Orbital elements to ECI

By: Noe Lepez Da Silva Duarte Created: 05 Feb. 2022

```
function [r_x, r_y, r_z] = oe2eci(a,e,inc,RAAN,omega,t)
% This function computes the radius vector (ECI) for a given set of orbital
% elements
% - a = Semi-major axis [m]
% - e = orbit of interest eccentricity.
% - inc = inclination [deg]
% - RAAN = Right Ascension of Ascending Node [deg]
% - omega = Argument of Perigee [deg]
% - t = time since initial mean anomaly [s].
%
% The function output 'r_x, r_y, r_z' corresponds to the x, y, z components
% of the position vector.
%

% Find the true anomaly
M0 = 0;
E = Kepler_Solver(t, e, M0, a);
nu = 2*atan(sqrt((1+e)/(1-e))*tan(E/2));

% Calculate r in the perifocal coordinate frame
r = (a*(1-e^2))/(1+e*cos(nu));
r_vec = [r*cos(nu);r*sin(nu);0];

% Make orbital elements into radians for MATLAB
omega_rad = deg2rad(omega);
i_rad = deg2rad(inc);
RAAN_rad = deg2rad(RAAN);

% Create a array of rotation angles to rotate the r vector obtained in the
% perifocal from into the ECI frame
angle_vec = [-omega_rad -i_rad -RAAN_rad];
total_rot = r_vec;

% Loop through the 3-1-3 rotations
i=1;
for j=[3 1 3]
    if j == 1
        inter_rot = Rx(angle_vec(i));
    elseif j == 3
        inter_rot = Rz(angle_vec(i));
    end
    total_rot = inter_rot*total_rot;
    i = i+1;
end

% Create outputs
r_x = total_rot(1);
r_y = total_rot(2);
```

```
r_z = total_rot(3);

    % Rotation matrices as function as to be easily reused
    function R1=Rx(ph)
        R1 = [1 0 0;
              0 cos(ph) sin(ph);
              0 -sin(ph) cos(ph)];
    end

    function R3=Rz(ps)
        R3 = [cos(ps) sin(ps) 0;
             -sin(ps) cos(ps) 0;
             0 0 1];
    end

end
```

Published with MATLAB® R2021b