Mathematics II Project
CITM - UPC

List of all the functions implemented:

1) Eaa2rotMat.m

   Explanation: This function takes the Euler Principal Angle and Axis and returns its Rotation Matrix. It automatically normalizes the Euler Principal Axis, so you can put a not normalized Axis. The Euler Principal angle must be given in degrees.
   Inputs: Euler P. Axis [3, 1], Euler P. Angle [1, 1].
   Output: Rotation Matrix [3, 3].

2) Quat2rotMat.m

   Explanation: This function takes a Quaternion and returns the Rotation Matrix. It automatically normalizes the quaternion.
   Input: Quaternion [4, 1].
   Output: Rotation Matrix [3, 3].

3) eAngles2rotM.m

   Explanation: This function takes the Euler Angles and transforms them into their Rotation Matrix. The Euler Angles must be given in degrees.
   Inputs: Euler Angle Phi [1, 1], Euler Angle Theta [1, 1], Euler Angle Psi [1, 1].
   Output: Rotation Matrix [3, 3].

4) rotVec2rotMat.m

   Explanation: This function takes a Rotation Vector and returns the corresponding Rotation Matrix. It automatically normalizes the vector.
   Input: Rotation Vector [3, 1].
   Output: Rotation Matrix [3, 3].

5) rotMat2EAA.m

   Explanation: This function does the opposite of the Eaa2rotMat.m function. It takes a Rotation Matrix and returns its corresponding Euler Principal Angle

and Axis. It will return the Angle in degrees and the Axis will be normalized. If the angle is 0 degrees, it returns one of the many possible Euler Axis, but if the angle is 180 degrees, it returns one of the two possible euler angles, because it doesn't consider the asymmetrical part of the Rodríguez formula.
Input: Rotation Matrix [3, 3].
Outputs: Euler P. Axis [3, 1], Euler P. Angle [1, 1].

6) rotM2Quat.m

Explanation: This function does the opposite of the Quat2rotMat.m function. It takes a Rotation Matrix and returns its corresponding Quaternion.
Input: Rotation Matrix [3, 3].
Output: Quaternion [4, 1].

7) rotM2eAngles.m

Explanation: This function does the opposite of the eAngles2rotM.m function. It takes a Rotation Matrix and returns its corresponding Euler Angles. It will return the Euler Angles in degrees.
Input: Rotation Matrix [3, 3].
Outputs: Euler Angle Phi [1, 1], Euler Angle Theta [1, 1], Euler Angle Psi [1, 1].

8) rotM2rotVec.m

Explanation: This function does the opposite of the rotVec2rotMat.m function. It takes a Rotation Matrix and returns its corresponding Rotation Vector.
Input: Rotation Matrix [3, 3].
Output: Rotation Vector [3, 1].