



# **Robótica Industrial**

## **Aula prática nº 8**

**Cinemática inversa – conclusão**  
**RRR antropomórfico e SCARA**

Vítor Santos

Universidade de Aveiro

14 Nov 2022

- 1 Cinemática inversa do robô RRR antropomórfico
- 2 Cinemática inversa do robô SCARA
- 3 Extensões do exercício sobre o SCARA (opcional)

# Exercício 1 - Cin. inversa do RRR antropomórfico

## Implementar a cinemática inversa do robô antropomórfico a 3 dof

- A função a implementar é  $Q = \text{invkinRRRantro}(x, y, z, L1, L2, L3)$ .
- $L1, L2, L3$  - os comprimentos dos elos
- $x, y, z$  - coordenadas da ponta (ou o mesmo que:  $P_{W_x}, P_{W_y}, P_{W_z}$ )
- devem ser retornadas as 4 redundâncias

## Equações da cinemática inversa

- $\theta_3 = \pm \arccos \frac{x^2 + y^2 + (z - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3}$
- $\theta_2 = 2 \arctan \frac{(L_3C_3 + L_2) \pm \sqrt{L_3^2S_3^2 + (L_3C_3 + L_2)^2 - (z - L_1)^2}}{L_3S_3 + z - L_1}$
- $\theta_1 = \arctan \frac{y \operatorname{sgn}(k)}{x \operatorname{sgn}(k)} = \operatorname{atan2}(y \operatorname{sgn}(k), x \operatorname{sgn}(k))$

com  $k = L_3C_{23} + L_2C_2$ . NB: Para calcular  $\theta_2$  aqui **não** se usa  $\operatorname{atan2}()$ !

# Ex 1 - Código base da função de cinemática

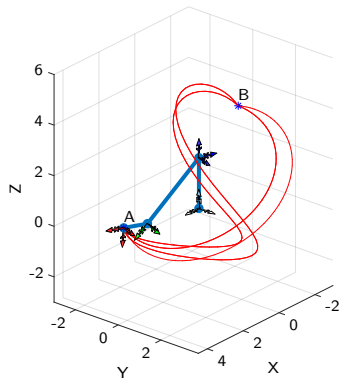
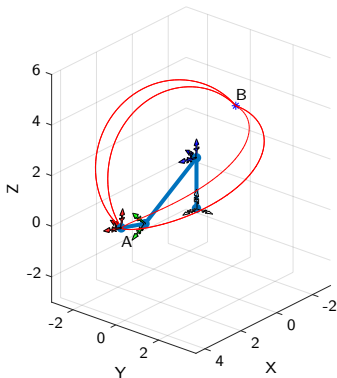
## Elementos do código sugerido

```
function Q=invkinRRRantro(x,y,z,L1,L2,L3)
q3A= %...
q3=[q3A,-q3A]; %Two solutions for the elbow
q2A = %...% depends on q3 and +sqrt; hence two solutions appear.
q2B = %...% depends on q3 and -sqrt; hence, two more solutions.
q3=[q3,q3]; % create the 4 possibilities for q3 (2 x 2)
q2=[q2A,q2B];%join the (now) 4 possibilities of q2
k=sign( L3*cos(q2+q3)+L2*cos(q2) );
q1=atan2(k*y, k*x); %obtain the 4 solutions for q1 at once
Q=[q1;q2;q3]; %Q has the 4 solutions for q1, q2 & q3
```

# Exercício 2 - Animação do RRR antropomórfico

## Animar o movimento do robô RRR antropomórfico

Usar  $L_1=2$ ,  $L_2=3$ ,  $L_3=1$ , e um movimento tal que o *end-effector* passe da postura  $A=[3,-1,0]$  para a postura  $B=[-1,1,4]$  usando as 4 redundâncias para chegar a B. De A parte sempre da redundância *cotovelo-em-baixo* & *ombro-direito* e à direita parte de *cotovelo-em-cima* & *ombro-esquerdo*.



## Ex. 2 - Valores dos ângulos para as quatro configurações nos pontos A e B

### Os valores concretos dos ângulos

Para  $L1=2$ ,  $L2=3$ ,  $L3=1$ , para os pontos  $A=[3,-1,0]$   $B=[-1,1,4]$ , o resultado da cinemática inversa, seguindo a ordem indicada, deverá ser o seguinte:

QA =

2.8198	2.8198	-0.3218	-0.3218	%q1
-2.7782	-2.3771	-0.7645	-0.3634	%q2
0.8411	-0.8411	0.8411	-0.8411	%q3

%-----

%	CB/OE	CC/OE	CB/OD	CC/OD
---	-------	-------	-------	-------

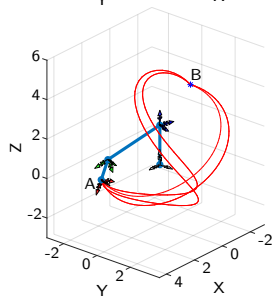
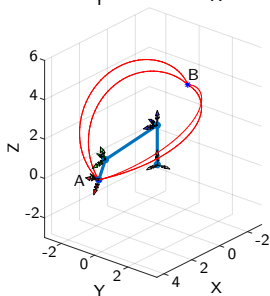
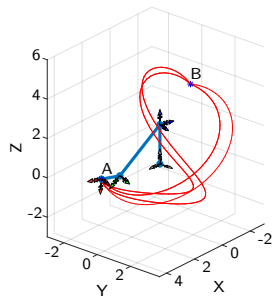
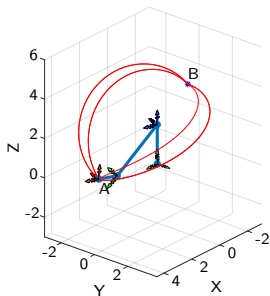
QB =

-0.7854	-0.7854	2.3562	2.3562	%q1
1.8771	2.4955	0.6461	1.2645	%q2
2.3005	-2.3005	2.3005	-2.3005	%q3

CB = cotovelo em baixo ( $q3 > 0$ ); CB = cotovelo em cima ( $q3 < 0$ )

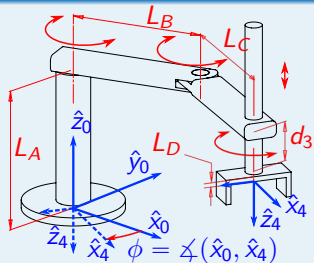
OD = ombro direito ( $|q1| < \pi/2$ ); OE = ombro esquerdo ( $|q1| > \pi/2$ );

## Ex. 2 - Os 16 caminhos para ir de A para B



# Exercício 3 - Cinemática inversa do SCARA

Sendo esta a cinemática direta do SCARA...



$${}^R T_H = \begin{bmatrix} C_{1-2-4} & S_{1-2-4} & 0 & L_B C_1 + L_C C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & L_B S_1 + L_C S_{1-2} \\ 0 & 0 & -1 & L_A - d_3 - L_D \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} x = L_B C_1 + L_C C_{1-2} \\ y = L_B S_1 + L_C S_{1-2} \\ z = L_A - d_3 - L_D \\ \phi = \arctan \frac{r_{21}}{r_{11}} = \arctan \frac{S_{1-2-4}}{C_{1-2-4}} = \theta_1 - \theta_2 - \theta_4 \end{cases}$$

... Criar a função  $Q = \text{invkinSCARA}(x, y, z, \phi, L_A, L_B, L_C, L_D)$  onde:

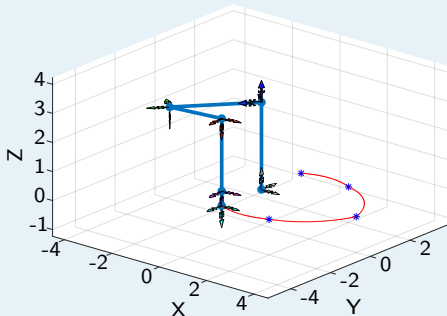
- $d_3$  só depende de  $z$
- $\theta_1$  e  $\theta_2$  são similares aos de um RR planar (porém:  $\theta_2 \rightarrow -\theta_2$ )
- $\theta_4$  depende de  $\phi$ ,  $\theta_1$  e  $\theta_2$ .



# Exercício 4 - SCARA em processo de montagem

Animar o robô passando pelos seguintes pontos num plano com  $z=0$  e orientação  $\phi = 0$

```
% x, y, z, phi  
PP = [0 -2 0 0  
      2 -2 0 0  
      4 0 0 0  
      2 2 0 0  
      0 2 0 0  
      ];
```



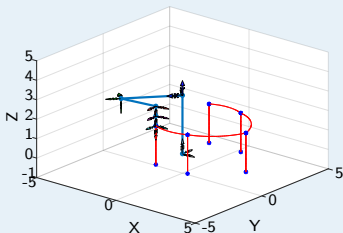
- Usar a primeira redundância
- Usar 30 pontos para cada segmento de animação
- Para outras orientações do *end-effector* bastaria alterar  $\phi$  em PP
- É preciso uma junta virtual para uma representação mais realista!

# Exercício 5 - Cotas de segurança no SCARA

## Inserir pontos com cotas de segurança em Z=2

- Inserir novos pontos antes e depois de cada ponto original para forçar o robô a passar em cotas de segurança antes de se movimentar no plano.
- Ou seja, passar para a lista de pontos à direita:

```
% x, y, z, phi
PP=[0 -2 0 0
    2 -2 0 0
    4 0 0 0
    2 2 0 0
    0 2 0 0
    ];
```



```
% x, y, z, phi
PP=[0 -2 2 0 %new
    0 -2 0 0
    0 -2 2 0 %new
    2 -2 2 0 %new
    2 -2 0 0
    2 -2 2 0 %new
    4 0 2 0 %new
    4 0 0 0
    4 0 2 0 %new
    2 2 2 0 %new
    2 2 0 0
    2 2 2 0 %new
    0 2 2 0 %new
    0 2 0 0
    0 2 2 0 %new
    ];
```

## Ex. 5 - Sugestões de execução

- Criar uma matriz intermédia com o numero de pontos final (newPP).
- Inserir nas linhas certas os pontos originais (PP).
- Copiar e ajustar as linhas antes e depois.
- Copiar o resultado de newPP para a matriz PP.
- O código seguinte pode ser completado para o efeito (substituir os '\*'):

```
newPP=zeros(*,width(PP));  
newPP(2:*:*)=PP;  
for n=*:*:~*  
    newPP(n-1,:)=newPP(n,:); newPP(*,*)=2;  
    newPP(n+1,:)=newPP(n,:); newPP(*,*)=2;  
end  
PP=newPP;
```

- Também é possível fazer sem usar o ciclo for!

# Exercício 6 (opcional) - Extensões do exercício 5

## Possibilidades de extensão do exercício anterior

- 1 Velocidades de animação diferentes para os movimentos verticais.
- 2 Definir os pontos de passagem no plano  $z=0$  como pontos aleatórios dentro do espaço de trabalho.
- 3 Definir as cotas de segurança com valor aleatório positivo.
- 4 Adicionar um caminho de retorno em sentido inverso.
- 5 Usar uma redundância aleatória em cada movimento (menos interesse).

