Matlab functions for attitude representation:

* *C=euler3132DCM(euler313)* gets DCM (**3x3 matrix**) from a **vector** of Euler angles (in **radians**) in the sequence 3-1-3.
* *euler313=DCM2euler313(C)* gets a **vector** of Euler angles (in **radians**) in the sequence 3-1-3 from DCM (**3x3 matrix**). It is fixed that the second angles is between 0 y +180.
* *q=DCM2quat(C)* from DCM (**3x3 matrix**) generates the corresponding quaternion (**4-dimensional unit vector**).
* *C=quat2DCM(q)* from a quaternion (**4-dimensional unit vector**) generates the corresponding DCM (**3x3 matrix**).
* *qp=qprod(q1,q2)* gives a quaternion (**4-dimensional unit vector**) which is the result of the the product of two quaternions q1 and q2, in that order (**4-dimensional unit vectors**).
* *qc=qconj(q)* gives a quaternion (**4-dimensional unit vector**) which is the conjugate of another (**4-dimensional unit vector**).
* *[axis,angle]=q2euler(q)* gives the Euler’s axis (**3-dimensional** **unit vector**) and Euler’s angle (**radians**), restricted to 0 to +180, from a quaternion (**4-dimensional unit vector**).
* *q=euler2q(axis,angle)* allows to obtain the quaternion (**4-dimensional unit vector**) from Euler’s axis (**3-dimensional** **unit vector**) and Euler’s angle (**radians**).

The scripts DCM2euler123 and euler1232DCM are similar to the first two.

In addition the following function is provided

* OrientationView: a function to visualize an attitude given as a quaternion. The best way to do is with examples (scripts representation and attitude\_path).