



INSTITUTO SUPERIOR TÉCNICO

ROBÓTICA

Direct and Inverse Kinematics of Serial Manipulators

1.º TRABALHO DE LABORATÓRIO

Autores:

João BORGES

75588

Rui GUERRA

75737

Ano Lectivo:

2015-2016

Lisboa, Março de 2016

1 Introdução

O problema proposto neste trabalho consiste na representação da pose (posição e orientação) da ponta de um braço robótico (denominado *end-effector*) recorrendo apenas aos graus de liberdade dados pelos ângulos das juntas deste, bem como a determinação das possíveis combinações destes ângulos para obter uma certa pose.

Numa primeira fase, pretende-se determinar esta pose a partir de um conjunto de 6 graus de liberdade $(\theta_1, \dots, \theta_6)$ recorrendo apenas a transformações de coordenadas baseadas na cinemática do braço robótico.

Numa segunda fase, partindo de uma pose $(x, y, z, \alpha, \beta, \gamma)$ conhecida, pretende-se conhecer todas as possíveis combinações de ângulos θ para que esta pose se verifique.

Com vista a solucionar estes problemas, implementaram-se dois programas de MATLAB que permitem determinar a pose a partir dos ângulos e vice-versa.

2 Cinemática Directa

O problema de determinar a pose do *end-effector* do braço robótico a partir dos ângulos $(\theta_1, \dots, \theta_6)$ é um problema de cinemática directa que, para ser resolvido, é necessário percorrer um conjunto de passos.

Em primeiro lugar, é necessário estabelecer os referenciais que correspondem a cada junta e definir as relações entre estes. Este passo é importante porque uma boa definição de referenciais será útil para a simplificação do cálculo a ser feito.

A partir destes referenciais será, então, criada uma matriz de transformação geral de onde se poderá extrair a pose do *end-effector*.

2.1 Estabelecimento de Referenciais

Os referenciais foram escolhidos tendo em vista a utilização da convenção Denavit-Hartenberg (D-H) para uma determinação simplificada das transformações entre estes. Uma representação esquemática das posições relativas dos referenciais encontra-se representada na figura 1.

Para além do referencial da base (0), introduziram-se referenciais correspondendo a cada grau de liberdade (1 a 6) de forma que o eixo de rotação corresposse sempre ao eixo z do referencial. O sentido dos ângulos $(\theta_1, \dots, \theta_6)$ é também visível no esquema. De notar que os referenciais 1 e 2 bem como os A, 5 e 6 têm a sua origem no mesmo ponto, mas encontram-se representados separadamente para maior facilidade de observação.

Um referencial auxiliar (A) foi também utilizado para representar uma translação para ser possível utilizar a convenção D-H para determinar a transformação entre todos os referenciais, que de outra forma não seria possível.

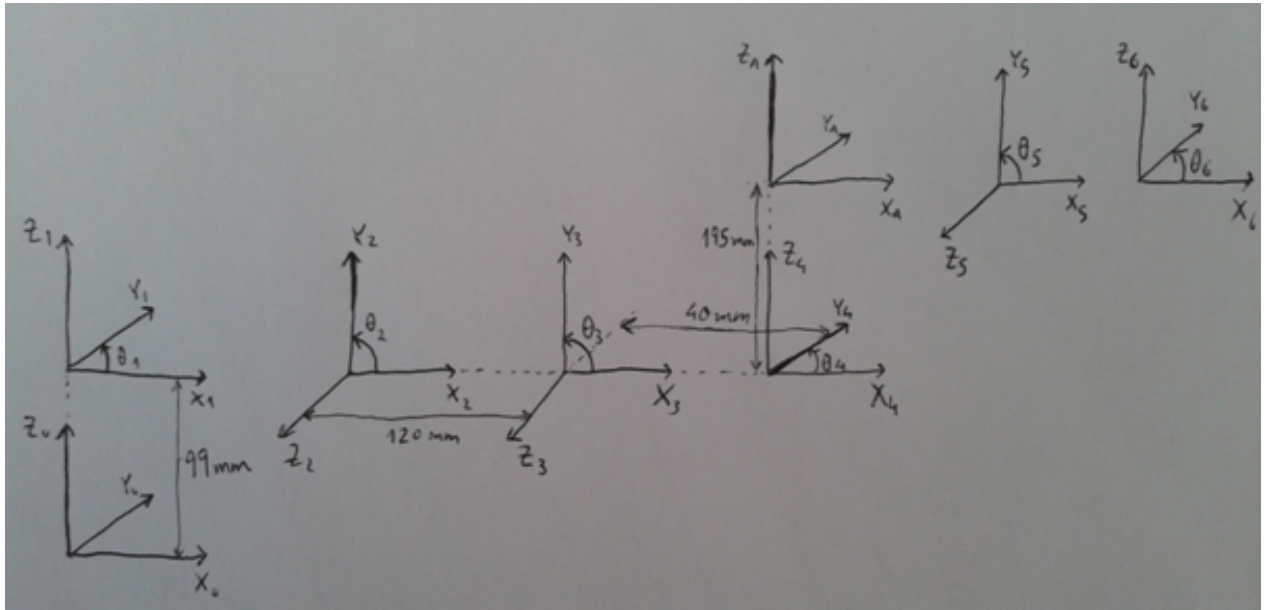


Figura 1: Representação gráfica dos referenciais usados.

2.2 Cálculo da Matriz de Transformação

A partir dos referenciais escolhidos, criou-se uma tabela com os parâmetros a ser utilizados pela convenção D-H. Esta encontra-se representada na tabela 1.

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	A=99 mm	θ_1
2	0	$\frac{\pi}{2}$	0	θ_2
3	B=120 mm	0	0	θ_3
4	C=40 mm	$-\frac{\pi}{2}$	0	θ_4
A	0	0	D=195 mm	0
5	0	$\frac{\pi}{2}$	0	θ_5
6	0	$-\frac{\pi}{2}$	0	θ_6

Tabela 1: Parâmetros da convenção D-H obtidos por inspeção dos referenciais.

Uma vez obtidos estes parâmetros, é possível escrever as matrizes de transformação entre referenciais consecutivos

$$\begin{aligned}
{}^0_1T &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & A \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2T &= \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^2_3T &= \begin{bmatrix} c_3 & -s_3 & 0 & B \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3_4T &= \begin{bmatrix} c_4 & -s_4 & 0 & C \\ 0 & 0 & 1 & 0 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^4_A T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^A_5T &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& & {}^5_6T &= \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{1}$$

Para simplificação de notação, as funções trigonométricas $\sin(\theta_x)$ e $\cos(\theta_x)$ serão abreviadas para s_x e c_x e as constantes relativas a distâncias serão representadas por A, B, C e D, como se indica na tabela 1. A matriz de transformação geral pode ser então obtida por

$${}^0_6T = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot {}^4_AT \cdot {}^A_5T \cdot {}^5_6T = \begin{bmatrix} {}^0R & {}^0P \\ 0 & 1 \end{bmatrix}. \tag{2}$$

2.3 Obtenção da Pose do *End-effector*

A partir da matriz 0_6T é possível determinar os parâmetros da pose do *end-effector*. A posição (x, y, z) é obtida directamente a partir das entradas (1,4), (2,4) e (3,4), respectivamente, desta matriz. Para definir os parâmetros da orientação (α, β, γ) escolheu-se uma convenção de ângulos de Euler Z-Y-X. Segundo esta convenção, estes ângulos podem ser obtidos somente a partir da matriz de rotação 0R (com entradas representadas por r_{ij}) através das expressões

$$\alpha = \text{atan2}\left(\frac{r_{21}}{c_\beta}, \frac{r_{11}}{c_\beta}\right) \quad \beta = \text{atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \quad \gamma = \text{atan2}\left(\frac{r_{32}}{c_\beta}, \frac{r_{33}}{c_\beta}\right) \tag{3}$$

para $c_\beta \neq 0$. Caso contrário $\alpha = 0$, $\beta = \pm \frac{\pi}{2}$ e $\gamma = \pm \text{atan2}(r_{12}, r_{22})$.

2.4 Testes Experimentais

Após implementar a função *direct_kinematics.m* em MATLAB, realizaram-se diversos testes experimentais para verificar o bom funcionamento do programa desenvolvido. Na figura 2 apresentam-se dois testes, onde se observam a representação do braço robótico e a pose final do *end-effector*. O teste à esquerda teve como *input* os ângulos $(0, 0, 0, 0, 0, 0)$ e o teste à direita teve os ângulos $(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, 0, \frac{\pi}{2}, 0)$.

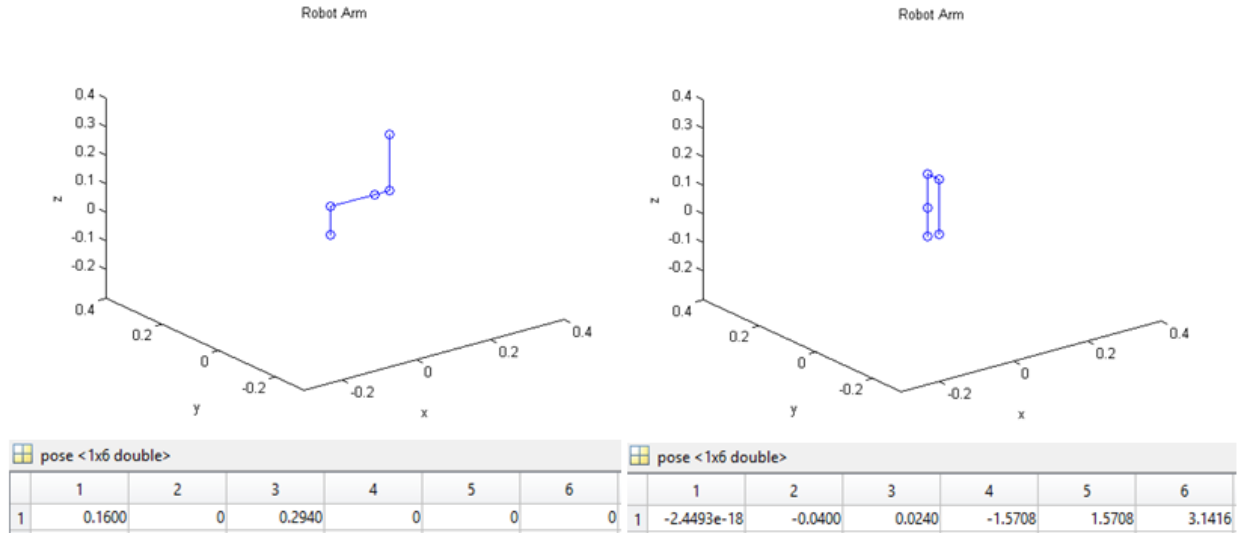


Figura 2: Representação gráfica do braço robótico e *output* da função *direct_kinematics.m* para dois exemplos.

É de notar que não é possível visualizar os efeitos dos três últimos graus de liberdade no gráfico devido a estes apenas influenciarem a orientação final do *end-effector*. De qualquer forma, o bom funcionamento do programa pode ser verificado de acordo com os referenciais estabelecidos anteriormente a partir do *output* da função.

3 Cinemática Inversa

Determinar os valores dos ângulos $(\theta_1, \dots, \theta_6)$ do braço robótico a partir da pose do *end-effector* é um problema mais complexo, devido à multiplicidade de soluções para cada pose e à complexidade do cálculo numérico para cada um dos ângulos.

3.1 Construção da Matriz de Transformação

Em primeiro lugar, é necessário criar a matriz de transformação a partir da pose de *input* $(x, y, z, \alpha, \beta, \gamma)$. Considerando novamente a convenção de ângulos de Euler Z-Y-X, é possível escrever esta matriz em função destes parâmetros através da expressão

$${}_{tool}^{base}T = {}_6^0T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & x \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & y \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

3.2 Cálculo dos Ângulos das Juntas

Utilizando ${}_{tool}^{base}T$ é possível determinar todas as soluções possíveis para $(\theta_1, \dots, \theta_6)$. A fórmula geral usada para obter as expressões de cada θ foi

$$[{}^0_n T(\theta_1, \dots, \theta_n)]^{-1} \cdot {}_{tool}^{base} T = {}^n_6 T(\theta_{n+1}, \dots, \theta_6) \quad (5)$$

para diferentes valores de n escolhidos de forma a simplificar o cálculo simbólico para cada θ . As expressões são obtidas escolhendo uma ou duas entradas da matriz que resultem em equações simples que dependam de apenas um ângulo desconhecido. Por exemplo, as expressões obtidas para θ_1 e θ_3 foram

$$\theta_1 = \text{atan2}\left(\frac{y}{x}\right) \quad \vee \quad \theta_1 = \text{atan2}\left(\frac{y}{x}\right) - \pi \quad (6)$$

e

$$\theta_3 = \text{atan2}(C, D) - \text{atan2}\left(K, \pm\sqrt{C^2 + D^2 - K^2}\right), \quad (7)$$

sendo

$$K = \frac{(c_1 x + s_1 y)^2 + (z - A)^2 - B^2 - C^2 - D^2}{2B}. \quad (8)$$

Os restantes ângulos têm expressões mais complexas mas que foram possíveis de obter apenas com dependências de outros ângulos já obtidos anteriormente. Estas poderão ser visualizadas no ficheiro *inverse_kinematics.m* no MATLAB.

Como se pode observar, θ_1 e θ_3 têm duas soluções cada. Analisando os restantes ângulos também se poderá concluir que os ângulos θ_4 , θ_5 e θ_6 têm duas combinações de soluções. Assim, poder-se-á concluir que existem no total 8 soluções para $(\theta_1, \dots, \theta_6)$ que satisfaçam a pose desejada. Contudo, é possível não haver solução para os ângulos, o que acontece quando a raiz quadrada utilizada no cálculo de θ_3 produz um número imaginário. É possível também acontecer uma singularidade caso θ_5 seja múltiplo de π . Neste caso, não é possível distinguir os efeitos de θ_4 e θ_6 pelo que os resultados obtidos para estas variáveis não são únicos, sendo que estas dependem uma da outra. Por fim, outra singularidade pode ocorrer caso $x = y = 0$. Quando isto sucede, θ_1 poderá tomar qualquer valor para satisfazer a pose desejada.

3.3 Testes Experimentais

De seguida demonstram-se vários exemplos de possíveis *outputs* para a função desenvolvida *inverse_kinematics.m*. Na figura 3 estão representadas as soluções obtidas para três poses. As soluções de cima correspondem às poses obtidas nos exemplos da figura 2, e a de baixo corresponde às soluções da pose $(0, 0, 0.01, 0, 0, 0)$.

theta <8x6 double>							theta <8x6 double>						
	1	2	3	4	5	6		1	2	3	4	5	6
1	0	3.8171e-16	-4.1633e-16	0	3.4626e-17	0	1	-1.5708	2.5507	1.5708	3.1416	2.5507	-1.0170e-16
2	0	1.7673	-2.7369	0	0.9696	0	2	-1.5708	1.5708	1.9754	3.1416	1.9754	-4.8213e-17
3	-3.1416	1.3742	-4.1633e-16	-3.1416	1.3742	-1.4638e-16	3	1.5708	1.5708	1.5708	2.2204e-16	1.5708	1.1119e-31
4	-3.1416	3.1416	-2.7369	-3.1416	0.4046	-2.3504e-16	4	1.5708	0.5909	1.9754	2.6464e-16	2.1461	1.4398e-16
5	0	3.8171e-16	-4.1633e-16	-3.1416	-3.4626e-17	-3.1416	5	-1.5708	2.5507	1.5708	0	-2.5507	-3.1416
6	0	1.7673	-2.7369	-3.1416	-0.9696	-3.1416	6	-1.5708	1.5708	1.9754	0	-1.9754	-3.1416
7	-3.1416	1.3742	-4.1633e-16	0	-1.3742	-3.1416	7	1.5708	1.5708	1.5708	-3.1416	-1.5708	-3.1416
8	-3.1416	3.1416	-2.7369	0	-0.4046	3.1416	8	1.5708	0.5909	1.9754	-3.1416	-2.1461	-3.1416

theta <8x6 double>						
	1	2	3	4	5	6
1	0	2.1972	1.5079	0	2.5780	0
2	0	0.9443	2.0383	3.1416	2.9827	3.1416
3	-3.1416	2.1972	1.5079	0	2.5780	3.1416
4	-3.1416	0.9443	2.0383	3.1416	2.9827	-2.4339e-16
5	0	2.1972	1.5079	-3.1416	-2.5780	-3.1416
6	0	0.9443	2.0383	0	-2.9827	0
7	-3.1416	2.1972	1.5079	-3.1416	-2.5780	0
8	-3.1416	0.9443	2.0383	0	-2.9827	-3.1416

Figura 3: *Output* da função *inverse_kinematics.m* para três exemplos.

Observa-se o bom funcionamento do programa desenvolvido no segundo exemplo, já que a entrada usada na cinemática directa corresponde à terceira solução obtida, a menos de erros de arredondamento. No primeiro exemplo, tem-se a situação de existir a singularidade de θ_5 ser múltiplo de π . Enquanto se observa que, de facto, a pose usada na cinemática inversa corresponde à primeira solução indicada na lista, o programa avisa que a singularidade existe e que, portanto, existem outras soluções. θ_4 e θ_6 não se distinguem, e como neste caso $\theta_5 = 0$, tem-se apenas que $\theta_4 = -\theta_6$ (caso $\theta_5 = \pi$, ter-se-ia $\theta_4 = \theta_6$). No terceiro exemplo observa-se o caso de existir a singularidade $x = y = 0$. Neste caso é dado um aviso a indicar que θ_1 pode tomar qualquer valor para além dos indicados na solução. No caso de o ponto ser inválido (por ser impossível o braço robótico chegar a essa posição), o *output* da função é simplesmente o valor -1.

4 Instruções de Utilização das Funções MATLAB

A função *direct_kinematics.m* recebe como argumento um vector de dimensão 6 correspondendo aos ângulos $(\theta_1, \dots, \theta_6)$. A saída é também um vector de dimensão 6 correspondendo à pose $(x, y, z, \alpha, \beta, \gamma)$ do *end-effector* de acordo com a convenção de ângulos de Euler Z-Y-X. Um exemplo de utilização desta função é o comando "pose=direct_kinematics(theta);", se theta corresponder ao vector de ângulos.

A função *inverse_kinematics.m* recebe como argumento um vector de dimensão 6 indicando a pose do *end-effector* desejada $(x, y, z, \alpha, \beta, \gamma)$ usando a mesma convenção. A saída é uma matriz de 8 linhas por 6 colunas correspondendo às 8 soluções possíveis de ângulos $(\theta_1, \dots, \theta_6)$, ou então apenas um valor igual a -1 caso o ponto seja inválido. Um exemplo de utilização é o comando "theta=inverse_kinematics(pose);".

5 Conclusões

No trabalho realizado foi possível resolver o problema de determinar a pose do *end-effector* do braço robótico a partir dos ângulos das suas juntas $(\theta_1, \dots, \theta_6)$, bem como o problema inverso. Após a definição dos referenciais a usar e a escolha da convenção de ângulos, o problema a resolver resumiu-se em grande parte à realização de manipulações algébricas.

O resultado obtido foi duas funções de MATLAB, denominadas *direct_kinematics.m* e *inverse_kinematics.m*, que produzem todas as soluções possíveis para cada problema, detectando singularidades e entradas inválidas.