euler angles are represented by a quaternion $q$, we must update it on the basis of the angular speeds $\{\dot{\phi}, \dot{\theta}, \dot{\psi}\}$ provided by gyroscopes.

By using the derivation rules of quaternions, we have:

$$q = q + \frac{\Delta T}{2} \begin{bmatrix} 0 & -\dot{\phi} & -\dot{\theta} & -\dot{\psi} \\ \dot{\phi} & 0 & \psi & -\dot{\theta} \\ \dot{\theta} & -\psi & 0 & \dot{\phi} \\ \dot{\psi} & \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} q$$
The equivalence quaternion/rotation matrix allows us to determine the formulas to compute Euler angles from a quaternion:

\[
\phi = \tan^{-1} \left( \frac{2(q_0 q_1 + q_2 q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right)
\]

\[
\theta = \sin^{-1} \left( 2(q_0 q_2 + q_1 q_3) \right)
\]

\[
\psi = \tan^{-1} \left( \frac{2(q_1 q_2 + q_0 q_3)}{1 - 2(q_2^2 + q_3^2)} \right)
\]
Quaternions are used in the cyan blocks
Attitude and Heading Reference System

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

Robotic Systems