INTRODUÇÃO À ROBÓTICA INDUSTRIAL ENGENHARIA DE CONTROLE E AUTOMAÇÃO INSTITUTO FEDERAL DO PARANÁ - CAMPUS JACAREZINHO

Ronald José Contijo
Professor: Dr. João Paulo Lima Silva de Almeida

March 6, 2025

Atividade 1

Exercício 1

Um vetor P_A é rotacionado em θ graus em torno de $\hat{Z}_A$ e, posteriormente, rotacionado em Θ graus em torno de $\hat{X}_A$. Obtenha a matriz de rotação que contemple ambas as rotações, na devida ordem.

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Z}_A$

A matriz de rotação em torno do eixo Z por um ângulo θ é dada por:

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Passo 2: Matriz de Rotação em Torno de $\hat{X}_A$

A matriz de rotação em torno do eixo X por um ângulo Θ é dada por:

$$R_X(\Theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta \\ 0 & \sin\Theta & \cos\Theta \end{bmatrix}$$

Passo 3: Combinação das Rotações

A matriz de rotação final R é obtida pela multiplicação das matrizes na ordem inversa:

$$R = R_X(\Theta) \cdot R_Z(\theta)$$

Substituindo as matrizes, obtém-se:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta \\ 0 & \sin\Theta & \cos\Theta \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicando as matrizes, resulta em:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \cos\Theta\sin\theta & \cos\Theta\cos\theta & -\sin\Theta \\ \sin\Theta\sin\theta & \sin\Theta\cos\theta & \cos\Theta \end{bmatrix}$$

Resolução no MATLAB

A resolução no MATLAB foi implementada utilizando a função `trplot` para visualizar a rotação. O código utilizado é apresentado abaixo:

```
% Angulos de rotação (em graus)
theta = 45; % Rotação em Z
Theta = 30; % Rotação em X

% Conversão para radianos
theta_rad = deg2rad(theta);
Theta_rad = deg2rad(Theta);

% Matriz de rotação em Z
Rz = [cos(theta_rad) -sin(theta_rad) 0;
       sin(theta_rad)  cos(theta_rad) 0;
       0               0              1];

% Matriz de rotação em X
Rx = [1      0              0;
      0      cos(Theta_rad) -sin(Theta_rad);
      0      sin(Theta_rad)  cos(Theta_rad)];

% Matriz de rotação final
R = Rx * Rz;

% Exibindo a matriz de rotação
disp('Matriz de Rotação Final:');
disp(R);
% Visualização da rotação
figure;
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 1); % Frame original
hold on;
trplot([R [0; 0; 0]; 0 0 0 1], 'frame', 'B', 'color', 'r', 'length', 1); % Frame rotacionado
title('Rotação em Z e depois em X');
xlabel('X');
ylabel('Y');
zlabel('Z');
grid on;
axis equal;
```

Listing 1: Código MATLAB para rotação em Z e X.

Visualização do Gráfico

A figura abaixo mostra o frame original (em preto) e o frame após as rotações em Z e X (em vermelho).

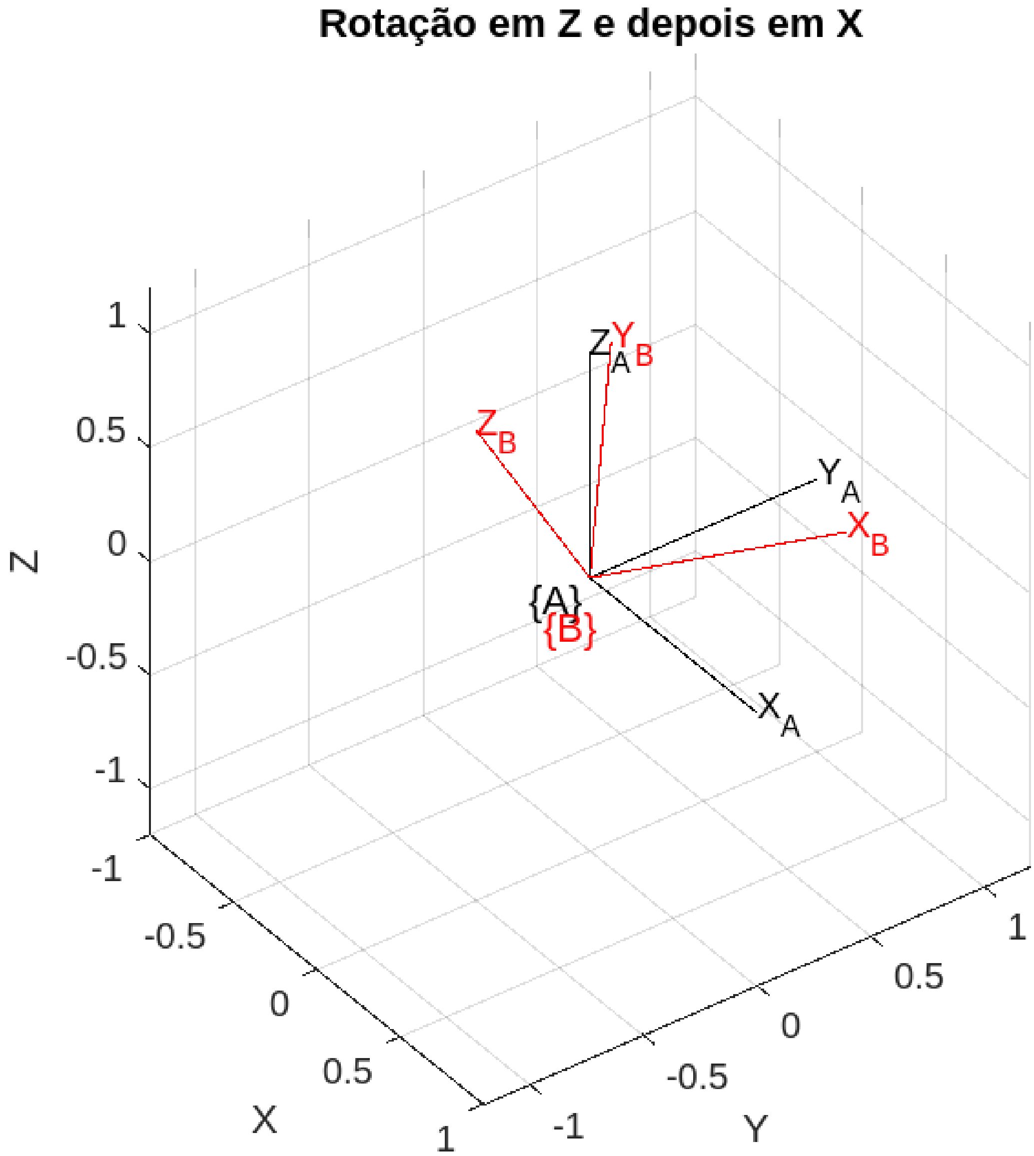

Figure 1: Visualização da rotação em Z e X.

Exercício 2

Um vetor P_A é rotacionado em $30°$ em torno de $\hat{Y}_A$ e, posteriormente, rotacionado em $45°$ em torno de $\hat{X}_A$. Obtenha a matriz de rotação que contemple ambas as rotações, na devida ordem.

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Y}_A$

A matriz de rotação em torno do eixo Y por um ângulo $\theta = 30°$ é dada por:

$$R_Y(30°) = \begin{bmatrix} \cos 30° & 0 & \sin 30° \\ 0 & 1 & 0 \\ -\sin 30° & 0 & \cos 30° \end{bmatrix}$$

Passo 2: Matriz de Rotação em Torno de $\hat{X}_A$

A matriz de rotação em torno do eixo X por um ângulo $\Theta = 45°$ é dada por:

$$R_X(45°) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & -\sin 45° \\ 0 & \sin 45° & \cos 45° \end{bmatrix}$$

Passo 3: Multiplicação das Matrizes

A matriz de rotação final R é obtida pela multiplicação na ordem dada:

$$R = R_X(45°) \cdot R_Y(30°)$$

Substituindo as matrizes:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & -\sin 45° \\ 0 & \sin 45° & \cos 45° \end{bmatrix} \cdot \begin{bmatrix} \cos 30° & 0 & \sin 30° \\ 0 & 1 & 0 \\ -\sin 30° & 0 & \cos 30° \end{bmatrix}$$

Calculando a multiplicação:

$$R = \begin{bmatrix} \cos 30° & 0 & \sin 30° \\ \sin 45° \sin 30° & \cos 45° & -\sin 45° \cos 30° \\ -\cos 45° \sin 30° & \sin 45° & \cos 45° \cos 30° \end{bmatrix}$$

Resolução no MATLAB

A resolução no MATLAB foi implementada utilizando a função `trplot` para visualizar a rotação. O código utilizado é apresentado abaixo:

```
% Angulos de rotação (em graus)
theta = 30; % Rotação em Y
Theta = 45; % Rotação em X

% Conversão para radianos
theta_rad = deg2rad(theta);
Theta_rad = deg2rad(Theta);

% Matriz de rotação em Y
```

```
Ry = [cos(theta_rad) 0 sin(theta_rad);
      0 1 0;
      -sin(theta_rad) 0 cos(theta_rad)];

% Matriz de rotação em X
Rx = [1 0 0;
      0 cos(Theta_rad) -sin(Theta_rad);
      0 sin(Theta_rad) cos(Theta_rad)];

% Matriz de rotação final
R = Rx * Ry;

% Exibindo a matriz de rotação
disp('Matriz de Rotação Final:');
disp(R);

% Visualização da rotação
figure;
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 1); % Frame original
hold on;
trplot([R [0; 0; 0]; 0 0 0 1], 'frame', 'B', 'color', 'r', 'length', 1); % Frame rotacionado
title('Rotação em Y e depois em X');
xlabel('X');
ylabel('Y');
zlabel('Z');
grid on;
axis equal;
```

Listing 2: Código MATLAB para rotação em Y e X.

Visualização do Gráfico

A Figura 2 mostra o frame original (em preto) e o frame após as rotações em Y e X (em vermelho).

Também foram modeladas algumas setinhas no software Blender para verificar se o resultado está correto. O frame original A está de branco e o frame modificado B está com o Z em azul, o Y em vermelho e o X em verde, como está ilustrado na figura 3.

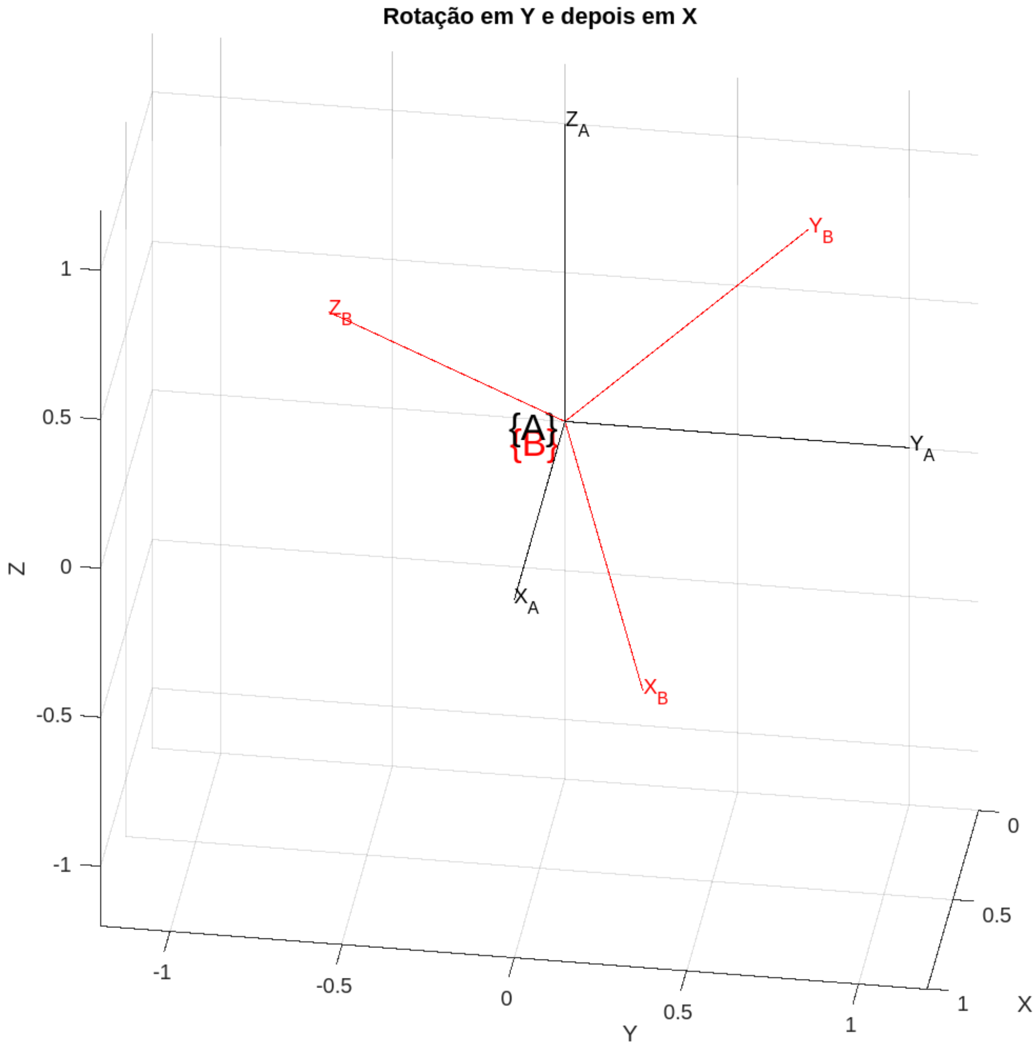

Figure 2: Visualização da rotação em Y e X.

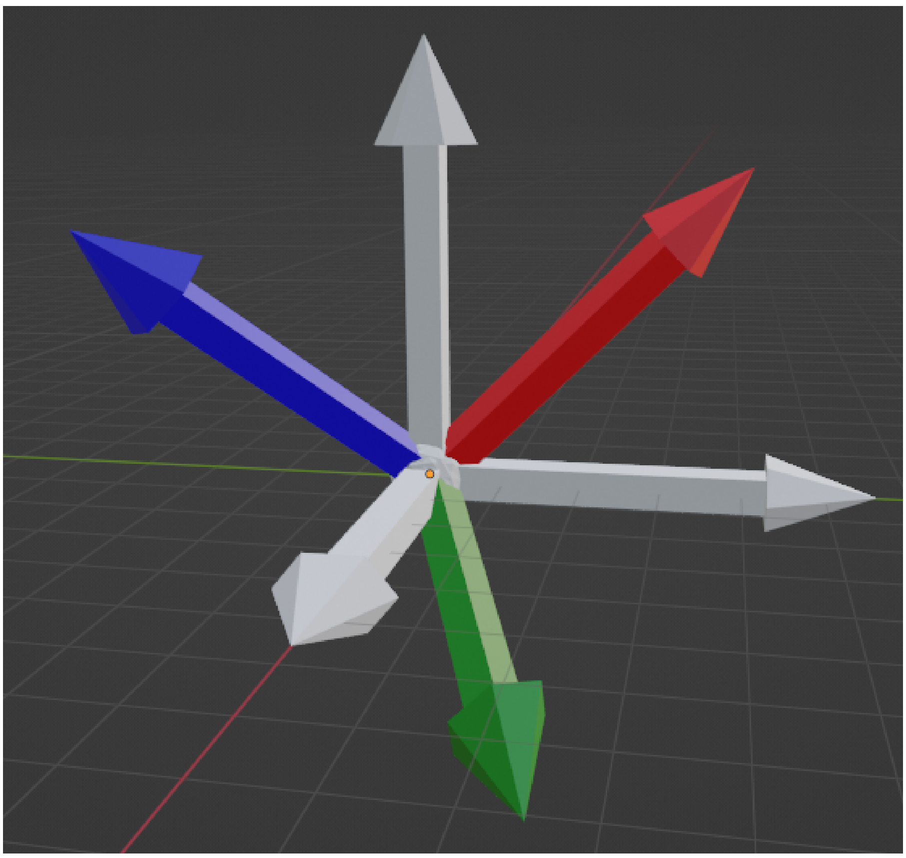

Figure 3: Visualização da rotação em Y e X no Blender.

Exercício 3

Com o uso do MATLAB, retome o Exercício 1 do conteúdo 03 e plote os pontos P_B, P_A e a origem em um gráfico tridimensional.

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Z}$

A rotação do sistema de coordenadas B em relação ao sistema de coordenadas A ocorre em 30° em torno do eixo Z. A matriz de rotação $R_Z(30°)$ é:

$$R_Z(30^\circ) = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ & 0 \\ \sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Passo 2: Transformação de P_B para P_A

Sabemos que o vetor P_B no sistema B é dado por:

$$P_B = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

Para encontrar P_A, Aplicam-se a transformação:

$$P_A = R_Z(30^\circ) \cdot P_B$$

Substituindo os valores:

$$P_A = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ & 0 \\ \sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

Calculando:

$$P_A = \begin{bmatrix} -2\sin 30^\circ \\ 2\cos 30^\circ \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1.732 \\ 0 \end{bmatrix}$$

Resolução no MATLAB

A seguir, é apresentado o código MATLAB que realiza a rotação e plota os pontos P_A, P_B e a origem.

```
%  ngulo  de rotação (graus)
theta = 30;

% Conversão para radianos
theta_rad = deg2rad(theta);

% Matriz de rotação em Z
Rz = [cos(theta_rad) -sin(theta_rad) 0;
      sin(theta_rad) cos(theta_rad) 0;
      0 0 1];

```

```matlab
% Ponto no sistema {B}
PB = [0; 2; 0];

% Transformação para o sistema {A}
PA = Rz * PB;

% Exibição dos resultados
disp('Coordenadas de P_A:');
disp(PA);

% Plotagem com trplot
figure;
hold on;
grid on;
axis equal;
xlabel('X'); ylabel('Y'); zlabel('Z');
view(3); % Garante que a visualização seja 3D
rotate3d on; % Permite interatividade no gráfico 3D

% Frame original {A}
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 2);

% Frame transformado {B} usando Rz
T_B = [Rz [0;0;0]; 0 0 0 1];
trplot(T_B, 'frame', 'B', 'color', 'b', 'length', 2);

% Plotando os pontos P_B, P_A
plot3(PA(1), PA(2), PA(3), 'bo', 'MarkerSize', 8, 'MarkerFaceColor',
    'b');
text(PA(1), PA(2), PA(3), '  P_B', 'Color', 'blue', 'FontSize', 12);

plot3(PA(1), PA(2), PA(3), 'ro', 'MarkerSize', 8, 'MarkerFaceColor',
    'r');
text(PA(1), PA(2), PA(3), '  P_A', 'Color', 'red', 'FontSize', 12);


% Linhas para representar as conexões entre pontos
plot3([0 PA(1)], [0 PA(2)], [0 PA(3)], 'b--');
plot3([0 PA(1)], [0 PA(2)], [0 PA(3)], 'r--');

% Título e legendas
title('Transformação de Coordenadas: P_B para P_A no Espaço 3D');
legend('Frame A', 'Frame B', 'P_B (Sistema B)', 'P_A (Sistema A)',
    'P_C (Ponto Extra)');
```

Listing 3: Código MATLAB para cálculo e plotagem de P_A, P_B e origem.

Visualização do Gráfico

A Figura 4 exibe o ponto P_B no sistema B, o ponto transformado P_A no sistema A e a origem.

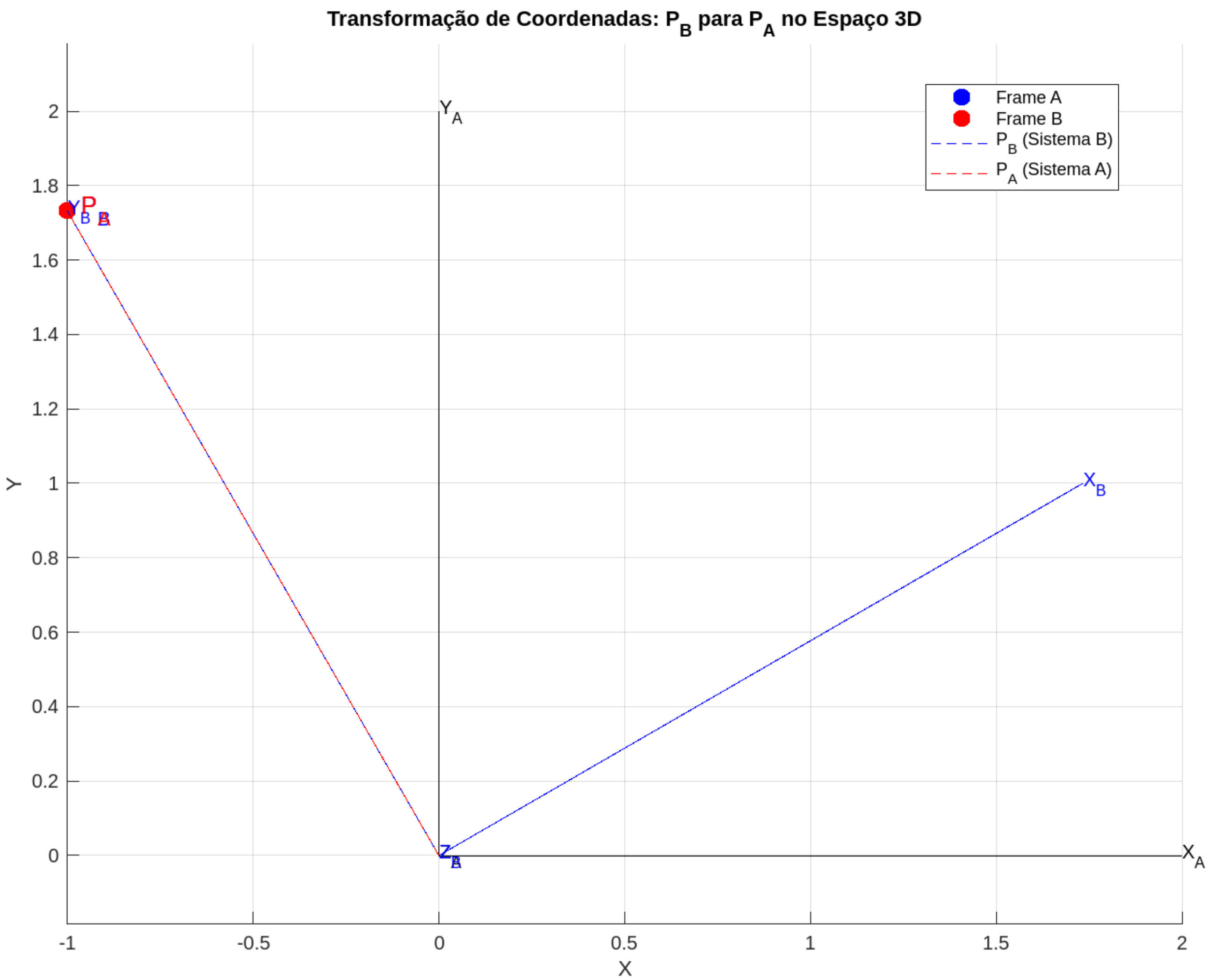

Figure 4: Plotagem de P_B, P_A e origem.

Exercício 4

Com o uso do MATLAB, retome o Exercício 2 do conteúdo 03 e plote os pontos P_B, P_A, $P_{B_{ORG}}$ e a origem em um gráfico tridimensional.

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Z}$

A rotação do sistema de coordenadas B em relação ao sistema de coordenadas A ocorre em 30° em torno do eixo Z. A matriz de rotação $R_Z(30°)$ é:

$$R_Z(30°) = \begin{bmatrix} \cos 30° & -\sin 30° & 0 \\ \sin 30° & \cos 30° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Passo 2: Aplicação da Translação

O sistema de coordenadas B está deslocado 10 unidades em X_A e 5 unidades em Y_A. A matriz de translação T é:

$$T = \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix}$$

Passo 3: Transformação de P_B para P_A

Sabemos que o vetor P_B no sistema B é dado por:

$$P_B = \begin{bmatrix} 3 \\ 7 \\ 0 \end{bmatrix}$$

A conversão para o sistema A é feita por:

$$P_A = R_Z(30°) \cdot P_B + T$$

Substituindo os valores:

$$P_A = \begin{bmatrix} \cos 30° & -\sin 30° & 0 \\ \sin 30° & \cos 30° & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 7 \\ 0 \end{bmatrix} + \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix}$$

Após o cálculo:

$$P_A = \begin{bmatrix} 3\cos 30° - 7\sin 30° + 10 \\ 3\sin 30° + 7\cos 30° + 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 3(0.866) - 7(0.5) + 10 \\ 3(0.5) + 7(0.866) + 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.598 - 3.5 + 10 \\ 1.5 + 6.062 + 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.0981 \\ 12.5622 \\ 0 \end{bmatrix}$$

Resolução no MATLAB

O código abaixo calcula os pontos e exibe a visualização em 3D com ‘trplot‘.

```matlab
% Angulo de rotação (graus)
theta = 30;

% Conversão para radianos
theta_rad = deg2rad(theta);

% Matriz de rotação em Z
Rz = [cos(theta_rad) -sin(theta_rad) 0;
      sin(theta_rad) cos(theta_rad) 0;
      0 0 1];

% Translação
T = [10; 5; 0];

% Ponto no sistema {B}
PB = [3; 7; 0];

% Transformação para o sistema {A}
PA = Rz * PB + T;


% Ponto de origem do sistema {B} antes da translação
PB_ORG = Rz * [0;0;0] + T;

% Exibição dos resultados
disp('Coordenadas de P_A:');
disp(PA);
disp('Coordenadas de P_B_ORG:');
disp(PB_ORG);

% Plotagem com trplot
figure;
hold on;
grid on;
axis equal;
xlabel('X'); ylabel('Y'); zlabel('Z');
view(3);
rotate3d on;

% Frame original {A}
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 2);

% Frame transformado {B} (incluindo rotação e translação)
T_B = [Rz T; 0 0 0 1];
trplot(T_B, 'frame', 'B', 'color', 'b', 'length', 2);

% Plotando os pontos P_B, P_A, P_B_ORG e Origem

PB = PA% Apesar de um pouco duvidoso isso funciona

plot3(PB(1), PB(2), PB(3), 'bo', 'MarkerSize', 8, 'MarkerFaceColor',
    'b');

```

```
text(PB(1), PB(2), PB(3), '  P_B', 'Color', 'blue', 'FontSize', 12);

plot3(PA(1), PA(2), PA(3), 'ro', 'MarkerSize', 8, 'MarkerFaceColor',
    'r');
text(PA(1), PA(2), PA(3), '  P_A', 'Color', 'red', 'FontSize', 12);

plot3(PB_ORG(1), PB_ORG(2), PB_ORG(3), 'go', 'MarkerSize', 8,
    'MarkerFaceColor', 'g');
text(PB_ORG(1), PB_ORG(2), PB_ORG(3), '  P_{B_{ORG}}', 'Color',
    'green', 'FontSize', 12);

plot3(0, 0, 0, 'ko', 'MarkerSize', 8, 'MarkerFaceColor', 'k');
text(0, 0, 0, '  Origem', 'Color', 'black', 'FontSize', 12);

% Linhas para representar as conexões entre pontos
plot3([PB_ORG(1) PB(1)], [PB_ORG(2) PB(2)], [PB_ORG(3) PB(3)], 'b--');
plot3([0 PA(1)], [0 PA(2)], [0 PA(3)], 'r--');
plot3([0 PB_ORG(1)], [0 PB_ORG(2)], [0 PB_ORG(3)], 'g--');

% Título e legendas
title('Transformação de Coordenadas: P_B para P_A no Espaço 3D');
legend('Frame A', 'Frame B', 'P_B (Sistema B)', 'P_A (Sistema A)',
    'P_{B_{ORG}} (Origem B)');
```

Listing 4: Código MATLAB para cálculo e plotagem de P_A, P_B, $P_{B_{ORG}}$ e origem.

Visualização do Gráfico

A figura abaixo exibe o ponto P_B, o ponto transformado P_A, o ponto de origem do sistema B antes da translação $P_{B_{ORG}}$, e a origem.

Para a visualização também foram realizadas as transformações no software Blender. O ponto foi adotado como a cabeça de uma macaca.

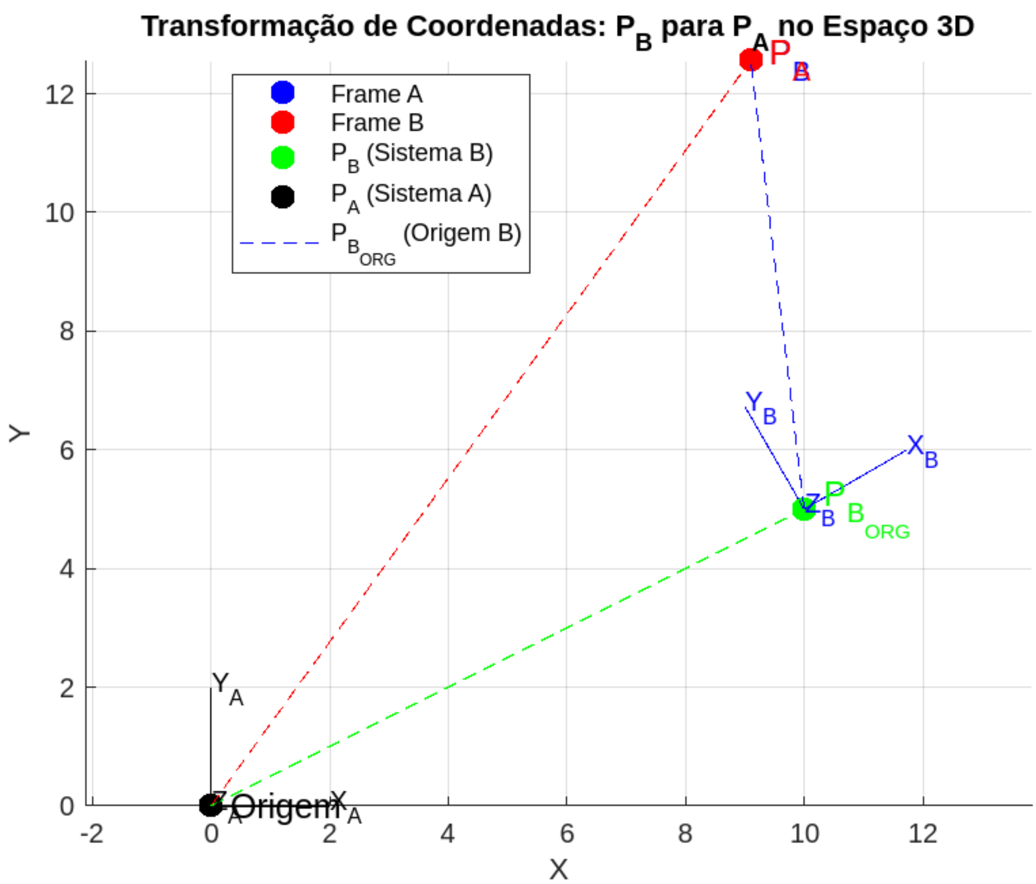

Figure 5: Plotagem de P_B, P_A, $P_{B_{ORG}}$ e origem.

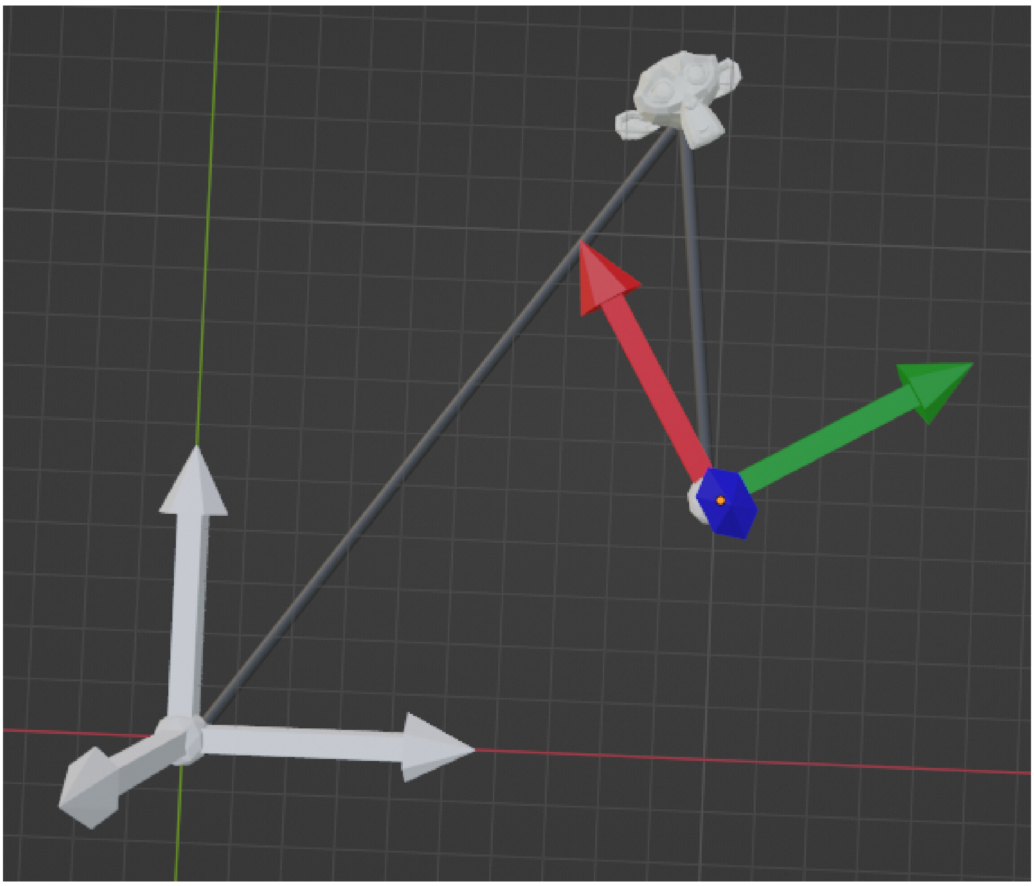

Figure 6: Visualização da rotação em Y e X no Blender.

Exercício 5

Considere o vetor $P_A^1 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$ mostrado na Figura 1 e realize as seguintes transformações:

- (a) Plote P_A^1 em um gráfico tridimensional no MATLAB.
- (b) Rotacione P_A^1 em 45° em torno de $\hat{Z}$ e plote o novo vetor obtido (P_A^2).
- (c) Rotacione P_A^1 em $-30°$ em torno de $\hat{Y}$ e plote o novo vetor obtido (P_A^3).
- (d) Rotacione P_A^1 em 90° em torno de $\hat{X}$ e plote o novo vetor obtido (P_A^4).

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Z}$

A matriz de rotação em torno do eixo Z por um ângulo θ é:

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Aplicando a rotação de 45°:

$$P_A^2 = R_Z(45°) P_A^1$$

Passo 2: Matriz de Rotação em Torno de $\hat{Y}$

A matriz de rotação em torno do eixo Y por um ângulo θ é:

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Aplicando a rotação de $-30°$:

$$P_A^3 = R_Y(-30°) P_A^1$$

Passo 3: Matriz de Rotação em Torno de $\hat{X}$

A matriz de rotação em torno do eixo X por um ângulo θ é:

$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Aplicando a rotação de 90°:

$$P_A^4 = R_X(90°) P_A^1$$

Resolução no MATLAB

O seguinte código MATLAB implementa as rotações e exibe os vetores transformados em um gráfico tridimensional.

```matlab
% Definição do vetor original
PA1 = [0; 2; 0];

% Angulos de rotação (graus)
thetaZ = 45; % Rotação em Z
thetaY = -30; % Rotação em Y
thetaX = 90; % Rotação em X

% Conversão para radianos
thetaZ_rad = deg2rad(thetaZ);
thetaY_rad = deg2rad(thetaY);
thetaX_rad = deg2rad(thetaX);

% Matrizes de rotação
Rz = [cos(thetaZ_rad) -sin(thetaZ_rad) 0;
      sin(thetaZ_rad) cos(thetaZ_rad) 0;
      0 0 1];

Ry = [cos(thetaY_rad) 0 sin(thetaY_rad);
      0 1 0;
      -sin(thetaY_rad) 0 cos(thetaY_rad)];

Rx = [1 0 0;
      0 cos(thetaX_rad) -sin(thetaX_rad);
      0 sin(thetaX_rad) cos(thetaX_rad)];

% Aplicando as rotações
PA2 = Rz * PA1; % Rotação em Z
PA3 = Ry * PA1; % Rotação em Y
PA4 = Rx * PA1; % Rotação em X

% Exibição dos resultados
disp('P_A^2:'); disp(PA2);
disp('P_A^3:'); disp(PA3);
disp('P_A^4:'); disp(PA4);

% Plotagem dos vetores
figure;
hold on;
grid on;
axis equal;
view(3);
rotate3d on;

xlabel('X'); ylabel('Y'); zlabel('Z');

% Frame de referência
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 1);

```

```
% Plotando os vetores
quiver3(0, 0, 0, PA1(1), PA1(2), PA1(3), 'r', 'LineWidth', 2);
quiver3(0, 0, 0, PA2(1), PA2(2), PA2(3), 'b', 'LineWidth', 2);
quiver3(0, 0, 0, PA3(1), PA3(2), PA3(3), 'g', 'LineWidth', 2);
quiver3(0, 0, 0, PA4(1), PA4(2), PA4(3), 'm', 'LineWidth', 2);

% Legenda
legend('P_A^1 (Original)', 'P_A^2 (Rotação Z)', 'P_A^3 (Rotação Y)',
    'P_A^4 (Rotação X)');
title('Transformações do Vetor P_A^1');
```

Listing 5: Código MATLAB para rotação e visualização de P_A^1, P_A^2, P_A^3 e P_A^4.

Visualização do Gráfico

A figura abaixo mostra a visualização tridimensional dos vetores transformados.

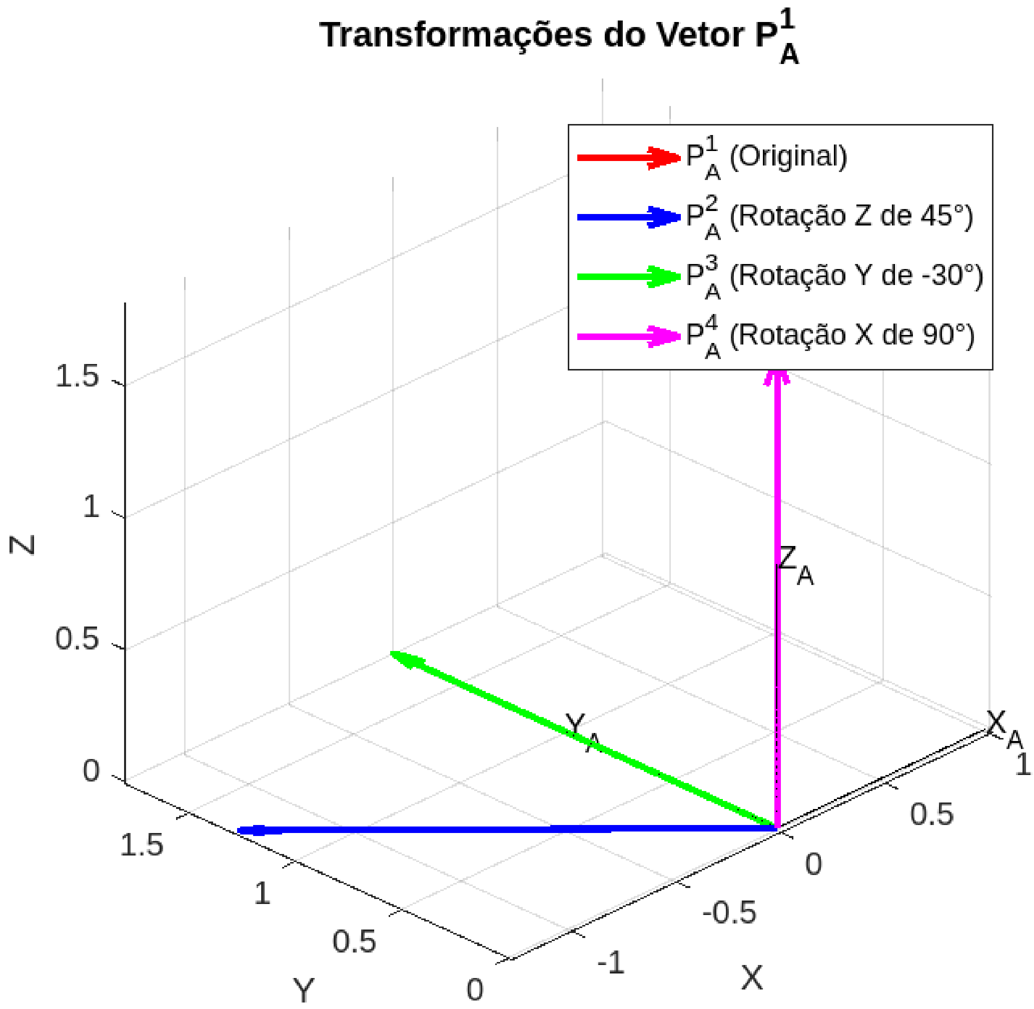

Figure 7: Plotagem dos vetores $P_A^1, P_A^2, P_A^3, P_A^4$ no MATLAB.

Exercício 6

Dado um vetor em um determinado sistema de referência, considere três rotações deste vetor, na seguinte ordem:

1. (a) 30° em torno de $\hat{Z}$;
2. (b) 45° em torno de $\hat{Y}$;
3. (c) 60° em torno de $\hat{X}$.

Verifique se a ordem em que as rotações ocorrem influencia a determinação do vetor rotacionado, ou seja, verifique se:

$$R_Z(30^\circ)R_Y(45^\circ)R_X(60^\circ) = R_X(60^\circ)R_Y(45^\circ)R_Z(30^\circ)$$

Elabore um gráfico tridimensional para ilustrar a análise.

Resolução Matemática

As matrizes de rotação utilizadas são:

Matriz de Rotação em Torno de $\hat{Z}$

$$R_Z(30^\circ) = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ & 0 \\ \sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matriz de Rotação em Torno de $\hat{Y}$

$$R_Y(45^\circ) = \begin{bmatrix} \cos 45^\circ & 0 & \sin 45^\circ \\ 0 & 1 & 0 \\ -\sin 45^\circ & 0 & \cos 45^\circ \end{bmatrix}$$

Matriz de Rotação em Torno de $\hat{X}$

$$R_X(60^\circ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 0 & \sin 60^\circ & \cos 60^\circ \end{bmatrix}$$

Cálculo das Matrizes Compostas

Aplicam-se as rotações na ordem dada e na ordem inversa para comparar os resultados.

$$P_{\text{final1}} = R_Z(30^\circ)R_Y(45^\circ)R_X(60^\circ)P$$

$$P_{\text{final2}} = R_X(60^\circ)R_Y(45^\circ)R_Z(30^\circ)P$$

Os resultados obtidos são:

$$P_{\text{final1}} = \begin{bmatrix} 3.0619 \\ 1.7678 \\ -3.5355 \end{bmatrix}$$

$$P_{\text{final2}} = \begin{bmatrix} 3.0619 \\ 3.9017 \\ 0.6341 \end{bmatrix}$$

Como pode-se ver, $P_{\text{final1}} \neq P_{\text{final2}}$, confirmando que a ordem das rotações influencia a transformação final do vetor.

Resolução no MATLAB

O seguinte código MATLAB aplica as rotações, plota cada etapa da transformação e compara os resultados.

```
% Angulos de rotação (graus)
thetaZ = 30;
thetaY = 45;
thetaX = 60;

thetaZ_rad = deg2rad(thetaZ);
thetaY_rad = deg2rad(thetaY);
thetaX_rad = deg2rad(thetaX);

% Matrizes de rotação
Rz = [cos(thetaZ_rad) -sin(thetaZ_rad) 0;
      sin(thetaZ_rad) cos(thetaZ_rad) 0;
      0 0 1];
Ry = [cos(thetaY_rad) 0 sin(thetaY_rad);
      0 1 0;
      -sin(thetaY_rad) 0 cos(thetaY_rad)];
Rx = [1 0 0;
      0 cos(thetaX_rad) -sin(thetaX_rad);
      0 sin(thetaX_rad) cos(thetaX_rad)];

% Vetor inicial conveniente
P = [5; 0; 0];

% Aplicação das rotações na ordem 1
P_final1 = Rz * Ry * Rx * P;

% Aplicação das rotações na ordem inversa
P_final2 = Rx * Ry * Rz * P;

% Comparação
disp('P_final1:'); disp(P_final1);
disp('P_final2:'); disp(P_final2);

% Plotagem
figure;
hold on;
grid on;
axis equal;
view(3);
rotate3d on;
xlabel('X'); ylabel('Y'); zlabel('Z');
```

```
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 1);

% Vetor inicial
quiver3(0, 0, 0, P(1), P(2), P(3), 'k', 'LineWidth', 2);

% Vetores finais
quiver3(0, 0, 0, P_final1(1), P_final1(2), P_final1(3), 'r',
    'LineWidth', 2);
quiver3(0, 0, 0, P_final2(1), P_final2(2), P_final2(3), 'b',
    'LineWidth', 2);

legend('Vetor Inicial', 'P_final1 (Ordem ZYX)', 'P_final2 (Ordem
    XYZ)');
title('Efeito da Ordem das Rotações');
```

Listing 6: Código MATLAB para verificação da ordem das rotações.

Visualização do Gráfico

A figura abaixo exibe a transformação do vetor em cada passo, comparando as ordens ZYX e XYZ.

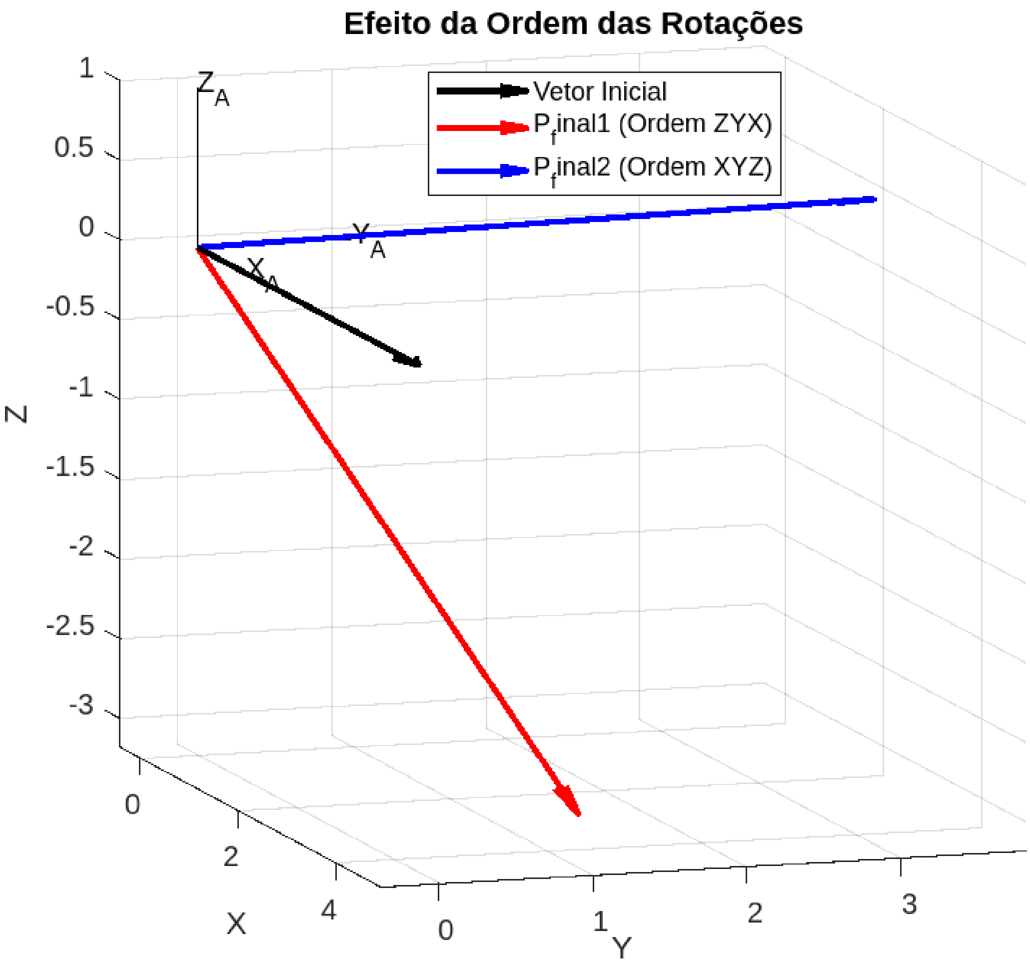

Figure 8: Plotagem das etapas das rotações e comparação dos resultados.

Exercício 7

Dado um sistema de referência $\{B\}$ que é rotacionado 45 graus em torno de $\hat{Z}$ em relação ao sistema de referência $\{A\}$ e transladado em quatro unidades em $\hat{X}_A$ e três unidades em $\hat{Y}_A$, determine BT_A e AT_B e use o MATLAB para plotar tais frames, para ilustrar.

Resolução Matemática

Passo 1: Matriz de Rotação em Torno de $\hat{Z}$

A rotação do sistema de coordenadas B em relação ao sistema de coordenadas A é de 45° em torno do eixo Z. A matriz de rotação $R_Z(45°)$ é:

$$R_Z(45^\circ) = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & -0.7071 & 0 \\ 0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Passo 2: Matriz de Transformação BT_A

A transformação homogênea BT_A é composta pela matriz de rotação $R_Z(45°)$ e pelo vetor de translação $(4, 3, 0)$:

$$BT_A = \begin{bmatrix} 0.7071 & -0.7071 & 0 & 4 \\ 0.7071 & 0.7071 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Passo 3: Matriz de Transformação AT_B

A matriz AT_B é o inverso de BT_A, ou seja:

$$AT_B = (BT_A)^{-1}$$

Para calcular AT_B, utilizamos a propriedade:

$$AT_B = \begin{bmatrix} R_Z(45^\circ)^T & -R_Z(45^\circ)^T \cdot T \\ 0 & 1 \end{bmatrix}$$

$$AT_B = \begin{bmatrix} 0.7071 & 0.7071 & 0 & -4.9497 \\ -0.7071 & 0.7071 & 0 & 0.7071 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Resolução no MATLAB

A seguir, é apresentado o código MATLAB para visualizar os frames de referência A e B.

```
% Angulo de rotação (graus)
theta = 45;

% Conversão para radianos
theta_rad = deg2rad(theta);

% Matriz de rotação em Z
Rz = [cos(theta_rad) -sin(theta_rad) 0;
```

```
      sin(theta_rad) cos(theta_rad) 0;
      0 0 1];

% Vetor de translação
T = [4; 3; 0];

% Matriz de transformação B T_A
B_T_A = [Rz, T; 0 0 0 1];

% Matriz de transformação A T_B (inversa)
A_T_B = inv(B_T_A);

% Exibição das matrizes
disp('Matriz B T_A:');
disp(B_T_A);
disp('Matriz A T_B:');
disp(A_T_B);

figure;
hold on;
grid on;
axis equal;
view(3);
rotate3d on;
xlabel('X'); ylabel('Y'); zlabel('Z');

% Frame original {A}
trplot(eye(4), 'frame', 'A', 'color', 'k', 'length', 1);

% Frame transformado {B}
trplot(B_T_A, 'frame', 'B', 'color', 'r', 'length', 1);

% Extração da posição do frame B a partir de B_T_A
orig_A = [0; 0; 0]; % Origem do sistema A
orig_B = B_T_A(1:3, 4); % Origem do sistema B

% Desenhar uma linha conectando os frames
plot3([orig_A(1), orig_B(1)], [orig_A(2), orig_B(2)], [orig_A(3),
    orig_B(3)], 'k--', 'LineWidth', 2);

% Título
title('Transformação de Coordenadas: Frames {A} e {B}');
```

Listing 7: Código MATLAB para plotagem dos frames A e B.

Visualização do Gráfico

A Figura 9 ilustra os frames de referência A e B após a rotação e translação.

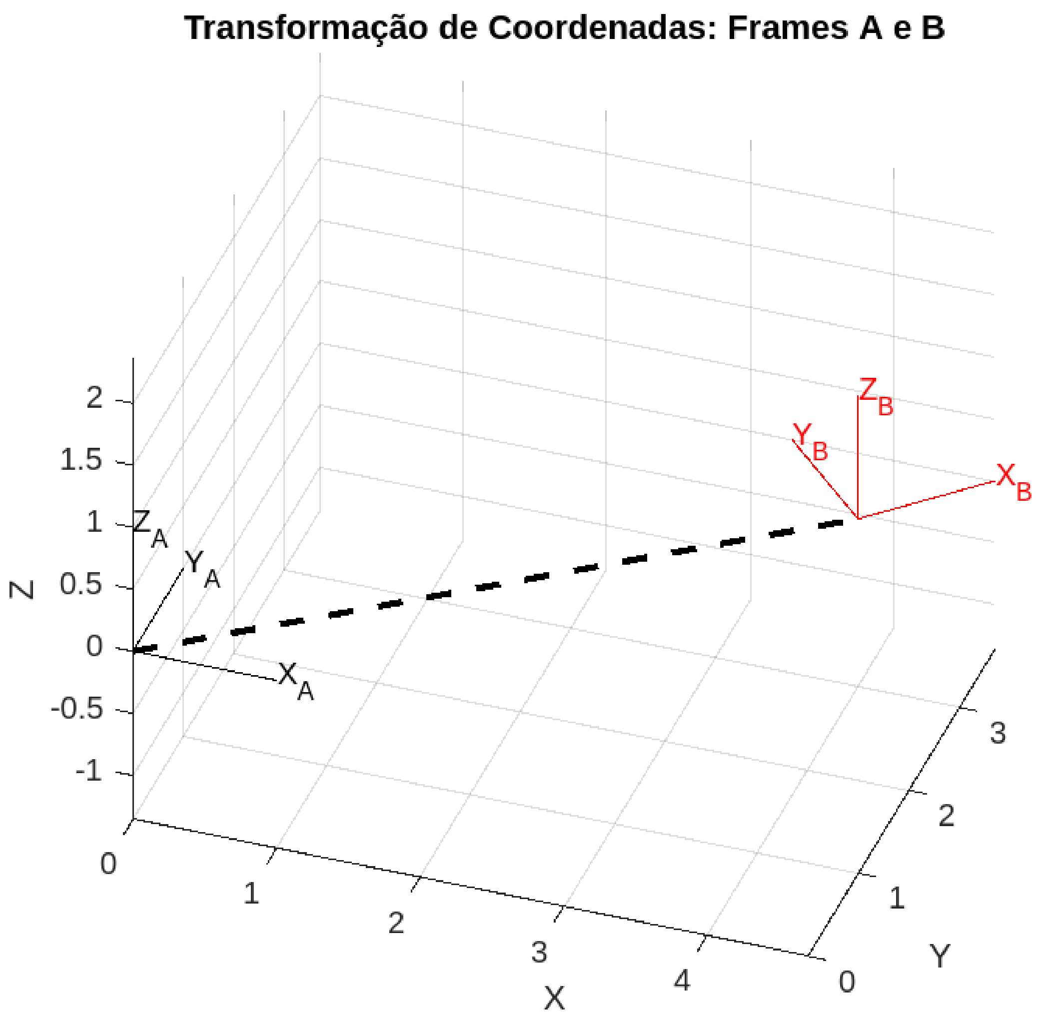

Figure 9: Plotagem dos frames A e B após a transformação.