



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Departamento de Engenharia Eletrotécnica e de Computadores
Secção de Robótica e Manufatura Integrada

Robótica

2014/2015 – 2.º Semestre
Trabalho 1

Cinemática Directa e Inversa

Pedro Martins, n.º 31501
João Barata Oliveira, n.º 31559
Alexandra Videira, n.º 37246

Regente: José António Barata de Oliveira
jab@uninova.pt

Índice

Introdução.....	3
Enquadramento Teórico	4
Cinemática Directa.....	4
Cinemática Inversa.....	7
Implementação	9
Funções correspondentes aos requisitos funcionais.....	9
Funções auxiliares de conversão	10
Graphical User Interface (GUI).....	10
Extra	10
Conclusão.....	11

Introdução

Este trabalho tem como principal objectivo consolidar os conceitos fundamentais descritos durante as aulas teóricas através da simulação do modelo cinemático do robot ROB3. O ROB3 é um robot antropomórfico com 6 graus de liberdade – um por cada eixo – que permitem modelar com grande nível de precisão os seus movimentos. Assim sendo, recorre-se à implementação das cinemáticas directa e inversa a fim de, respectivamente:

- 1) Calcular a posição e orientação da gripper, dado um conjunto de ângulos;
- 2) Calcular os ângulos para cada eixo, dada uma determinada posição e orientação da gripper.

Para se poder manipular a posição angular dos eixos do robot, é necessário mapear a posição de cada eixo para as correspondentes coordenadas cartesianas (x, y, z) e respectiva orientação $(roll, pitch, yaw)$.

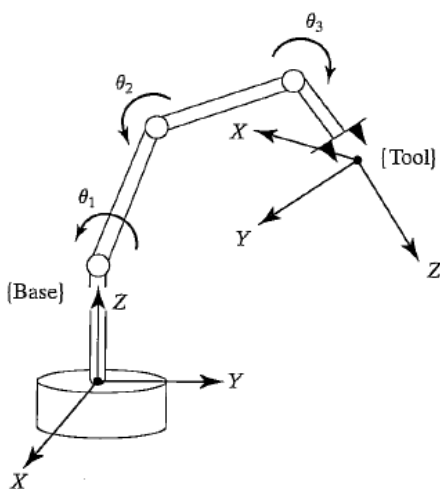
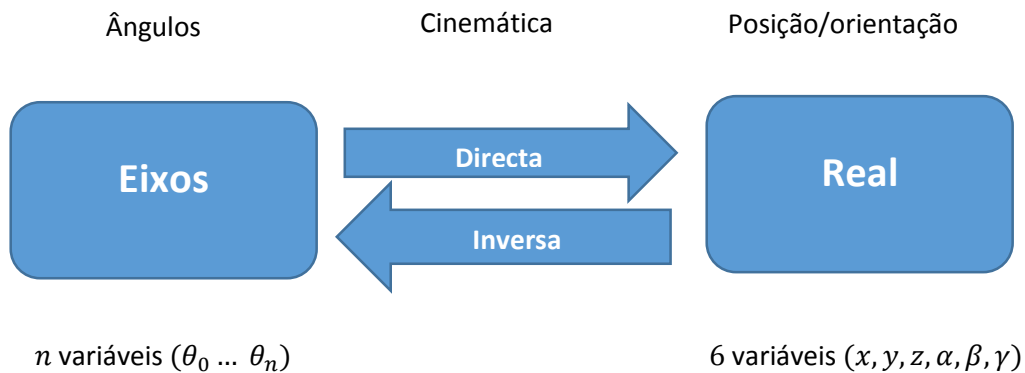
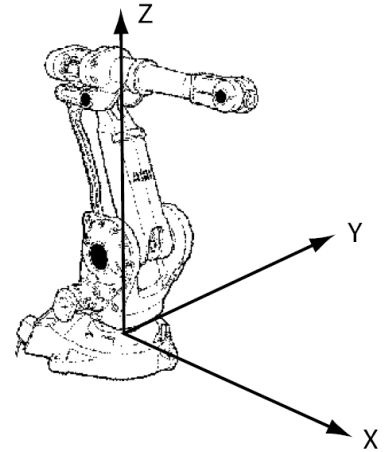
Os requisitos funcionais implementados no trabalho são os seguintes:

R.F	Descrição
1	Rodar uma junta para uma posição em steps e graus.
2	Rodas todas as juntas para as posições correspondentes em steps e graus.
3	Rodar uma junta para uma posição com velocidade em steps e graus.
4	Rodas todas as juntas para as posições correspondentes com velocidade em steps e graus.
5	Determinar a posição de uma junta em steps e graus.
6	Determinar a posição de todas as juntas em steps e graus.
7	Movimentar cada junta. Cada movimento deve providenciar a posição da garra do robot em coordenadas cartesianas robóticas (x, y, z) e a respectiva orientação $(roll, pitch, yaw)$.
8	Movimentar a garra para uma posição determinada por coordenadas cartesianas robóticas (x, y, z) e a respectiva orientação $(roll, pitch, yaw)$.
9	Calibrar o robot
10	Extra: Fazer uma série de movimentos consecutivos para imitar um determinado comportamento com o robot (p.e aperto de mão).

Enquadramento Teórico

Dadas as características das juntas e os eixos do robot, como se pode determinar a sua localização e a orientação da gripper?

A cinemática estuda os movimentos dos robots através da dedução das equações dos movimentos relativos dos vários eixos. Existem então dois tipos de cinemática: a directa e a inversa.



Cinemática Directa

A cinemática directa permite a obtenção de uma posição (x, y, z) a partir dos ângulos dos eixos do robot. Através desta cinemática pode-se calcular a posição e orientação da gripper em relação à base.

$$Rot_{X_i}(a_{i,i+1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a_{i,i+1} & -\sin a_{i,i+1} & 0 \\ 0 & \sin a_{i,i+1} & \cos a_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim sendo, todas as transformações anteriores dão origem à matriz seguinte com as suas equações demonstradas abaixo.

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = \cos t_0 \cos t_4 \sin(t_1 + t_2 + t_3) - \sin t_0 \sin t_4$$

$$n_y = \cos t_4 \sin t_0 \sin(t_1 + t_2 + t_3) + \cos t_0 \sin t_4$$

$$n_z = -\cos(t_1 + t_2 + t_3) \cos t_4$$

$$s_x = -\cos t_0 \cos t_4 - \cos t_0 \sin(t_1 + t_2 + t_3) \sin t_4$$

$$s_y = \cos t_0 \cos t_4 - \sin t_0 \sin(t_1 + t_2 + t_3) \sin t_4$$

$$s_z = \cos(t_1 + t_2 + t_3) \sin t_5$$

$$a_x = \cos(t_0) \cos(t_1 + t_2 + t_3)$$

$$a_y = \cos(t_1 + t_2 + t_3) \sin t_0$$

$$a_z = \sin(t_1 + t_2 + t_3)$$

$$p_x = \cos(t_0) (a_2 \cos(t_1) + a_3 \cos(t_1 + t_2) + d_5 \cos(t_1 + t_2 + t_3))$$

$$p_y = (a_2 \cos(t_1) + a_3 \cos(t_1 + t_2) + d_5 \cos(t_1 + t_2 + t_3)) \sin t_0$$

$$p_z = d_1 + a_2 \sin t_1 + a_3 \sin(t_1 + t_2) + d_5 \sin(t_1 + t_2 + t_3)$$

Parâmetros Denavit-Hartenberg Stanford Arm

i	θ	α	a_i	d_i
1	0	90	0	275
2	0	0	200	0
3	0	0	130	0
4	90	90	0	0
5	180	0	0	130

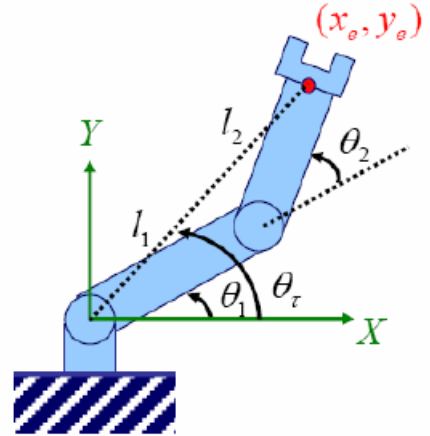
$$roll = \tan^{-1} \frac{n_x}{n_y}$$

$$pitch = \tan^{-1} \frac{n_x \cos roll + n_y \sin roll}{-n_z}$$

$$yaw = \tan^{-1} \frac{s_y \cos roll - s_x \sin roll}{-a_y \cos roll + a_x \sin roll}$$

Cinemática Inversa

O problema da cinemática inversa aborda a determinação do conjunto de ângulos dos eixos que correspondem à posição da gripper, especificado a sua orientação. É, portanto, o problema oposto da cinemática directa.



As equações que constam na matriz Denavit-Hartenberg são as que se seguem:

$$n_x = \cos(roll) \cos(pitch)$$

$$n_y = \sin(roll) \cos(pitch)$$

$$n_z = -\sin(pitch)$$

$$s_x = \cos(roll) \sin(pitch) \sin(yaw) - \sin(roll) \cos(yaw)$$

$$s_y = \sin(roll) \sin(pitch) \sin(yaw) + \cos(roll) \cos(yaw)$$

$$s_z = \cos(pitch) \sin(yaw)$$

$$a_x = \cos(roll) \sin(pitch) \cos(yaw) + \sin(roll) \sin(yaw)$$

$$a_y = \sin(roll) \sin(pitch) \cos(yaw) - \cos(roll) \sin(yaw)$$

$$a_z = \cos(pitch) \cos(yaw)$$

$$\theta_0 = \tan^{-1} \frac{p_y}{p_x}$$

$$\theta_{123} = \tan^{-1} \frac{a_z}{a_x \cos \theta_0 + a_y \sin \theta_0}$$

$$\theta_2 = \tan^{-1} \frac{y}{x}$$

Onde:

$$x = \frac{(p_x \cos \theta_0 + p_y \sin \theta_0 - d_5 \cos \theta_{123})^2 + ((p_z - d_1 - d_5) \sin \theta_{123})^2}{2.0 a_2 a_3}$$

$square\ root = 1 - x^2$ (considera-se igual a zero se o valor de square root for inferior a zero)

$$y = \sqrt[2]{square\ root}$$

$$\theta_1 = \tan^{-1} \frac{y}{x}$$

Onde:

$$x = (a_2 + a_3 \cos \theta_2)(-d_5 \cos \theta_{123} + \cos \theta_0 p_x + \sin \theta_0 p_y) + a_3 \sin \theta_2(-d_1 + p_z - d_5 \sin \theta_{123})$$

Implementação

Funções correspondentes aos requisitos funcionais

Todas as funções abaixo citadas foram implementadas com base nas limitações dos eixos presentes na seguinte tabela:

Eixo 1 (base)	160°
Eixo 2 (shoulder)	100°
Eixo 3 (elbow)	100°
Eixo 4 (wrist)	200°
Eixo 5 (tool)	200°
Gripper	60 mm

- **move_one_axis** – move um eixo para um determinado ângulo
- **move_one_axis_speed** – move um eixo para um determinado ângulo com uma dada velocidade
- **move_multiple_axis** – move todos os eixos para os ângulos especificados
- **move_multiple_axis_speed** – move todos os eixos para os ângulos especificados com as velocidades atribuídas
- **motor_status** – determina a posição de um dos eixos
- **all_motor_status** – determina a posição de todos os eixos
- **direct_kinematic** – função que permite determinar a localização da gripper quando são conhecidos os ângulos correspondentes a cada eixo, tendo em conta que estes são os parâmetros de entrada da função. A posição da gripper é dada em coordenadas cartesianas (x,y,z) e a sua respectiva orientação (roll, pitch, yaw)
- **backward_kinematic** – função que permite movimentar a gripper para uma determinada posição, determinada por coordenadas cartesianas (x,y,z) e a sua respectiva orientação (roll, pitch, yaw)
- **calibrate** – coloca todos os eixos do robot na sua posição inicial

Funções auxiliares de conversão

- **degrees_to_steps** – função que converte graus para steps
- **mm_to_steps** – função que converte milímetros para steps
- **steps_to_degrees** – função que converte steps para graus
- **steps_to_mm** – função que converte steps para milímetros

Graphical User Interface (GUI)

A interface gráfica foi realizada em Windows Forms e contém todos os requisitos funcionais. A GUI foi dividida em separadores, de maneira a tornar-se mais *user friendly*. Num dos separadores encontra-se o controlo do robot por steps e noutro o controlo do robot por ângulos. Também se encontram presentes as cinemáticas directa e inversa com uma pequena ilustração para o utilizador saber qual a orientação dos eixos (x, y, z). A interface foi feita com base nas funções presentes nos requisitos funcionais que anteriormente já tinham sido realizadas, apesar de ter havido uma adaptação das mesmas. Foi tida em conta a integridade do robot ao longo da realização de todas as funções, tal como se pode verificar quando é testada através da GUI.

Extra

Realizou-se uma série de movimentos consecutivos de maneira a que o robot imite o gesto de um aperto de mão. A partir de qualquer posição o robot consegue realizar este movimento, voltando posteriormente à posição inicial. Acede-se a esta funcionalidade através da interface gráfica.

Conclusão

Este projeto, tal como pretendido, permitiu consolidar o conhecimento sobre as cinemáticas directa e inversa, tendo em conta que se teve de estudar as deduções das mesmas por forma a compreender a sua aplicação prática relativamente ao robot. As expressões finais relativas a ambas as cinemáticas foram dadas pelo docente, o que facilitou bastante para a integração das mesmas no código.

Relativamente aos requisitos funcionais, foi pedida a realização de diversas etapas em que algumas se tornaram bastante semelhantes, pois diferem apenas no facto de mexer um eixo ou todos em conjunto (apenas um exemplo). Isto verificou-se numa extensão de linhas código desnecessária, visto que levou à repetição do mesmo código inúmeras vezes.

Apesar de tudo, foi bastante interessante ver a capacidade de adaptação das cinemáticas na simulação do robot e saber que o código realizado pode ser aplicado no mundo real. Para tal, foi também tida em conta a integridade do robot, de maneira a que não se comprometa as limitações físicas do mesmo.

Lab-Work Results nr. 1	
Course	Robotics
Year	2014/2015
Student n.º 31051	Name: Pedro Martins
Student n.º 31559	Name: João Barata Oliveira
Student n.º 37246	Name: Alexandra Videira

Requirements' answers					
Functional Requirement	Success (100%)	Almost done	Unable to finish with source-code (partial results)	Unable to fulfill it	Professor's review (leave it blank)
1	X				
2	X				
3	X				
4	X				
5	X				
6	X				
7	X				
8	X				
9	X				
10	X				
11	X				
12	X				

Non-Functional Requirement	Success	Almost done	Unable to finish with source-code (partial results)	Unable to fulfill it	Professor's review (leave it blank)
User Interface	X				
Robot Integrity	X				