Robótica Industrial
Aula prática nº 4
Funções iniciais da cinemática direta

Vítor Santos

Universidade de Aveiro

17 Out 2022

Exercício 1 - Transformação de Elos

Criar as seguintes funções elementares para implementar a cinemática direta

- `A=Tlink(`θ`,l,d,`α`)`
  - Transformação associada a um elo. Devolve a matriz respetiva e aceita os 4 parâmetros de D-H
  - Implementa o seguinte: `A=rotz(`θ`)trans(l,0,d)rotx(`α`)`
- `AA=Tlinks(DH)`
  - Aceita uma matriz de **Denavit-Hartenberg (DH)** e devolve uma matriz de transformação para cada elo (linha de DH) dispostas ao longo da 3ª dimensão da **hipermatriz** AA.
  - Invoca a função `Tlink()` anterior.

Exemplo de matriz DH para um robô RR planar

elo i	θ_i	l_i	d_i	α_i
1	θ_1	3	0	0
2	θ_2	1.5	0	0

$$\Longrightarrow DH = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \end{bmatrix}$$

Exercício 2a - Origens dos referenciais de um robô

Criar a função `Org=LinkOrigins(AA)`

- Esta função deve devolver uma matriz com as origens dos diversos sistemas de coordenadas de um manipulador, dada a **hipermatriz** de transformações obtida com `Tlinks()`.
- Aceita uma hipermatriz AA de matrizes de transformação geométrica (uma para cada elo), e devolve as coordenadas das diversas origens começando em $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.
- Sabe-se que: `size(Org,2)=size(AA,3)+1`

Indicação para o cálculo genérico de uma coluna de `Org`

Em notação MATLAB, o valor de `Org(:,i)`, para $i > 1$, obtém-se a partir da 4ª coluna de $A_1 A_2 \ldots A_i = \prod_{k=1}^{i} A_k$, onde A_k (ou também como é conhecida ${}^{k-1}A_k$) é a transformação associada ao elo k. Naturalmente, tem-se que: `Org(:,1)`$=\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.

Exercício 2b - Exemplo de um caso de origens

Se um robô a 3 DOF tiver a seguinte matriz DH para um caso particular de juntas:

```
       %th    l     d   al
DH=[ -pi/4   1     0   0;
       pi/2  1.5   0   0;
      -pi/3  0.5   0   0;
    ];
```

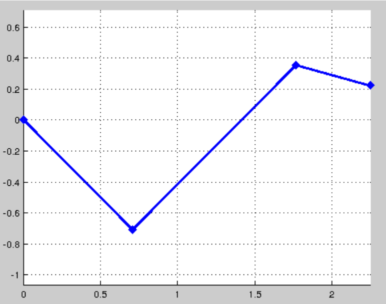
Deverá vir o seguinte para as origens dos diversos elos:

```
Org =      0     0.7071     1.7678     2.2507
           0    -0.7071     0.3536     0.2241
           0          0          0          0
```

Exercício 3 - DrawLinks

Criar a função `h=DrawLinks(Org)`

- A função deve desenhar um gráfico que representa os elos do robô.
- Esse gráfico pode ser uma simples linha poligonal representando os elos. `Org` é uma matriz onde em cada coluna estão os pontos extremos dos elos do robô; `h` é o *handle* gráfico da linha desenhada.
- Para os elementos do exercício anterior, o resultado de `DrawLinks(Org)` é:



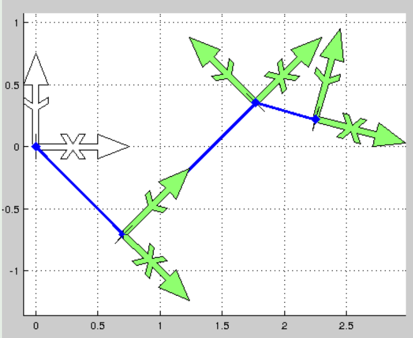
Exercício 4 - DrawFrames

Criar a função `H=DrawFames(AA,P,F)`

- A função deve desenhar gráficos que representam outros objetos associados aos elos, como os referenciais (sistemas de coordenadas).
- Desenha os sistemas de coordenadas todos do robô. O objeto `P`, e as respetivas faces `F`, obtidos por "seixos3.m", ou outro, deverá ser desenhado em cada elo. A hipermatriz `AA` tem as transformações que permitirão calcular as posições dos objetos a desenhar. `H` é um vetor de *handles* gráficos para todos os sistemas desenhados.
- Devem desenhar-se também o primeiro (da base) e o último (mão) sistemas de coordenadas.

Exercício 4 - Exemplo de DrawFrames

- O resultado de `H=DrawFrames(AA,P,F)` para os elementos definidos anteriormente é o ilustrado de seguida:



- NB. O primeiro sistema de coordenadas está na origem, e a sua cor foi diferenciada para mostrar que é a referência.

Exercício 4 - Outras matrizes DH

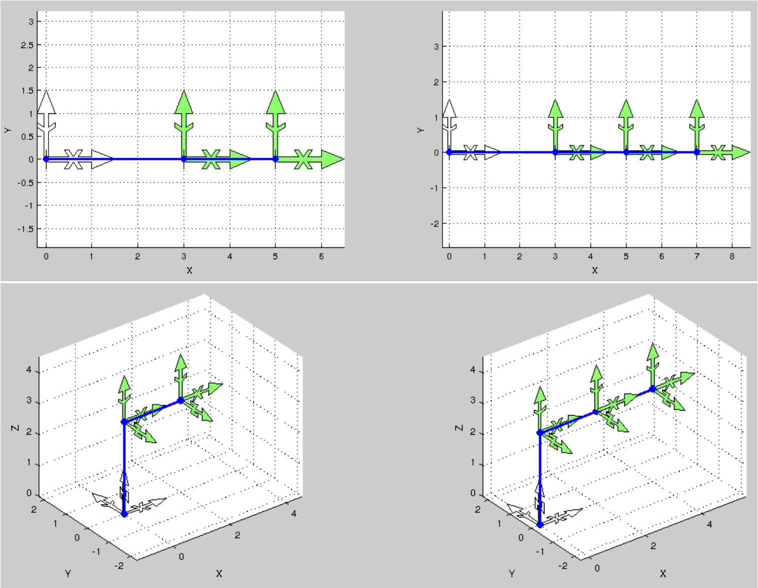
Criar as matrizes DH para os seguintes 4 robôs:

1. RR planar: DH_RR
2. RRR planar: DH_RRR
3. RR a 3D: DH_RR3D
4. RRR antropomórfico: DH_RRA

Representar os 4 robôs na sua posição zero

- Usar também a função `DrawLinks()` anterior.
- Usar os seguintes valores para elos:
  - L1=3
  - L2=2
  - L3=2

Exercício 4 - Resultados de DrawFrames

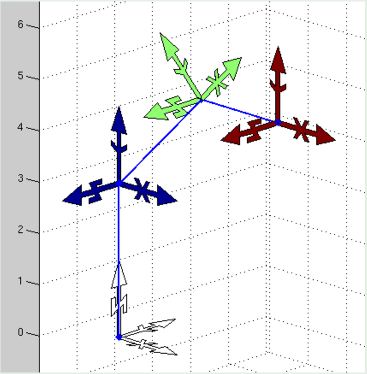


Exercício 5 - O RRR antropomórfico noutra posição

Representar um robô RRR antropomórfico

- Usar as funções anteriores (`DrawLinks` e `DrawFrames`), e representar o robô na posição $\Theta = (0^\circ, +45^\circ, -45^\circ)$
- Sugestões:
  - usar elos L1=3 ; L2=3 ; L3=2
  - Usar um vetor adicional como parâmetro em `DrawFrames` para passar a cor dos diversos referenciais a desenhar.

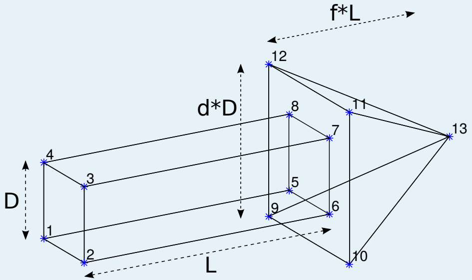
Modelo resultante



Exercício 6 (opcional) - Seta alternativa a 3D

Criar um modelo paramétrico de uma seta a 3D

- Com base nos seguintes parâmetros criar um modelo de um sólido em forma de seta para o eixo dos XX: (valores por defeito)
  - lado da base do corpo: D (0.02)
  - comprimento do corpo: L (1.0)
  - fração entre a base da ponta e D: d (1.5)
  - fração entre o comprimento da ponta e L: f (0.15 ou $d \times D/L$)
- Criar a função: `function [V,F]=seta3Dx(D,L,d,f)`

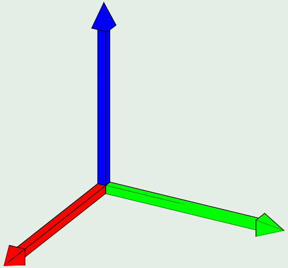


Exercício 7 (opcional) - Sistema de eixos alternativo

Criar uma função que retorna um modelo completo a 3D

- `[V,F,fColor]=setas3D()`
  - Criar os pontos das setas para yy e zz por rotações da seta em xx
  - Construir **V**, a matriz global de vértices por concatenação.
  - Construir **F**, a matriz global de faces por concatenação e ajuste dos números de vértices.
  - Construir **fColor**, a matriz de cores para as faces: [1 0 0] para o eixo xx, [0 1 0] para o eixo yy e [0 0 1] para o eixo zz.

Modelo resultante



- Para fazer o *display* do objeto, basta o comando usual: `h=patch('Vertices',V,'Faces',F, 'FaceVertexCData', fColor, 'FaceColor','flat')`
- A alteração de coordenadas continua a fazer-se com `set()`, mas com a vantagem de ser feito nos vértices: `set(h,'Vertices', V2)`