

**Instituto Superior Técnico**  
**Mestrado em Engenharia de Informática e Computadores**  
**Cinemática Direta e Inversa de Manipuladores Seriais**

Matheus Mello **89147**

Leonor Silva **81155**

Ricardo Pires **75513**

**Lisboa - Portugal**  
**13 de Abril de 2018**

# 1 Resumo

A cinemática de um robô manipulador é o estudo da posição e da velocidade do seu efetuador e dos seus ligamentos. Quando nos referimos a posição, é também contemplado tanto a parte de orientação. A cinemática pode ser diferenciada em direta e a inversa. Nessa primeira é desejável a obtenção da posição e a orientação do efetuador, para uma dada posição das articulações. A cinemática inversa é o oposto da cinemática direta, ou seja, dado uma posição e a orientação do efetuador quer-se obter as posições e orientação correspondentes das articulações.

## 2 Cinemática Direta

A cinemática direta consiste na determinação da posição do end-effector e da sua orientação, dados os ângulos das juntas. Este método utiliza transformações matriciais sucessivas entre cada um dos referenciais até se obter a matriz de transformação homogênea que relaciona o primeiro e o último referencial do robô.

$$T_{06} = T_{01} * T_{12} * T_{23} * T_{34} * T_{45} * T_{56} * T_{67} * T_{78} \quad (1)$$

A posição do corpo n em relação ao n-1 pode ser representada por uma matriz de posição indicada com o símbolo T ou M. Através da substituição direta dos parâmetros D-H obtidos anteriormente, obtém-se a matriz de transformação homogênea para cada ligação do braço robótico, a partir da qual é possível retirar as expressões de cada um dos ângulos gerais, alfa, beta e gama obtendo-se portanto a orientação final do end-effector.

### 2.1 Denavit-Hartenberg

Os parâmetros de Denavit-Hartenberg ou também chamados de parâmetros DH são quatro parâmetros associados a uma convenção para fixar sistemas de referência aos elos de uma cadeia cinemática espacial, ou manipulador robótico.

### 2.2 Convenção de Denavit-Hartenberg

A evolução no tempo das coordenadas das juntas de um robô representa o modelo cinemático de um sistema articulado no espaço tridimensional. A notação de Denavit-Hartenberg é uma ferramenta utilizada para sistematizar a descrição cinemática de sistemas mecânicos articulados com N graus de liberdade.

No nosso caso as relação entre frames é dada pela imagem abaixo, onde o eixo de rotação é sempre em volta de x.

Tabela 1: Parâmetros Denavit-Hartenberg

$\alpha$ (rads)	a (mm)	d (mm)	$\theta$ (rads)
0	0	99	x1
$\pi/2$	30	0	x2 + $\pi/2$
0	120	0	x3
0	25	0	0
$\pi/2$	0	140	x4
$-\pi/2$	0	0	x5
$\pi/2$	0	0	x6
0	0	25	0

### 3 Cinemática Inversa

A importância da cinemática inversa é devido à necessidade de controlar os manipuladores. Este problema baseia-se em dados os ângulos no espaço das articulações converter em posição e orientação no espaço cartesiano. Existem duas abordagens a este problema: solução algébrica ou solução geométrica. No nosso trabalho optamos pela solução geométrica. Esta abordagem tem como base a decomposição do problema em segmentos menores de modo a separar o problema de obter a orientação e o de obter a posição. Ao observar a fisionomia do braço observamos que a posição é dada por  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  e  $\theta_5$ , onde a quinta junta corresponde à posição antes do end-effector(chamado o pulso). Assim, basta eliminar a contribuição da quinta junta e a solução para  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  passa a ser um simples problema trigonométrico. Usando a equação de transformação de Denavit-Hartenberg temos que:

$$T_{06} = T_{05} * T_{56} \quad (2)$$

$$P_{06} = R_{05} * P_{56} + P_{05} \quad (3)$$

Assim, dado que sabemos a orientação do end-effector e dado que sabemos a representação do ponto 6 na frame  $\mathbb{P}_{56}$  e que ambos os pontos têm a mesma orientação, obtemos a representação do ponto 5 no world frame e, conseqüentemente, eliminamos a contribuição da última frame para a posição. Assim para encontrar  $\theta_1$  basta considerar a distância em x e em y, e depois considerar o seu complementar. Para encontrar  $\theta_2$  e  $\theta_3$ , o problema complica-se pois as juntas não tem um ponto de encontro, mas sim um desnível de 25 mm. Assim para evitar este problema, consideramos o desnível no fim da junta 3 em vez de entre a junta 2 e 3. A partir do momento, em que obtivemos  $\theta_1$ ,  $\theta_2$  e  $\theta_3$ , temos que:

$$R_{06} = R_{03} * R_{36} \quad (4)$$

e dado que,

$$\begin{bmatrix} \cos(\theta_4)*\cos(\theta_5)*\cos(\theta_6)-\sin(\theta_4)*\sin(\theta_6) & -\cos(\theta_6)*\sin(\theta_4)-\cos(\theta_4)*\cos(\theta_5)*\sin(\theta_6) & \cos(\theta_4)*\sin(\theta_5) \\ \cos(\theta_6)*\sin(\theta_5) & -\sin(\theta_5)*\sin(\theta_6) & -\cos(\theta_5) \\ \cos(\theta_4)*\sin(\theta_6)+\cos(\theta_5)*\cos(\theta_6)*\sin(\theta_4) & \cos(\theta_4)*\cos(\theta_6)-\cos(\theta_5)*\sin(\theta_4)*\sin(\theta_6) & \sin(\theta_4)*\sin(\theta_5) \end{bmatrix}$$

Para os restantes ângulos basta resolver um sistema de equações.

(5)

### 4 Singularidades

No processo de cálculo da cinemática inversa podem existir pontos nos quais existem infinitas configurações para o robô atingir esse ponto. Se não se lidar com as singularidades devidamente estas podem forçar o robô a ter de ser reiniciado e ao seu trabalho a ter de ser interrompido, pois este pode mover-se de formas imprevisíveis.

No caso do robô de laboratório este apresenta bastantes semelhanças ao PUMA 560, braço robótico industrial com 6 graus de liberdade. Daí se conclui que partilham as mesmas singularidades. Existem 3 tipos de singularidades neste caso.

Singularidade no pulso, que ocorre quando as juntas 4 e 6 estão alinhadas, isto faz com que a junta 4 e 6 possa ter infinitas posições. Singularidades no ombro estas ocorrem quando a junta 1 e 6 estão alinhadas isto permite que a junta 1 possa adquirir infinitos pontos. Singularidades de cotovelo, estas ocorrem quando o braço robótico se encontra totalmente estendido, ou seja,

quando as juntas 2,3 e 4 estão no mesmo plano. O que acontece quando se tenta atingir posições para além do alcance do robô.

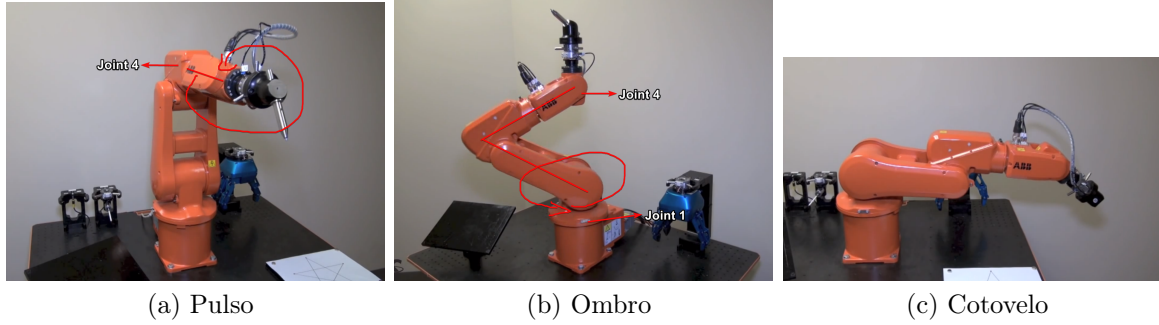


Figura 1: Singularidades

## 5 Instruções

O código está dividido em dois ficheiros matlab, sendo cada um deles uma função. Para correr o código, no caso da cinemática direta basta fazer:

$$[x, y, z, angz0, angy, angz1] = direct\_kinematics(x1, x2, x3, x4, x5, x6) \quad (6)$$

onde (x,y,z) corresponde ao ponto obtido e (angz0,angy,angz1) corresponde á orientação. No caso da cinemática inversa basta fazer:

$$[result] = inverse\_kinematics(x, y, z, angz0, angy, angz1) \quad (7)$$

onde result corresponde a uma matriz com todas os conjuntos de ângulos possíveis obtidos. No caso de um singularidade, um pop-up aparece identificando a singularidade.

## 6 Análise de Dados

### 6.1 Cinemática Direta

```
direct_kinematics(0, 0, 0, 0, 0, 0) = 195.000 -0.000 244.000 -0.000 1.571 3.142
direct_kinematics(pi, pi/6, 0, 0, pi, pi/2) = -57.093 0.000 282.074 -0.000 2.094 1.571
direct_kinematics(pi/2, pi/2, pi/2, pi/2, pi/2, pi/2) = 25.000 -230.000 74.000 -0.000 1.571
3.142
direct_kinematics(0, pi/2, 0, 0, pi/2, pi/2) = -140.000 -0.000 239.000 -3.142 1.571 1.571
direct_kinematics(0, pi/2, 0, 0, pi/3, 0) = -136.651 -0.000 251.500 -3.142 1.047 0.000
direct_kinematics(pi/6, pi, 0, pi, 0, pi/2) = -116.913 -67.500 -46.000 -2.618 1.571 -1.571
direct_kinematics(-pi, 0, pi/2, pi/2, 0, 0) = -5.000 -0.000 384.000 1.571 0.000 0.000
direct_kinematics(0, 0, 0, pi/2, pi/2, 0) = 170.000 -25.000 244.000 -1.571 1.571 -1.571
direct_kinematics(pi, pi, pi, pi, pi, pi) = -145.000 0.000 4.000 -0.000 1.571 0.000
direct_kinematics(pi, pi/4, pi/2, pi/4, pi/4, 0) = 192.864 12.500 268.831 0.530 1.424 0.530
```

## 6.2 Cinemática Inversa

```

inverse_kinematics(195, 0, 244, 0, 1.571, 3.142) =
    0      0      0.0018      0      0.0018      0
    0 -1.5357  2.7900   3.1416  1.2543  -3.1416
3.1416  1.3170  0.7094      0      1.1152  -3.1416
3.1416  0.5700  2.0824      0      0.4892  -3.1416
inverse_kinematics(-57.093, 0.000, 282.074, -0.000, 2.094, 1.571) =
3.1416  0.5236  0.0017      0      3.1394  1.5710
3.1416 -1.0122  2.7901      0      1.8869  1.5710
6.2832  1.1100  0.2948   3.1416  1.9280  1.5710
6.2832 -0.0950  2.4971   3.1416  2.9252  1.5710
inverse_kinematics(25, -230, 74, 0, 1.571, 3.142) = [wristsingularity]
-1.5708 -0.9273  0.0018 -1.5708  1.5708  0.6453
-1.5708 -2.4630  2.7900 -1.5708  1.5708  1.8978
 1.5708  1.7625  1.2192  1.5708  1.5708  1.7307
 1.5708  1.5708  1.5726  1.5708  1.5708  1.5690
inverse_kinematics(-140, 0, 239, -3.142, 1.571, 1.571) = [wristsingularity]
-3.1415  0.4615 -0.4054  3.1328  0.0563  1.5798
 0.0001  1.5708  0.0019 -0.0005  1.5692  1.5710
 0.0001  0.0351  2.7900 -0.0016  0.3168  1.5725
inverse_kinematics(-140, 0, 239, -3.142, 1.571, 0) = [wristsingularity]
-3.1415  0.4615 -0.4054  3.1328  0.0563  0.0088
 0.0001  1.5708  0.0019 -0.0005  1.5692      0
 0.0001  0.0351  2.7900 -0.0016  0.3168  0.0015
inverse_kinematics(-136.651, -0.000, 251.500, -3.142, 1.047, 0.000)
-3.1415  0.4615 -0.4055  0.0009  0.4678  3.1405
 0.0001  1.5708  0.0018 -0.0005  1.0452  0.0000
 0.0001  0.0351  2.7900 -3.1396  0.2073  3.1394
inverse_kinematics(-5, 0, 384, -1.571, 0, 0) =
-3.1416  0.1917  1.2192      0      0.1599 -1.5710
-3.1416      0      1.5726  1.5710  0.0018      0
inverse_kinematics(170, -25, 244, -1.571, 1.571, -1.571) = [wristsingularity]
    0      0      0.0019  1.5710  1.5710 -0.0021
    0 -1.5357  2.7899  1.5707  1.5711 -1.2545
3.1416  1.3169  0.7096 -1.5709  1.5711 -1.1153
3.1416  0.5701  2.0822 -1.5707  1.5711 -0.4895
inverse_kinematics(-145, 0, 4, 0, 1.571, 0) =
3.1416 -1.1924 -0.3516  3.1416  1.5974 -3.1416
3.1416 -3.1416 -3.1434  3.1416  3.1395 -3.1416
6.2832  2.6358  0.2606  3.1416  2.8967      0
6.2832  1.3926  2.5312      0      2.3592 -3.1416
inverse_kinematics(192.864, 12.500, 268.831, 0.530, 1.424, 0.530) = [elbowsingularity]
    0 -0.0591  0.2164 -1.5494  0.5239 -0.9800
    0 -1.3518  2.5754 -2.5552  1.1285  0.3385
3.1416  0.9770  1.2195  0.6898  0.9041  0.1438
3.1416  0.7855  1.5724  0.7866  0.7844  0.0016
inverse_kinematics(0, 0, 200, 0, 0, 0) = [shouldersingularity]
    0 1.9033 -1.1343  0  0.8018 -3.1416
3.1416 1.9033  1.1343  0  0.8018      0

```

\* Apesar de os resultados apenas apresentarem 3 casas decimais, para o cálculo da cinemática inversa foram utilizados os resultados totais apresentados pelo matlab para se conseguir obter a máxima precisão.

\*\* Quando uma singularidade é encontrada o programa lança uma mensagem referindo o tipo de singularidade.