AMZ 556 HW1 Shuai Zhao 7722927131

Problem 1.

$$\|\vec{v}\| = \sqrt{\vec{v}^T \cdot \vec{v}}$$

$$\|R\vec{v}\| = \sqrt{(R\vec{v})^T(R\vec{v})} = \sqrt{\vec{v}^T R^T R \vec{v}}$$

Since R is a rotation matrix, R is orthogonal, which means $R^T = R^{-1}$

$$R^T R = R^{-1} R = I$$

So,

$$\|R\vec{v}\| = \sqrt{\vec{v}^T R^T R \vec{v}} = \sqrt{\vec{v}^T \vec{v}} = \|\vec{v}\|$$

Problem 2

Similarly to Problem 1:

$$\begin{aligned} \|R\vec{p_1} - R\vec{p_2}\| &= \sqrt{(R\vec{p_1} - R\vec{p_2})^T (R\vec{p_1} - R\vec{p_2})} \\ &= \sqrt{(\vec{p_1} - \vec{p_2})^T R^T R (\vec{p_1} - \vec{p_2})} \\ &= \sqrt{(\vec{p_1} - \vec{p_2})^T (\vec{p_1} - \vec{p_2})} \\ &= \|\vec{p_1} - \vec{p_2}\| \end{aligned}$$

Contents

- Euler Angle & Rotation Matrix
- p & p1
- Plot

```
clc;
clear;
close all;
```

Euler Angle & Rotation Matrix

```
roll = pi/3;
pitch = -pi/4;
yaw = pi/2;

R = eul2rotm([yaw, pitch, roll], 'ZYX');
disp('Rotation Matrix R:');
disp(R);
```

```
Rotation Matrix R:
   0.000000000000000  -0.500000000000000   0.866025403784439
   0.707106781186548  -0.612372435695794  -0.353553390593274
   0.707106781186547   0.612372435695795   0.353553390593274
```

p & p1

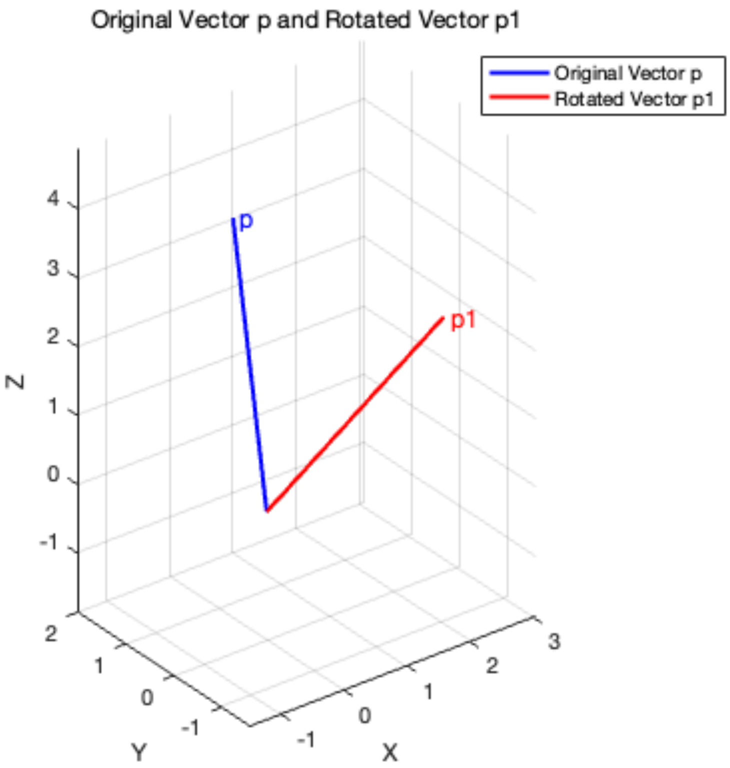
```
p = [1; 2; 3];
p1 = R * p;
disp('Coordinates of p1:');
disp(p1);
```

```
Coordinates of p1:
   1.598076211353316
  -1.578298261984863
   2.992511824357958
```

Plot

```
figure;
hold on;

plot3([0 p(1)], [0 p(2)], [0 p(3)], 'b', 'LineWidth', 2);
text(p(1), p(2), p(3), ' p', 'FontSize', 12, 'Color', 'b');

plot3([0 p1(1)], [0 p1(2)], [0 p1(3)], 'r', 'LineWidth', 2);
text(p1(1), p1(2), p1(3), ' p1', 'FontSize', 12, 'Color', 'r');

axis equal;
grid on;
xlabel('X');
```

```
ylabel('Y');
zlabel('Z');
title('Original Vector p and Rotated Vector p1');
legend('Original Vector p', 'Rotated Vector p1');
view(3);
hold off;
```



Published with MATLAB® R2023b

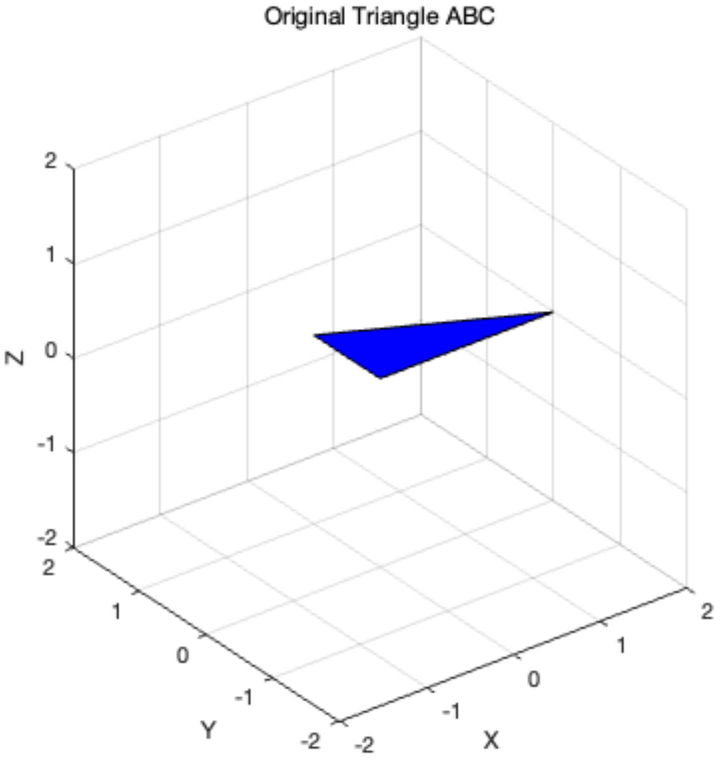
Contents

- (a) Triangle
- (b) rotate pi/6 about x-axis
- (c) rotate -pi/4 about y-axis
- (d) rotate 2pi/3 about z-axis
- (e) rotate back
- rotation function

```
clc;
clear;
close all;
```

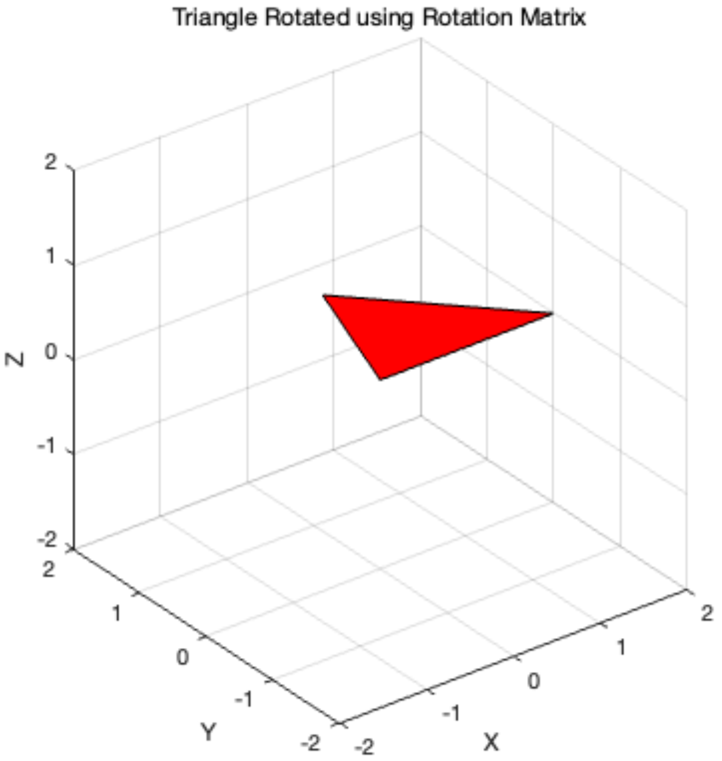
(a) Triangle

```
A0 = [0; 0; 0];
B0 = [2; 0; 0];
C0 = [0; 1; 0];
triangle0 = [A0, B0, C0];

figure;
view(3);
axis equal;
grid on;
patch(triangle0(1, :), triangle0(2, :), triangle0(3, :), 'blue');
xlabel('X');
ylabel('Y');
zlabel('Z');
title('Original Triangle ABC');

xlim([-2 2]);
ylim([-2 2]);
zlim([-2 2]);
```



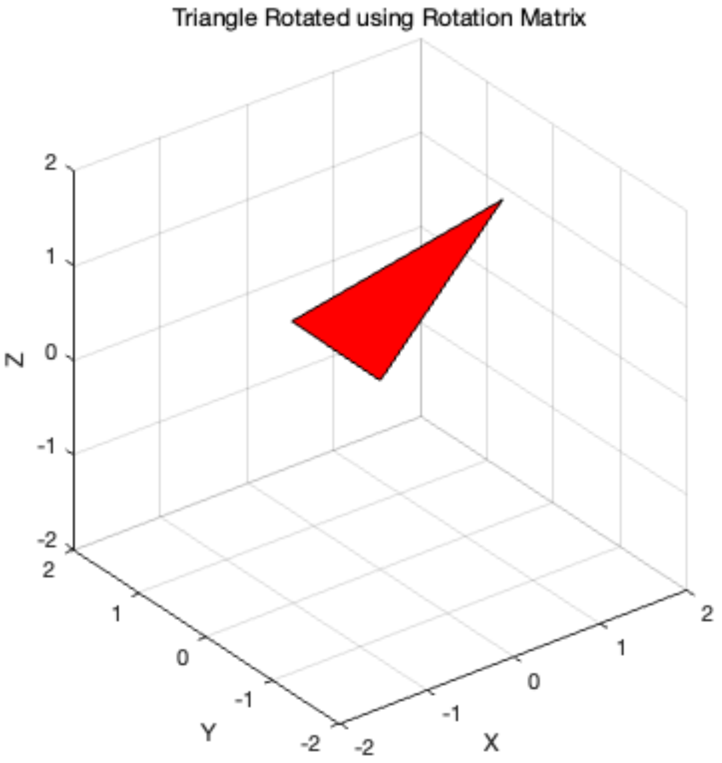
(b) rotate pi/6 about x-axis

```
R1 = eul2rotm([pi/6, 0, 0], 'XYZ');
triangle1 = rotateTriangle(triangle0, R1);
```



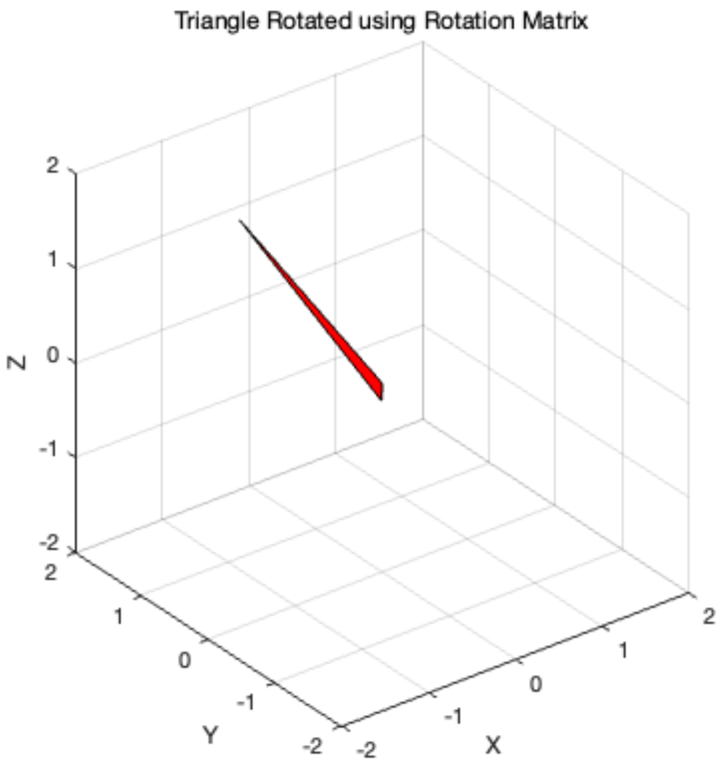
(c) rotate -pi/4 about y-axis

```
R2 = eul2rotm([0, -pi/4, 0], 'XYZ');
triangle2 = rotateTriangle(triangle1, R2);
```



(d) rotate 2pi/3 about z-axis

```
R3 = eul2rotm([0, 0, 2*pi/3], 'XYZ');
triangle3 = rotateTriangle(triangle2, R3);
```



(e) rotate back

```
R_total = R3 * R2 * R1;
R_inverse = inv(R_total);

triangle4 = rotateTriangle(triangle3, R_inverse);
```

rotation function

```
function rotatedTriangle = rotateTriangle(triangle, R)

    rotatedTriangle = R * triangle;

    % plot
    figure;
    view(3);
    axis equal;
    grid on;
    patch(rotatedTriangle(1, :), rotatedTriangle(2, :), rotatedTriangle(3, :), 'red');
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    title('Triangle Rotated using Rotation Matrix');

    xlim([-2 2]);
    ylim([-2 2]);
    zlim([-2 2]);
end
```

Published with MATLAB® R2023b

Problem 5

(1) Rotate by ϕ about world x-axis :

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

(2) Rotate by θ about world z-axis

$$R_2 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) Rotate by ψ about current x-axis

$$R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

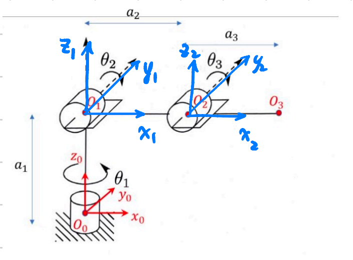
(4) Rotate by α about the world z-axis

$$R_4 = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Only (3) is wrt local frame,
So, the total rotation matrix is :

$$R_{total} = R_4 R_2 R_1 R_3$$

Problem 6



a. $O_0\ (0, 0, 0)$

$O_1\ (0, 0, a_1)$

O_2 : $\{0\} \rightarrow \{1\}$: θ_1 around z_0 , wrt $\{0\}$

$$H_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\{1\} \rightarrow \{2\}$: θ_2 around y_1 , wrt $\{1\}$

$$H_2^1 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & a_2\cos\theta_2 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 & -a_2\sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^0 = H_1^0 \cdot H_2^1$$

$$\overrightarrow{O_0O_2} = H_2^0 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = H_1^0 H_2^1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

O_3 : $\{2\} \rightarrow \{3\}$: θ_3 around y_2 , wrt $\{2\}$

$$H_3^2 = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & a_3\cos\theta_3 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_3 & 0 & \cos\theta_3 & -a_3\sin\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^0 = H_1^0 H_2^1 H_3^2$$

$$\overrightarrow{O_0O_3} = H_3^0 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = H_1^0 H_2^1 H_3^2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Contents

- b.
- b1
- b2
- b3
- b4
- c.
- c1
- c2
- c3
- Function: cal point O
- Plot b
- plot c

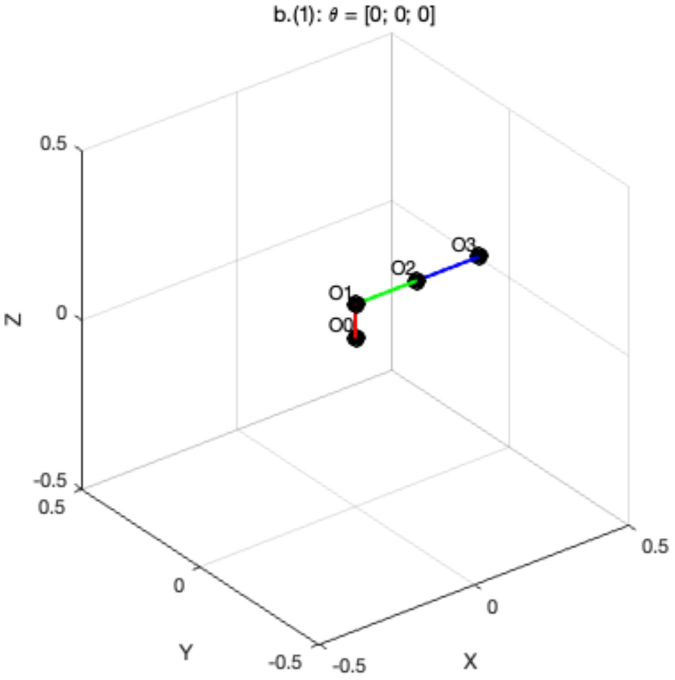
```
clc;
clear;
close all;

% parameters
a1 = 0.1;
a2 = 0.2;
a3 = 0.2;
```

b.

```
theta1 = [0; 0; 0];
theta2 = [0; pi/4; pi/4];
theta3 = [pi/6; pi/4; -pi/2];
theta4 = [pi; pi/2; pi/2];
```

b1

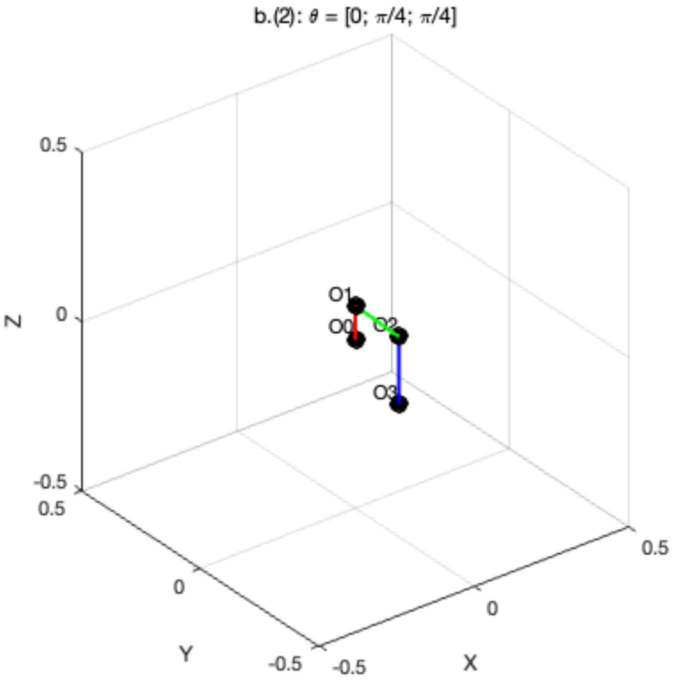
```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta1);
plot_robot_arm(O0, O1, O2, O3);
title('b.(1): \theta = [0; 0; 0]');
```



b.(1): θ = [0; 0; 0]

b2

```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta2);
plot_robot_arm(O0, O1, O2, O3);
title('b.(2): \theta = [0; \pi/4; \pi/4]');
```

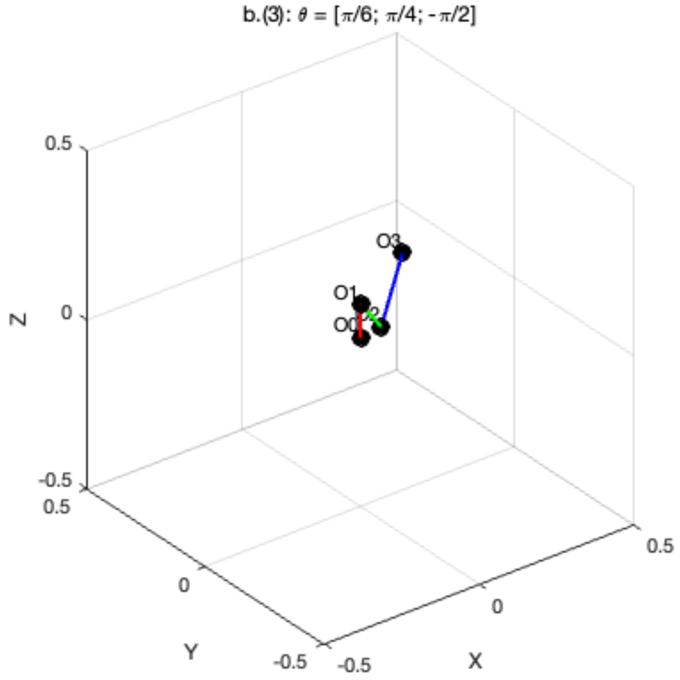


b.(2): θ = [0; π/4; π/4]

b3

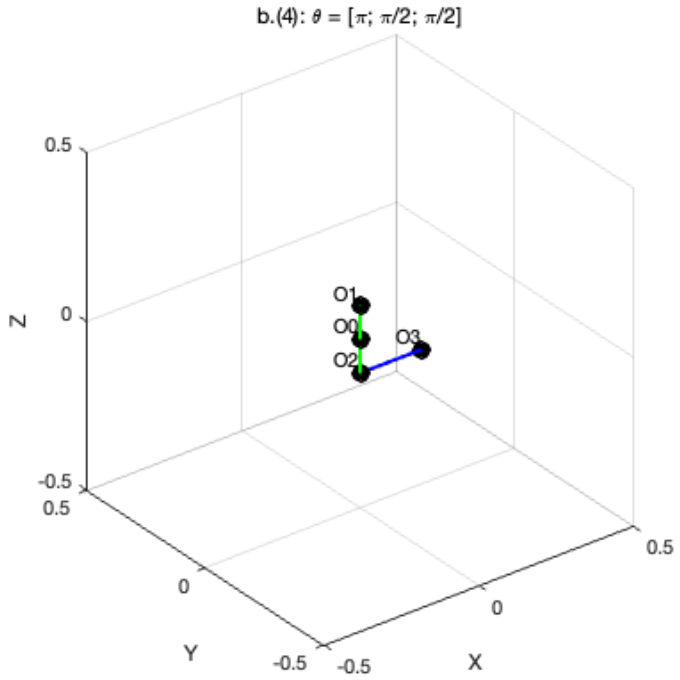
```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta3);
```

```
plot_robot_arm(O0, O1, O2, O3);
title('b.(3): \theta = [\pi/6; \pi/4; -\pi/2]');
```



b4

```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta4);
plot_robot_arm(O0, O1, O2, O3);
title('b.(4): \theta = [\pi; \pi/2; \pi/2]');
```



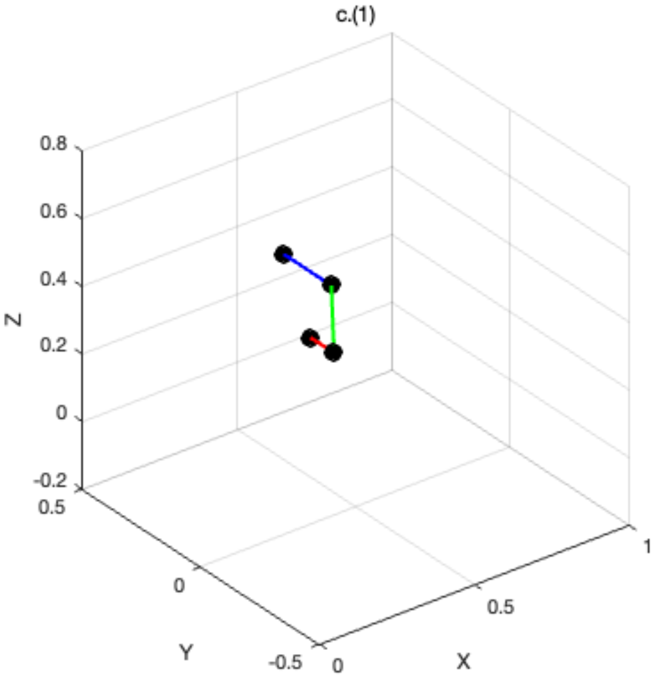
c.

arm

```
theta_c = [0; pi/4; pi/4];

% Rigid Motion Matrixs
Trans = [
    1, 0, 0, 0.5;
    0, 1, 0, 0;
    0, 0, 1, 0;
    0, 0, 0, 1
];

Roll = [
    1, 0, 0, 0;
    0, 0, -1, 0;
    0, 1, 0, 0;
    0, 0, 0, 1
];

Yaw = [
    cos(pi/4), -sin(pi/4), 0, 0;
    sin(pi/4), cos(pi/4), 0, 0;
    0, 0, 1, 0;
    0, 0, 0, 1
];

% c1
H1 = Roll * Yaw * Trans;

% c2
H2 = Trans * Roll * Yaw;

% c3
H3 = Roll * Yaw * Trans;
```

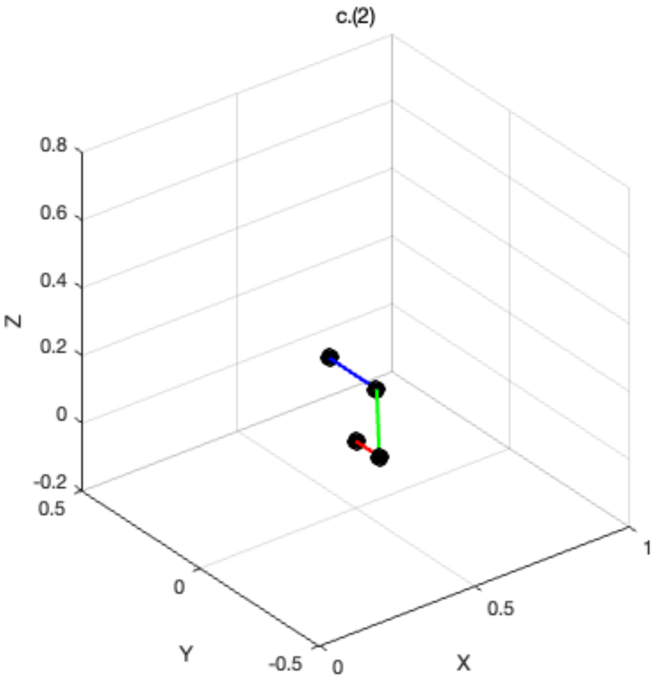
c1

```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta_c);
plot_robot_arm_with_base(O0, O1, O2, O3, H1);
title('c.(1)');
```



c2

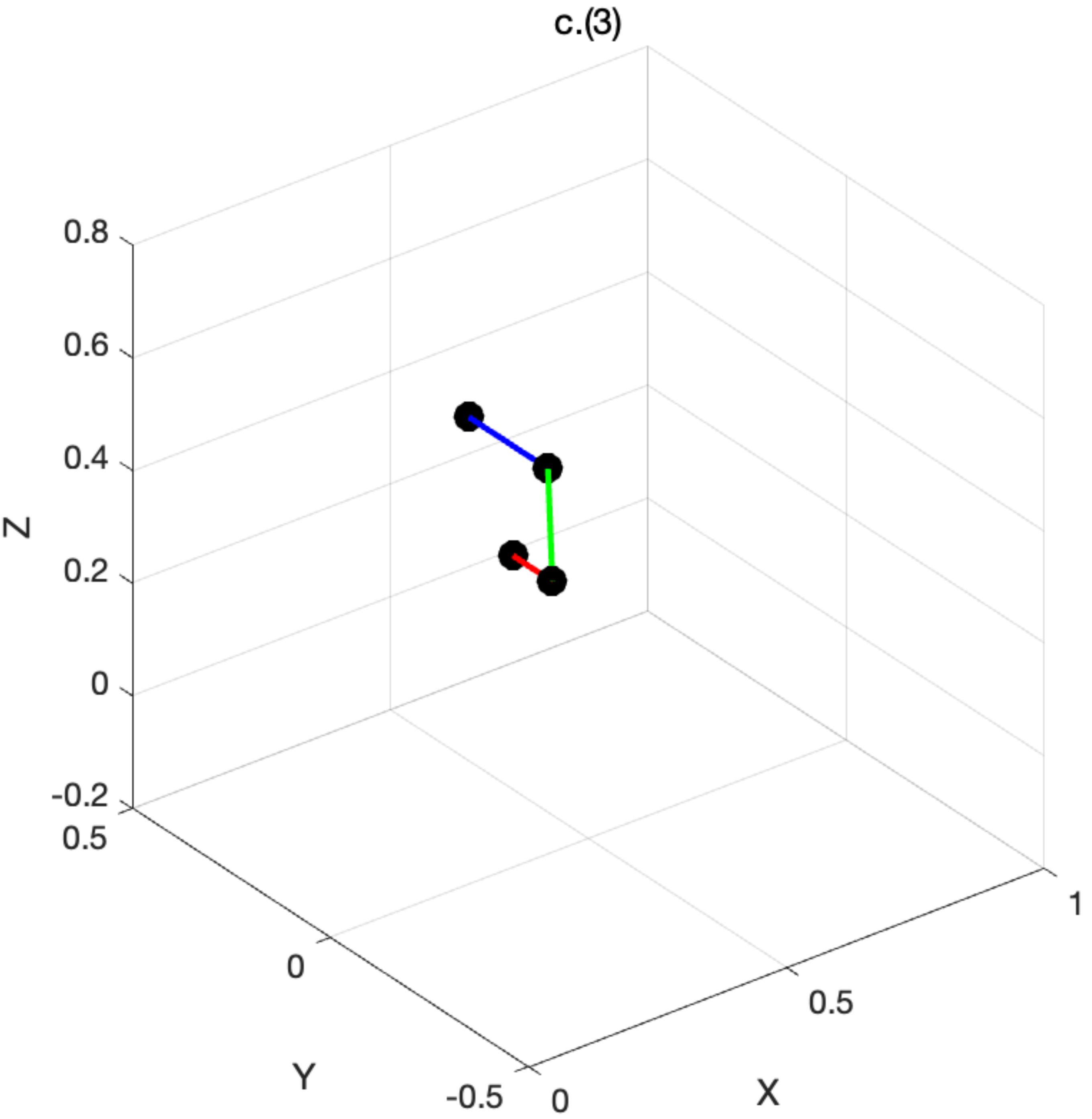
```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta_c);
plot_robot_arm_with_base(O0, O1, O2, O3, H2);
title('c.(2)');
```



c3

```
figure;
[O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta_c);
```

```
plot_robot_arm_with_base(O0, O1, O2, O3, H3);
title('c.(3)');
```



Function: cal point O

```
function [O0, O1, O2, O3] = calculate_joint_positions(a1, a2, a3, theta)
    theta1 = theta(1);
    theta2 = theta(2);
    theta3 = theta(3);

    % Matrix
    T1 = [
        cos(theta1), -sin(theta1), 0, 0;
        sin(theta1), cos(theta1),  0, 0;
        0,           0,            1, a1;
        0,           0,            0, 1
    ];
    T2 = [
        cos(theta2), 0, sin(theta2), a2*cos(theta2);
        0,           1, 0,           0;
        -sin(theta2),0, cos(theta2), -a2*sin(theta2);
        0,           0, 0,           1
    ];

    T3 = [
        cos(theta3), 0, sin(theta3), a3*cos(theta3);
        0,           1, 0,           0;
        -sin(theta3),0, cos(theta3), -a3*sin(theta3);
        0,           0, 0,           1
    ];

    % vectors
    O_1 = T1;
    O_2 = T1 * T2;
    O_3 = T1 * T2 * T3;

    % coordinates
    O0 = [0; 0; 0];
    O1 = O_1(1:3, 4);
    O2 = O_2(1:3, 4);
    O3 = O_3(1:3, 4);
end
```

Plot b

```
function plot_robot_arm(O0, O1, O2, O3)

    % lines
    plot3([O0(1), O1(1)], [O0(2), O1(2)], [O0(3), O1(3)], 'r', 'LineWidth', 2); hold on;
    plot3([O1(1), O2(1)], [O1(2), O2(2)], [O1(3), O2(3)], 'g', 'LineWidth', 2);
    plot3([O2(1), O3(1)], [O2(2), O3(2)], [O2(3), O3(3)], 'b', 'LineWidth', 2);

    % joints
    plot3(O0(1), O0(2), O0(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O1(1), O1(2), O1(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O2(1), O2(2), O2(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O3(1), O3(2), O3(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');

    % remarks
    text(O0(1), O0(2), O0(3), 'O0', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
    text(O1(1), O1(2), O1(3), 'O1', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
    text(O2(1), O2(2), O2(3), 'O2', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
    text(O3(1), O3(2), O3(3), 'O3', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');

    xlabel('X'); ylabel('Y'); zlabel('Z');
    axis equal;
    grid on;
    axis([-0.5 0.5 -0.5 0.5 -0.5 0.5]);
    view(3);
end
```

plot c

```
function plot_robot_arm_with_base(O0, O1, O2, O3, T_base)

    % rotated coordinates
    O0_new = T_base * [O0; 1];
    O1_new = T_base * [O1; 1];
    O2_new = T_base * [O2; 1];
    O3_new = T_base * [O3; 1];
    plot3([O0_new(1), O1_new(1)], [O0_new(2), O1_new(2)], [O0_new(3), O1_new(3)], 'r', 'LineWidth', 2); hold on;
    plot3([O1_new(1), O2_new(1)], [O1_new(2), O2_new(2)], [O1_new(3), O2_new(3)], 'g', 'LineWidth', 2);
    plot3([O2_new(1), O3_new(1)], [O2_new(2), O3_new(2)], [O2_new(3), O3_new(3)], 'b', 'LineWidth', 2);

    plot3(O0_new(1), O0_new(2), O0_new(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O1_new(1), O1_new(2), O1_new(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O2_new(1), O2_new(2), O2_new(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
    plot3(O3_new(1), O3_new(2), O3_new(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');

    xlabel('X'); ylabel('Y'); zlabel('Z');
    axis equal;
    grid on;
    axis([0 1 -0.5 0.5 -0.2 0.8]);
    view(3);
end
```

Published with MATLAB® R2023b