MEAM-5430 HW7

Problem 1

(a) This is intrisic roatation, sequence: (1) Y(pitch) 20deg, (2) X(roll) -30deg, (3) Y(pitch) -20deg, (4) X(roll) 30deg.

```
clear;
clc;

phi = 30 * pi / 180;
theta = 20 * pi / 180;
psi = 0;

R1 = euler2rotm321(0, theta, 0);
R2 = euler2rotm321(-phi, 0, 0);
R3 = euler2rotm321(0, -theta, 0);
R4 = euler2rotm321(phi, 0, 0);
R = R4 * R3 * R2 * R1;

[rx, ry, rz] = rotm2euler321(R);
rx = rx * (180 / pi)
```

```
rx = -1.4981
```

```
ry = ry * (180 / pi)
```

```
ry = -2.7636
```

```
rz = rz * (180 / pi)
```

```
rz = -9.7777
```

The euler angles of the final orientation are (X: -1.49deg, Y: -2.76deg, Z: -9.78deg).

(b) The three rotations are: (1) `Y(pitch) 90deg, (2) X(roll) 90deg, (3) Z(yaw) -90deg`

```
R1 = euler2rotm321(0, pi/2, 0);
R2 = euler2rotm321(pi/2, 0, 0);
R3 = euler2rotm321(0, 0, -pi/2);
R = R3 * R2 * R1
```

```
R = 3×3
   -1.0000   -0.0000   -0.0000
    0.0000    0.0000   -1.0000
    0.0000   -1.0000    0.0000
```

```


R*R'
```

```
ans = 3×3
    1.0000         0   -0.0000
         0    1.0000         0
   -0.0000         0    1.0000
```

```
R/R
```

```
ans = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
det(R)
```

```
ans = 1
```

```

[axis, angle] = rotm2axisangle(R);
axis
```

```
axis = 3×1
   -0.0000
    0.7071
   -0.7071
```

```
angle = angle * 180 / pi
```

```
angle = 180
```

```

[rx, ry, rz] = rotm2euler321(R);
rx = rx * (180 / pi)
```

```
rx = -90
```

```
ry = ry * (180 / pi)
```

```
ry = -3.5084e-15
```

```
rz = rz * (180 / pi)
```

```
rz = 180
```

The aircraft need to rotate euler angles of (x: -90deg, y: 0deg, z: 180deg) to get back to the original orientation. Note that there are computation accuracy losses when calculating this case.

```
function R = euler2rotm321(phi, theta, psi)
    R = [cos(theta)*cos(psi), cos(theta)*sin(psi), -sin(theta);
         sin(phi)*sin(theta)*cos(psi) - cos(phi)*sin(psi),
sin(phi)*sin(theta)*sin(psi) + cos(phi)*cos(psi), sin(phi)*cos(theta);
         cos(phi)*sin(theta)*cos(psi) + sin(phi)*sin(psi),
cos(phi)*sin(theta)*sin(psi) - sin(phi)*cos(psi), cos(phi)*cos(theta)];
end

function [phi, theta, psi] = rotm2euler321(R)
    if R(3,1) < 1
        if R(3,1) > -1
            theta = asin(-R(3,1));
            psi = atan2(R(2,1)/cos(theta), R(1,1)/cos(theta));
```

```matlab
            phi = atan2(R(3,2)/cos(theta), R(3,3)/cos(theta));
        else % B(3,1) = -1
            theta = pi/2;
            psi = atan2(-R(1,2), R(1,3));
            phi = 0;
        end
    else % B(3,1) = 1
        theta = -pi/2;
        psi = atan2(-R(1,2), R(1,3));
        phi = 0;
    end

    % Ensure the angles are within the range [-pi, pi]
    theta = wrapToPi(theta);
    psi = wrapToPi(psi);
    phi = wrapToPi(phi);
end

function [axis, angle] = rotm2axisangle(R)
    % Ensure the matrix is a proper rotation matrix
    % if abs(det(R) - 1) > 1e-2 || ~isequal(R'*R, eye(3), 'abs_tol', 1e-4)
    %     error('The provided matrix is not a valid rotation matrix');
    % end

    % Calculate the angle of rotation
    angle = acos((trace(R) - 1) / 2);

    % If the angle is very small, the axis can be arbitrary
    if abs(angle) < 1e-6
        axis = [1; 0; 0];  % Arbitrary axis
        angle = 0;
    elseif abs(angle - pi) < 1e-6
        % Angle is around pi, rotation axis is still defined
        % Find the largest diagonal entry to avoid a division by a small number
        [~, idx] = max([R(1,1), R(2,2), R(3,3)]);
        if idx == 1
            axis = (1/sqrt(2*(1+R(1,1)))) * [R(1,1)+1; R(2,1); R(3,1)];
        elseif idx == 2
            axis = (1/sqrt(2*(1+R(2,2)))) * [R(1,2); R(2,2)+1; R(3,2)];
        else
            axis = (1/sqrt(2*(1+R(3,3)))) * [R(1,3); R(2,3); R(3,3)+1];
        end
    else
        % Calculate the axis of rotation
        axis = 1/(2*sin(angle)) * [R(3,2) - R(2,3); R(1,3) - R(3,1); R(2,1) -
R(1,2)];
    end

    % Ensure the axis is a unit vector
    axis = axis / norm(axis);
```

```
end
```