Positioning method and algorithm:

Needs:

- Calculate CubeSat orientation:
  - Estimate an angular rotation
- Determine the trajectory to have, to reach the desired orientation.
- Calculate the orientation:
  - Chose the priority (solar productivity / Tether orientation)
  - Determine the most interesting position
- Traduce it in physical input for actuators

Representation of attitude for a system:

To fix the attitude of any object we first need reference frame called (X, Y, Z) and then a frame for our mobile system (X', Y', Z').

a) Euler angles:

OXYZ basis is related to solid OX'Y'Z 'by three successive rotations:

-The Precession around Oz (going from OXYZ to OUVZ)

-The Wobble around OR (going from OUVZ to OUWZ ')

-The Own rotation around OZ '(going from OUWZ' to OX'Y'Z ')

$$\vec{\Omega} = \dot{\psi}\,\vec{z} + \dot{\theta}\,\vec{u} + \dot{\varphi}\,\vec{z}\,'$$

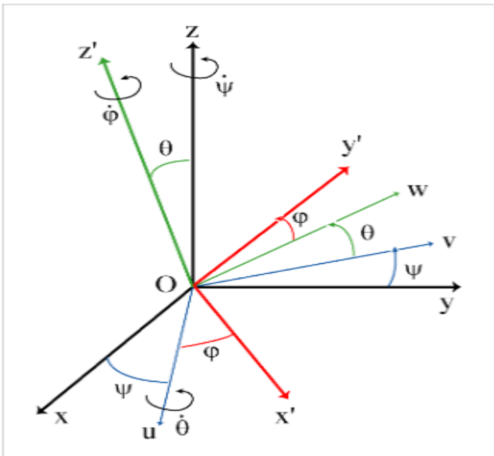

Figure 1 : Euler Angles representation

Then the instantaneous rotation vector is:

Thus, any vector x in a given base can be expressed in another frame as a composition of

$$\vec{t} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}_M = Rz'.Ru.\,Rz.\begin{bmatrix} x \\ y \\ z \end{bmatrix}_R$$

rotations:

With vectors of rotation:

$$Rz' = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Ru = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$Rz = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

However, there are singular points that prevent the orientation calculation in certain positions. Indeed, when the second rotation around the axis u is zero or multiple of π, it is impossible to differentiate the two other rotations because in this case the Z and Z ' axes are confused (related to cosine / sine)

And with the composition of rotations it is possible to set up symmetrical rotations {R121, R131, R212, R232, R313, R323} and antisymmetric rotations or gimbal angles {R123, R132, R213, R231, R312, R321}

b) Gimbal angles:

- roll angle (around X) defined in [-π, π]
- pitch angle (around Y) defined in [-π / 2, π / 2]
- yaw angle (around Z) defined in [-π, π]

As the Euler angles, the gimbal angles contain points called "Gimbal lock" (when the second angle theta is equal to +/- π / 2)

There are other representations which have no singular points (such as the representation of quaternions).

c) Quaternions:

This representation, unlike the Euler Angles, is not intuitive at all but the associated calculations are less complex. Thus it requires less computation power, time and less energy.

The quaternions respect the following properties:

$$i^2 = j^2 = k^2 = ijk = -1$$

The rotation quaternion is represented as such:

$$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = w + \vec{v}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \cos(\alpha/2) + \vec{u}\sin(\alpha/2)$$

Where $\vec{u}$ is a normalized vector that gives the direction of the rotation axis and α is the rotation angle in rad.

To rotate any vector $\vec{v}$ around the $\vec{u}$ axis by the α angle, we can apply the following equation:

$$\vec{v'} = q\vec{v}q^{-1} = \left(\cos\frac{\alpha}{2} + \vec{u}\sin\frac{\alpha}{2}\right)\vec{v}\left(\cos\frac{\alpha}{2} - \vec{u}\sin\frac{\alpha}{2}\right)$$

Where $\vec{v'}$ is the rotated vector.

In our case, we have the initial and the final attitude ($\vec{v}$ and $\vec{v'}$). We can use the quaternion representation to get the rotation quaternion (q) with relatively simple operations from a computational point of view. We can then deduce the rotation axis and angle and convert them to the Euler angle format, that we can use to calculate the output to the actuators.

d) Measuring the attitude:

Two non-collinear and non-zero vectors within two frames are sufficient to determine the attitude of a solid. In many systems they point on far fixed stars from the system (using Star Tracker), the Sun (using Sun sensors), or the Earth (using magnetometer and Earth sensors).

In the terrestrial reference frame, the magnetic fields and the gravitational acceleration are known

eg Paris g = 9.81 m/s and Bh = 20.6µT and Bv = 42.24µT with: $\vec{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} et\ \vec{B} = \begin{bmatrix} Bh \\ 0 \\ Bv \end{bmatrix}$

We must then measure the vectors of two fields in the mobile frame. Let's call A the accelerations on each axis and M the magnetometers measurements on the axis:

$$\begin{bmatrix} Ax \\ Ay \\ Az \end{bmatrix} = R.\vec{g} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

$$\begin{bmatrix} Mx \\ My \\ Mz \end{bmatrix} = R.\vec{B} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \cdot \begin{bmatrix} Bh \\ 0 \\ Bv \end{bmatrix}$$

By developing:

$$Ax = R_{13}.g \qquad Mx = R_{11}.Bh + R_{13}.Bv$$

$$Ay = R_{23}.g \qquad My = R_{21}.Bh + R_{23}.Bv$$

$$Az = R_{33}.g \qquad Mz = R_{31}.Bh + R_{33}.Bv$$

Then, using the Gimbal angles (R 321) the equations are obtained:

$$\begin{gathered} Ax = -\sin(\theta) \\ Ay = \sin(\psi).\cos(\theta) \\ Az = \cos(\psi).\cos(\theta) \\ Mx = \cos(\theta).\cos(\varphi).Bh - \sin(\theta).Bv \\ My = (\sin(\psi).\sin(\theta).\cos(\varphi) - \cos(\psi).\sin(\varphi)).Bh + \sin(\psi).\cos(\theta).Bv \\ Mz = (\cos(\psi).\sin(\theta).\cos(\varphi) + \sin(\psi).\sin(\varphi)).Bh + \cos(\psi).Bv \end{gathered}$$

Solving the system:

$$\begin{gathered} \theta = -\arcsin(Ax) \\ \psi = arctg2\left(\frac{Ay}{Az}\right) \\ \cos(\varphi) = \cos(\varphi)(Mx, Bh, Bv, \theta) \\ \sin(\varphi) = \sin(\varphi)(My, Bh, Bv, \theta, \psi, \cos(\varphi)) \\ \varphi = arctg2\left(\frac{\sin(\varphi)}{\cos(\varphi)}\right) \end{gathered}$$

Arctg 2 is equivalent to Arctg on - π / π

This method is rarely used because it requires a large number of trigonometric calculations.

e) TRIAD Algorithm:

Triad algorithm is one of the earliest and simplest solutions to the spacecraft attitude determination problem. It consists in constructing two orthonormal bases using two pairs of vector measurements.

Two in the orbital reference frame, noted $r_1$and r_2 and two in the body reference frame, noted b_1 and b_2, representing the same magnitude expressed in a different referential.

The following equations are used to build $R_b = \begin{bmatrix} t_{1b} & t_{2b} & t_{3b} \end{bmatrix}$, the basis attached to the body referential and $R_r = \begin{bmatrix} t_{1r} & t_{2r} & t_{3r} \end{bmatrix}$ the basis attached to the orbital referential.

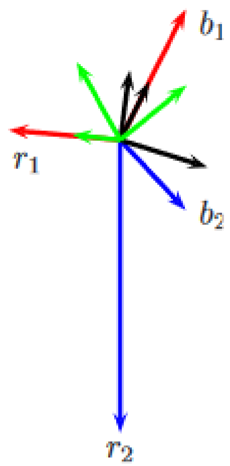

Figure 2 : [t1b t2b t3b] are represented in black and [t1r t2r t3r] in green

Given the knowledge of two vectors in the reference and body coordinates of a satellite, the TRIAD (TRIaxis Attitude Determination) algorithm obtains the direction cosine matrix relating both frames. The two vectors are typically the unit vector to the sun and the Earth's magnetic field vector (it can also be unit vector to two star using star tracker for example).

This algorithm is not an optimal solution, but it provides a reliable estimation of the satellite's attitude while being quite cheap regarding computation needs.

f) Kalman Filter:

The Kalman filter uses mathematical method to **filter signal from noise or inaccurate measure**. It is useful to determine position or orientation even with potential measurement errors. This filter can be used to filter, smooth or predict data (past/present/future). One of its advantage is that it **provides an estimation of the error**.

In a discrete context, the Kalman filter is a recursive estimator: to estimate the current state it only needs the previous state and the current measures.

To use Kalman filter, the system **needs** to be **linearly modeled**. But if the modeling is too approximate, the filter will not be efficient enough and the estimation error will not converge fast enough.

The Kalman filter has 2 distinct states:

- Prediction (using the previous state it estimates the actual state)
- Correct (uses measurement to correct the predicted state)

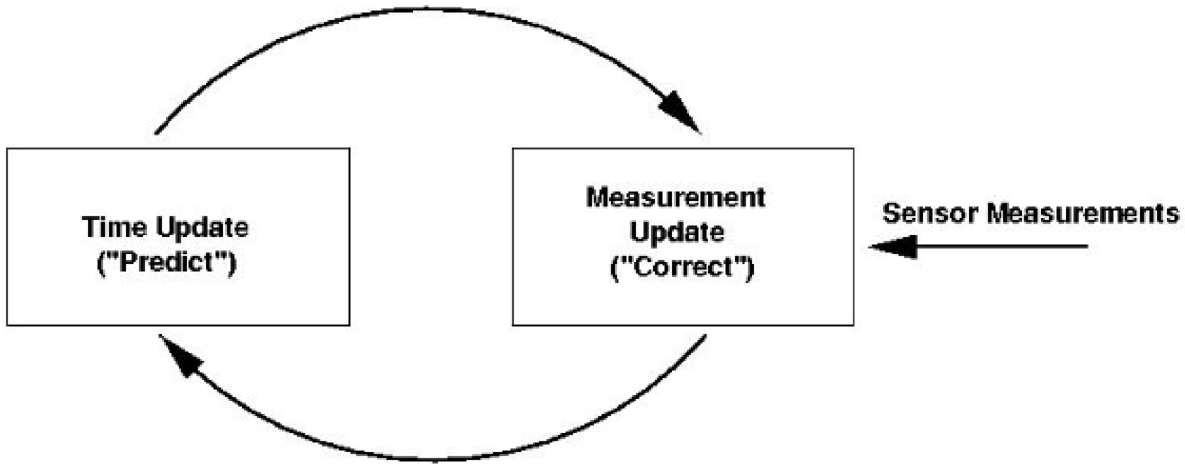

Figure 3 : The 2 states from the Kalman Filter

An extended version of the Kalman filter exist, the principal difference being the possibility to use differentiable function instead of linear (for observation and prediction).