

Universidade de São Paulo
EESC

SEM0530 - Problemas de Engenharia Mecatrônica II
Prof. Marcelo Areias Trindade

Prática 6 - Transformação de vetores

Aluno: Marcus Vinícius Costa Reis (12549384)

23/07/2022

Contents

1	Problema	3
1.1	Solução	3
2	Screenshots	4
2.1	Rotação - Sequência 1	4
2.2	Rotação - Sequência 2	8
3	Scripts	13
3.1	Explicação	13
3.2	O código	14

1 Problema

- Simular a rotação de um corpo rígido, dado um sequenciamento de rotações pré-definido. Para simular uma rotação executada por juntas, consideramos as rotações sempre em torno de eixos coordenados solidários ao corpo (base móvel)
- Primeira sequência:
 1. Rotação em torno de z de ângulo θ
 2. Rotação em torno de y de ângulo θ
 3. Rotação em torno de x de ângulo 4θ (com velocidade 4 vezes maior)
- Segunda sequência:
 - Executar as três rotações anteriores simultaneamente
- Considerar $\theta = [-(90+N)/4]^\circ$, sendo formado pelos dois últimos algarismos do Número USP do aluno. Nesse caso,

$$N = 84 \implies \theta = -43.5^\circ \approx -0.7592 \text{ rad}$$

1.1 Solução

Para o problema em questão, a solução foi encontrada como resultado de um processo de familiarização com o código original inicialmente disponibilizado.

A compreensão do mesmo possibilitou a elaboração do *script* apresentado na seção Scripts, a partir do qual se obteve as animações necessárias à feitura dos *screenshots*, os quais são apresentados na seção de mesmo nome.

2 Screenshots

2.1 Rotação - Sequência 1

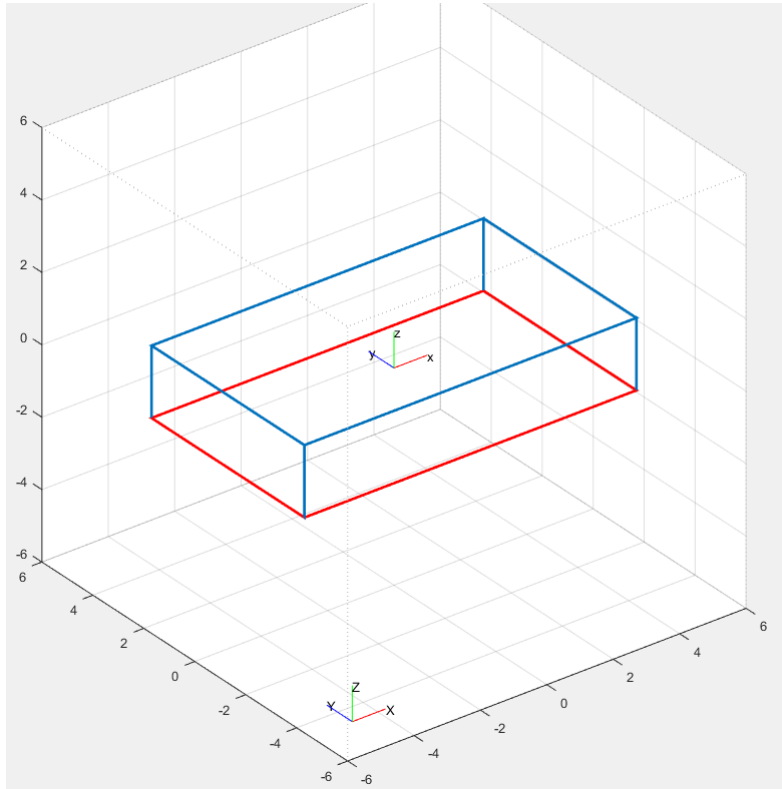
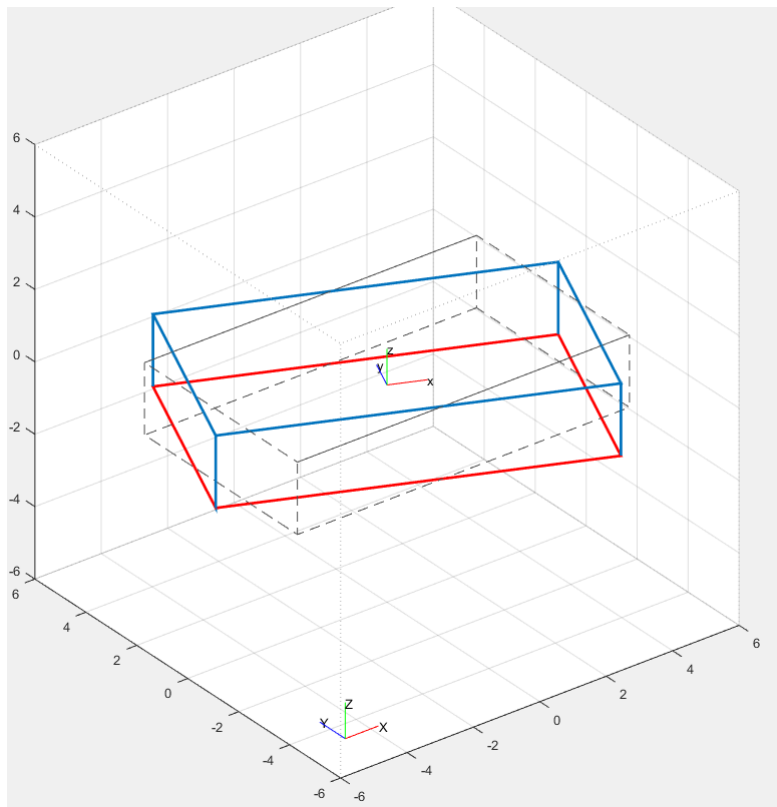
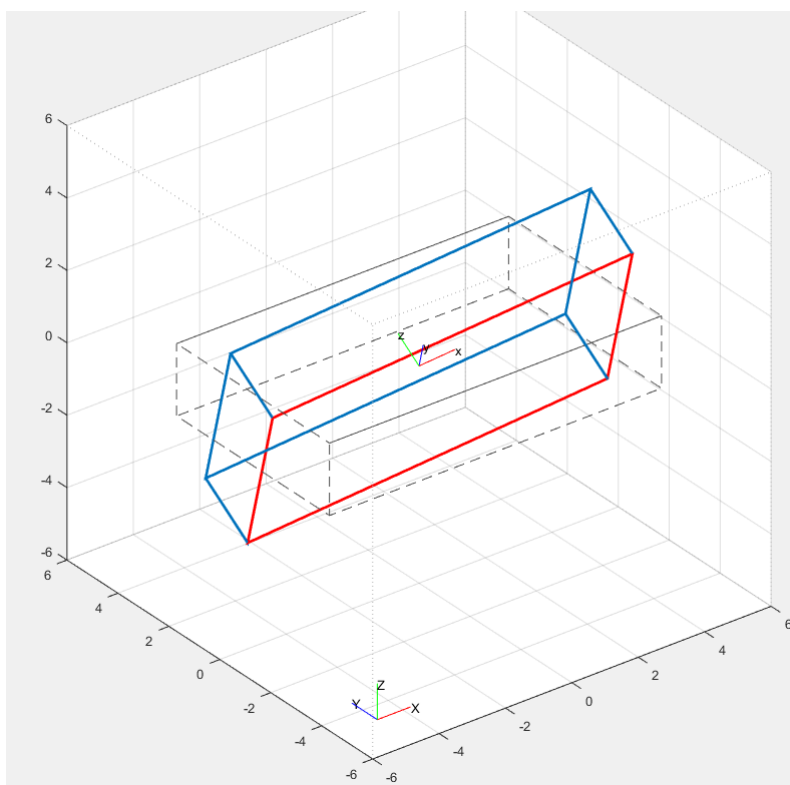
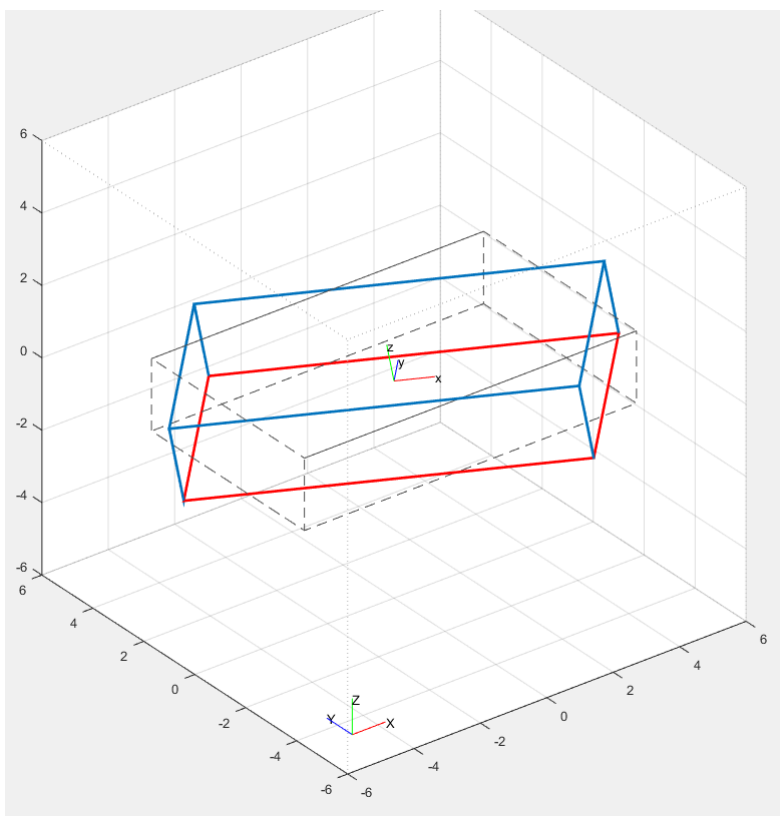
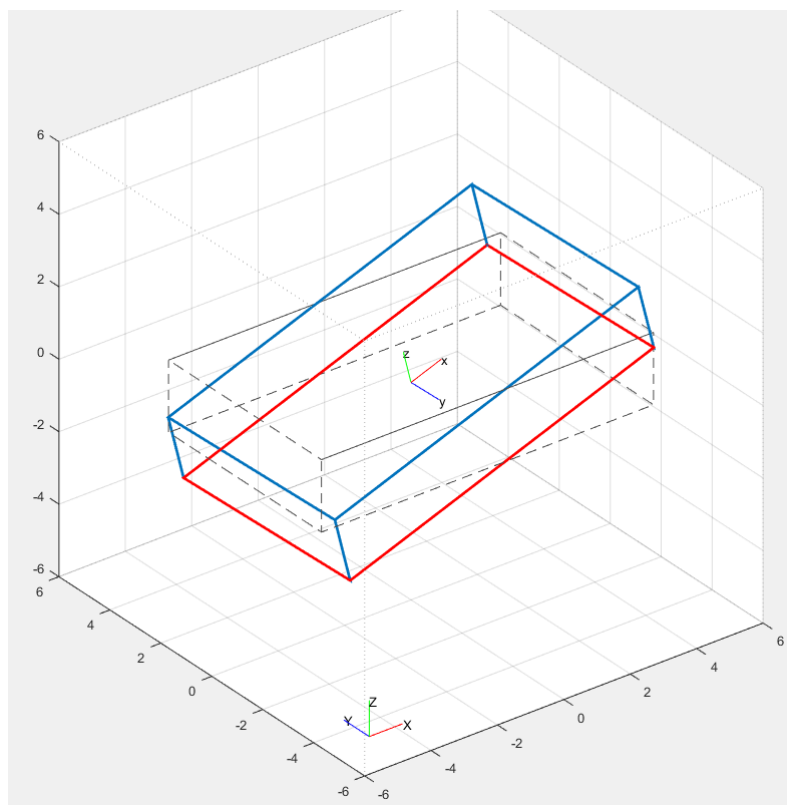
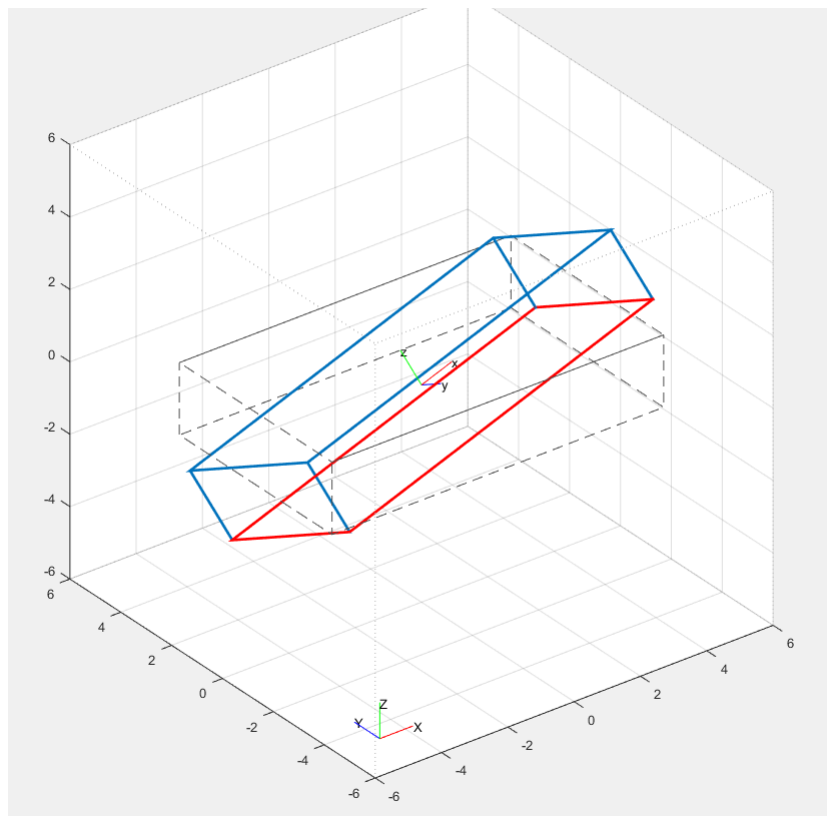


Figure 1: posição inicial







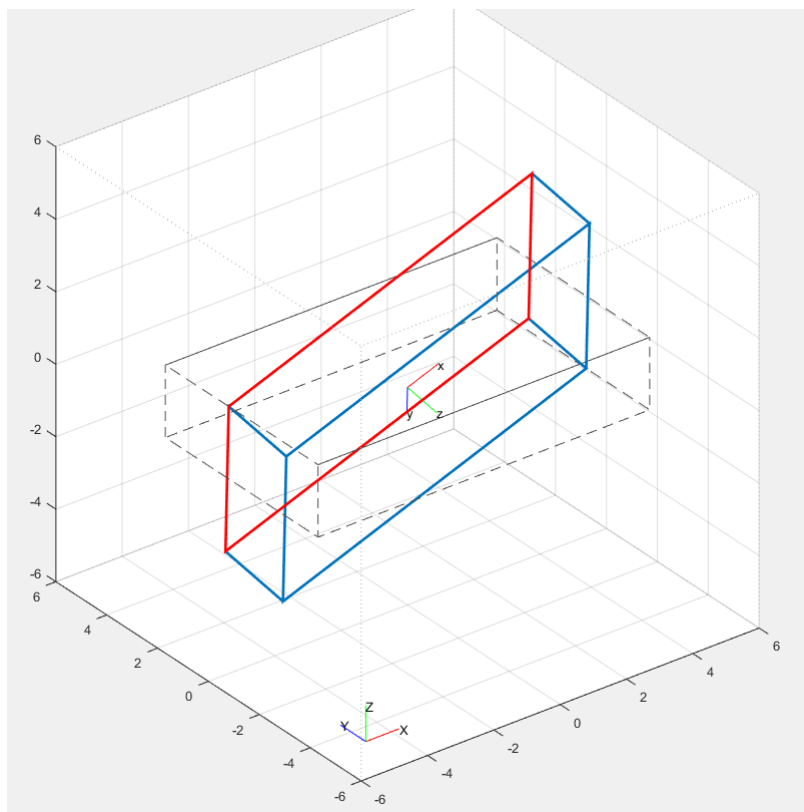
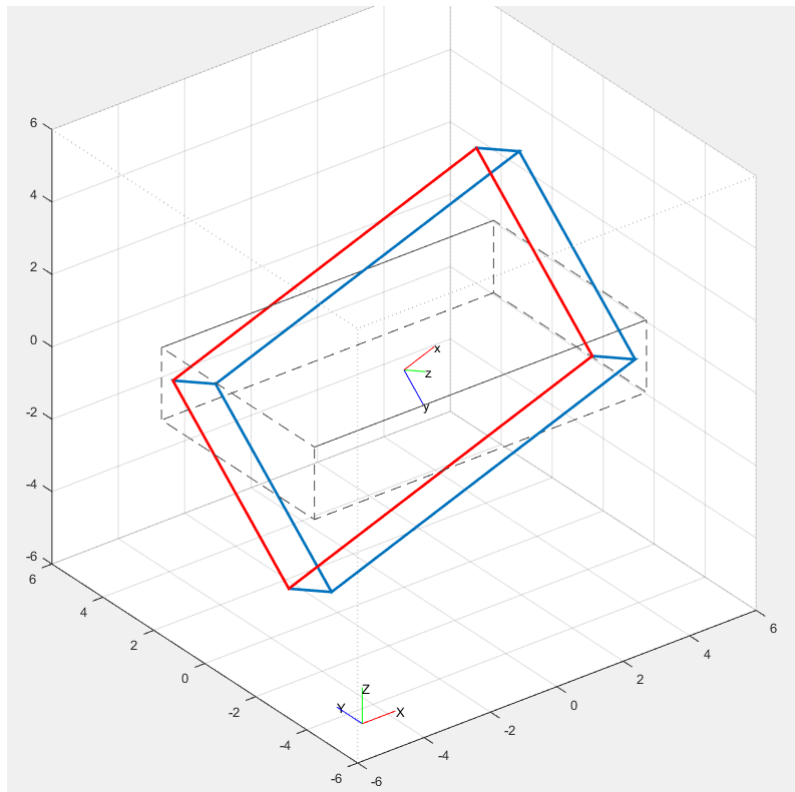


Figure 2: posição final

2.2 Rotação - Sequência 2

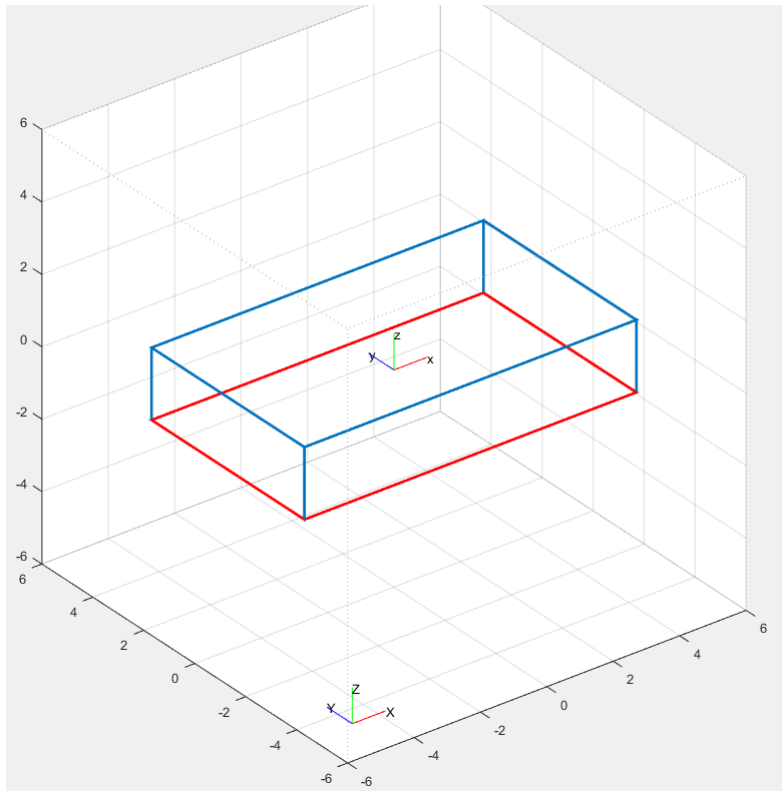
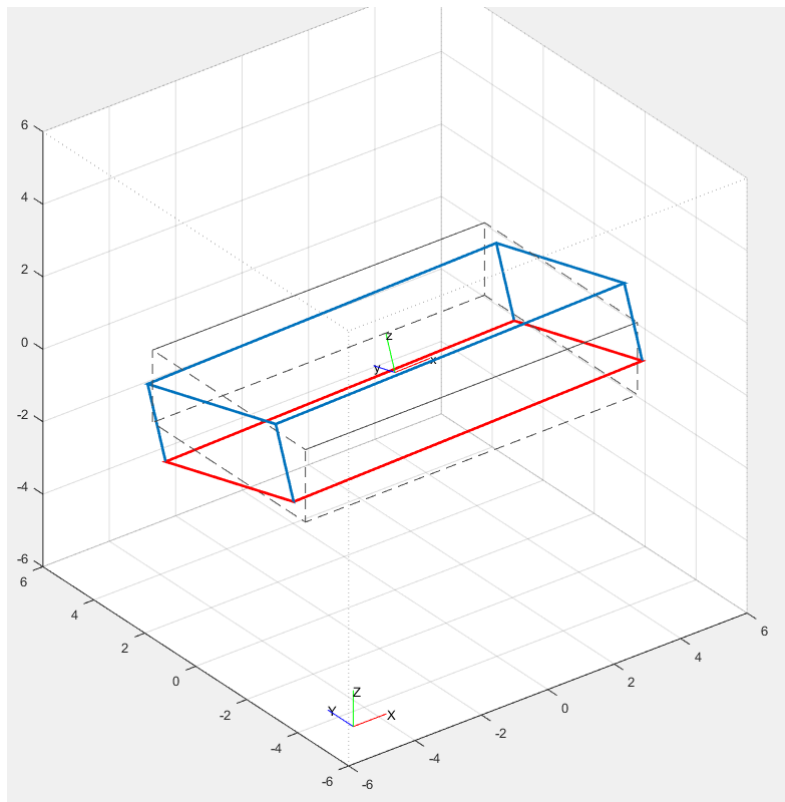
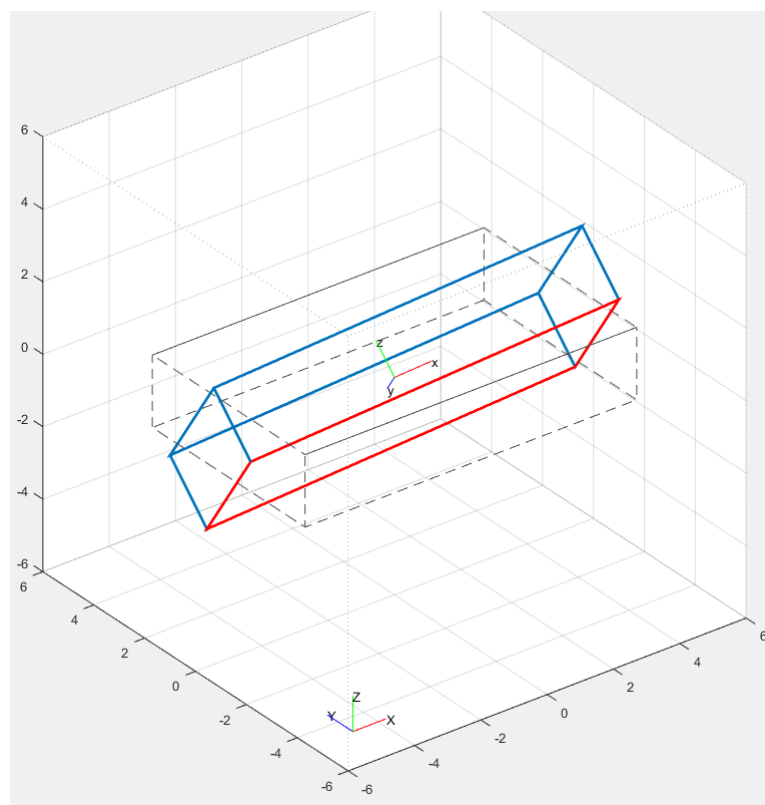
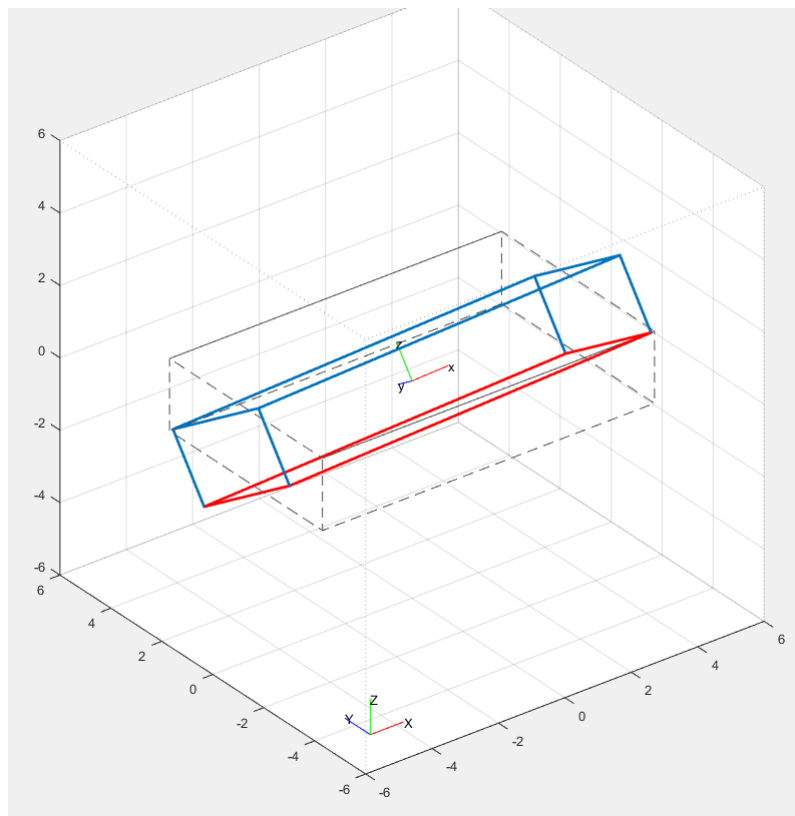
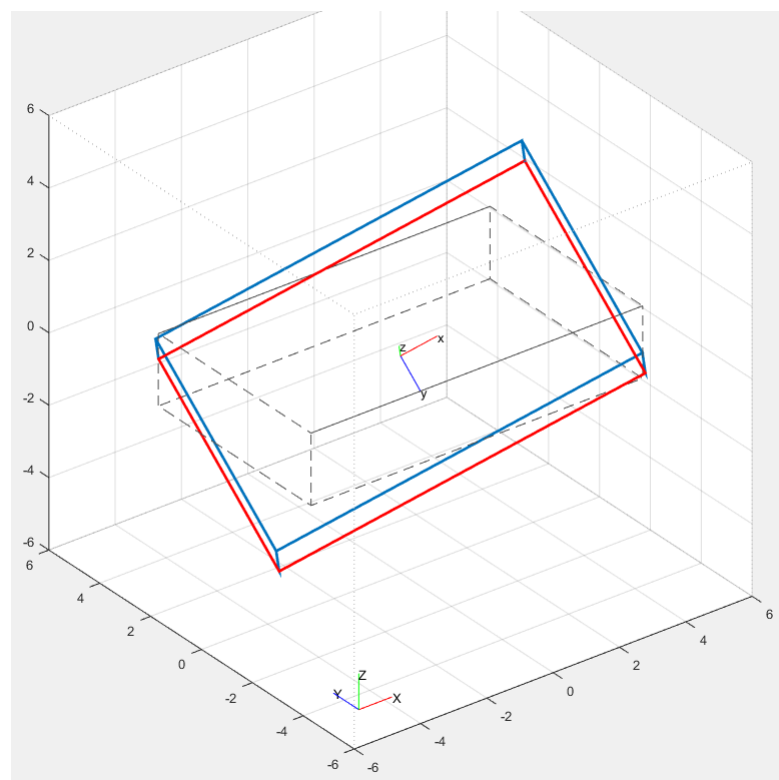
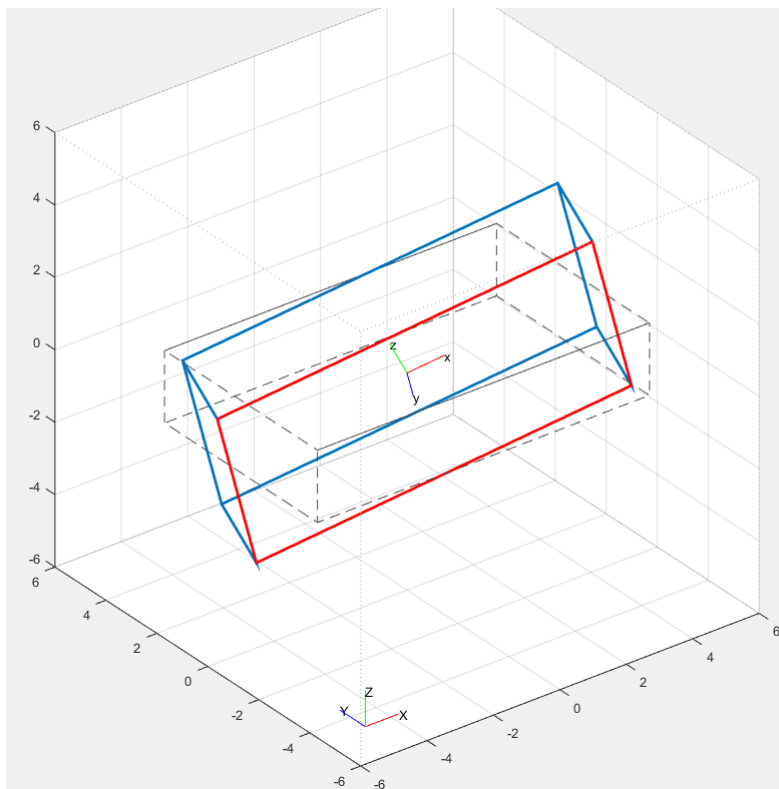
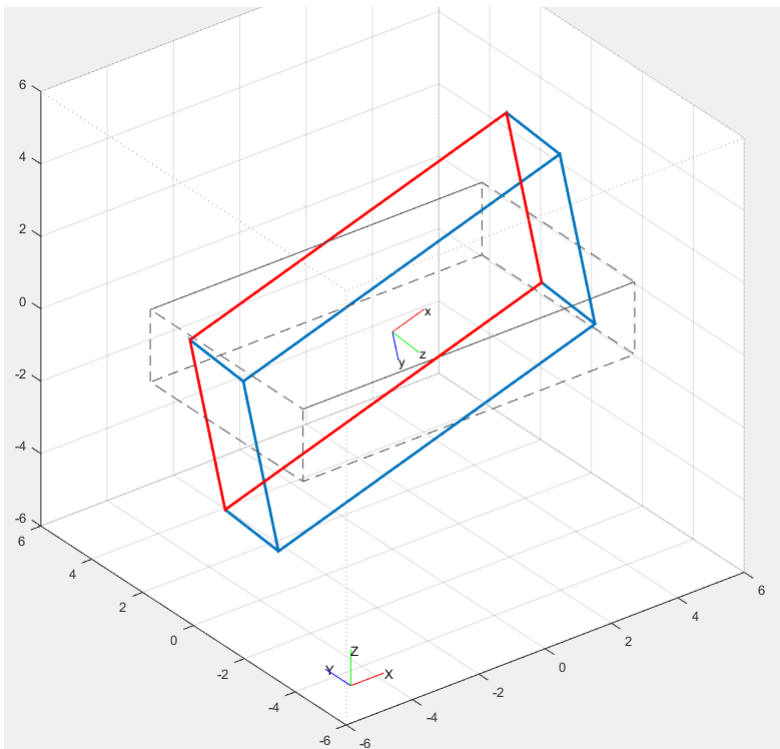
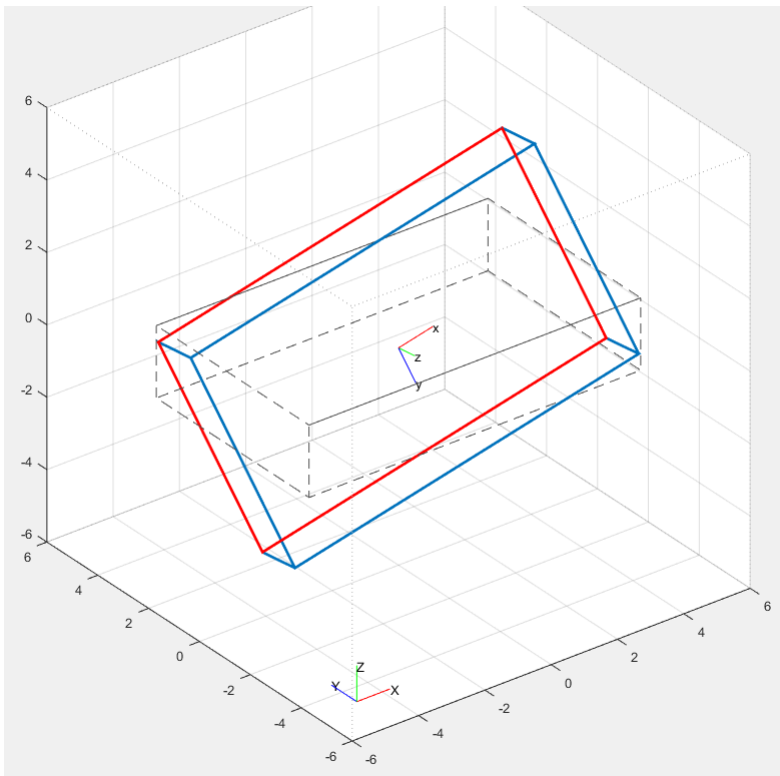


Figure 3: posição inicial









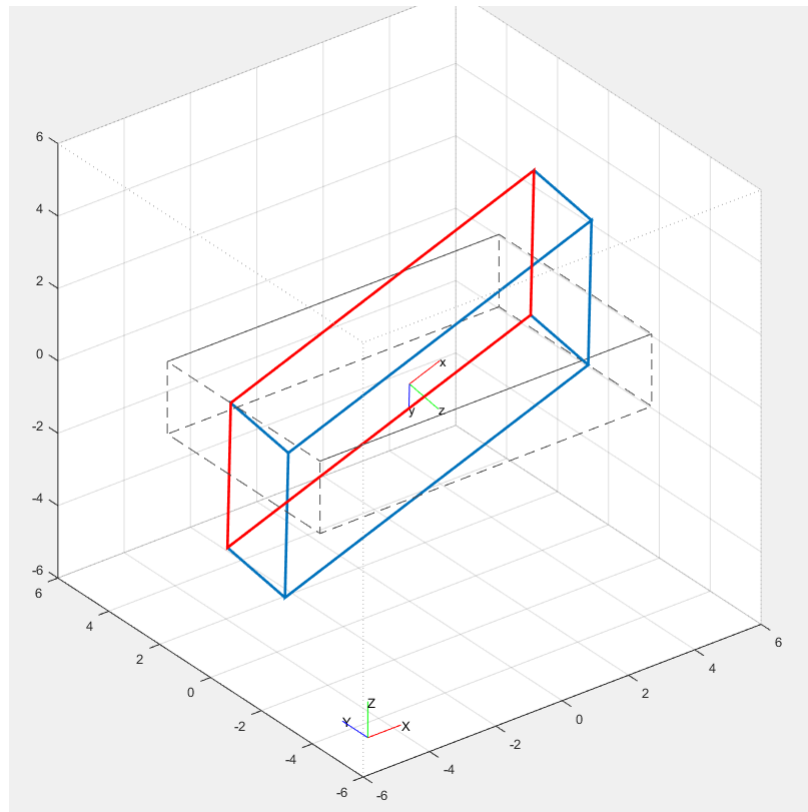


Figure 4: posição final

3 Scripts

3.1 Explicação

A seguir, esclarece-se brevemente o código elaborado para a realização das rotações.

Em linhas gerais, a finalidade da função *p6* é preencher a matriz tridimensional *R*, composta por todas as configurações de rotação compreendidas entre as posições inicial e final do desenho tridimensional proposto. Cada "folha" de *R* é oriunda do produto $R_z R_y R_x$, que indica a ordem de rotação.

Tratando do vetor *theta* que é usado para definir as matrizes *thetaxv*, *thetayv* e *thetazv*, utilizadas, por sua vez, como argumento das funções seno e cosseno presentes nas matrizes de rotação, tem-se sua estrutura básica:

$$theta = [0 : d\Theta : \Theta_f],$$

onde $d\Theta$ é o passo das rotações, determina a quantidade de "folhas" da matriz *R*, além de estar relacionado com a duração da animação obtida, já que o intervalo de tempo decorrido para a execução de cada "folha" é invariável. A seu turno, Θ_f representa o valor de -0.7592 rad .

Definiu-se $d\Theta = -\pi/500$, sendo que, para a primeira sequência utilizou-se $2d\Theta$ como passo, enquanto para a segunda, $0.5d\Theta$. Essa decisão foi tomada convenientemente de maneira a facilitar a feitura dos *screenshots*.

Outrossim, deve-se destacar a presença da expressão $4*theta$ nas matrizes *thetaxv* da primeira e segunda sequências. Não somente o valor de Θ_f será quadruplicado, como também o será $d\Theta$. No entanto, como dito anteriormente, o tempo para a execução de cada "folha" é invariável, de modo que a rotação estará sendo feita com uma velocidade 4 vezes maior, quando se compara o $d\Theta$ efetivo com o original (para cada sequência).

Por fim, a matriz tridimensional *R* é passada como argumento para a função *fcnrot3d*, responsável pelos desenhos (animação). Deve-se, ainda, ressaltar que não se realizou modificação alguma no código desta função.

Fica, desse modo, solucionado o problema proposto.

O código é apresentado a seguir.

3.2 O código

```
1 function p6(seq)
2
3 N = 84; % 2 ultimos Algarismos do numero USP
4 theta_f = (-1)*((90+N)/4)*(pi/180); % angulo de rotacao (rad)
5 dtheta = (-1)*(pi/500); % passo
6
7 if seq == 1
8     % 1a sequencia [z (theta) -> y (theta) -> x (4*theta)]
9     theta = [0:2*dtheta:theta_f];
10    thetaxv = [0*theta 0*theta 4*theta];
11    thetayv = [0*theta theta theta(end)+0*theta];
12    thetazv = [theta theta(end)+0*theta theta(end)+0*theta];
13 else
14    % 2a sequencia [simultaneamente]
15    theta = [0:0.5*dtheta:theta_f];
16    thetaxv = [4*theta];
17    thetayv = [theta];
18    thetazv = [theta];
19 end
20
21 for k = 1:length(thetayv)
22    thetax = thetaxv(k); thetay = thetayv(k); thetaz = thetazv(k);
23    Rx = [1 0 0; 0 cos(thetax) -sin(thetax); 0 sin(thetax) cos(thetax)];
24    Ry = [cos(thetay) 0 sin(thetay); 0 1 0; -sin(thetay) 0 cos(thetay)];
25    Rz = [cos(thetaz) -sin(thetaz) 0; sin(thetaz) cos(thetaz) 0; 0 0 1];
26    R(:, :, k) = Rz*Ry*Rx;
27 end
28
29 fcnrot3d(R)
30
```

Listing 1: script utilizado para a geração das rotações