```
clear all
close all
clc
startup_rvc;
```

```
Robotics, Vision & Control: (c) Peter Corke 1992-2011 http://www.petercorke.com
- Robotics Toolbox for MATLAB (release 10.2)
 - ARTE contributed code: 3D models for robot manipulators (C:\Users\Abigail Musa Gindaus\Downloads\rvctools\robot\data\ARTE)
 - pHRIWARE (release 1.1): pHRIWARE is Copyrighted by Bryan Moutrie (2013-2019) (c)
```

Geração das matrizes de transformação

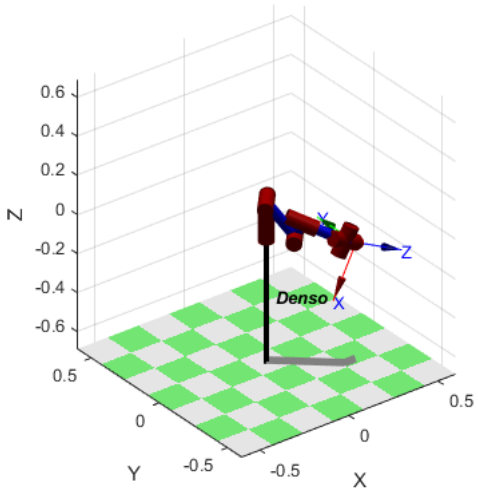
```
syms q1 q2 q3 q4 q5 q6;
% Rz
R_0_1 = [cos(q1) -sin(q1) 0; sin(q1) cos(q1) 0; 0 0 1];
% R_x quando q1 = pi/2
R_x = [1 0 0; 0 0 -1; 0 1 0];
 % Matriz de transformação do 0_1
 T_0_1 = [cos(q1) 0 sin(q1) 0; sin(q1) 0 -cos(q1) 0; 0 1 0 0.125; 0 0 0 1]
  % Matriz de transformação do 1_2
 T_1_2 = [cos(q2) -sin(q2) 0 0.210*cos(q2); sin(q2) cos(q2) 0 0.210*sin(q2); 0 0 1 0; 0 0 0 1]
  % Matriz de transformação do 2_3
 T_2_3 = [cos(q3) 0 -sin(q3) -0.075*cos(q3); sin(q3) 0 -cos(q3) -0.075*sin(q3); 0 -1 0 0; 0 0 0 1]
  % Matriz de transformação do 3_4
 T_3_4 = [cos(q4) 0 -sin(q4) 0; sin(q4) 0 -cos(q4) 0;0 1 0 0.210; 0 0 0 1]
  % Matriz de transformação do 4_5
 T_4_5 = [cos(q5) 0 -sin(q5) 0; sin(q5) 0 -cos(q5) 0; 0 -1 0 0; 0 0 0 1]
  % Matriz de transformação do 5_6
 T_5_6 = [cos(q6) -sin(q6) 0 0; sin(q6) cos(q6) 0  0; 0 0 0 0.07; 0 0 0 1]

  % Matriz de transformação do 0_6
 T_0_6 = T_0_1 * T_1_2 * T_2_3 * T_3_4 * T_4_5 * T_5_6
```

```
T_0_1 =

[ cos(q1), 0,  sin(q1),   0]
[ sin(q1), 0, -cos(q1),   0]
[       0, 1,        0, 1/8]
[       0, 0,        0,   1]


T_1_2 =

[ cos(q2), -sin(q2), 0, (21*cos(q2))/100]
[ sin(q2),  cos(q2), 0, (21*sin(q2))/100]
[       0,        0, 1,                0]
[       0,        0, 0,                1]


T_2_3 =

[ cos(q3),  0, -sin(q3), -(3*cos(q3))/40]
[ sin(q3),  0, -cos(q3), -(3*sin(q3))/40]
[       0, -1,        0,               0]
[       0,  0,        0,               1]


T_3_4 =

[ cos(q4), 0, -sin(q4),      0]
[ sin(q4), 0, -cos(q4),      0]
[       0, 1,        0, 21/100]
[       0, 0,        0,      1]


T_4_5 =

[ cos(q5),  0, -sin(q5), 0]
[ sin(q5),  0, -cos(q5), 0]
[       0, -1,        0, 0]
[       0,  0,        0, 1]


T_5_6 =

[ cos(q6), -sin(q6), 0,     0]
[ sin(q6),  cos(q6), 0,     0]
[       0,        0, 0, 7/100]
[       0,        0, 0,     1]


T_0_6 =

[ - sin(q6)*(cos(q4)*sin(q1) - sin(q4)*(cos(q1)*cos(q2)*cos(q3) - cos(q1)*sin(q2)*sin(q3))) - cos(q6)*(cos(q5)*(sin(q1)*sin(q4) - cos(q4)*(cos(q1)*cos(q2)*cos
[   sin(q6)*(cos(q1)*cos(q4) - sin(q4)*(sin(q1)*sin(q2)*sin(q3) - cos(q2)*cos(q3)*sin(q1))) + cos(q6)*(cos(q5)*(cos(q1)*sin(q4) - cos(q4)*(sin(q1)*sin(q2)*sin
[                                                                                                                                 sin(q4)*sin(q6)*(cos(q2)*sin(q3) + cos(q3)*sin(q2)) - cos(q6)*(sin
[
```

Primeiro, a função Link é usada para criar cada elo. Os parâmetros da função Link são exatamente os parâmetros de Denavit-Hartemberg do elo em questão, nesta ordem: theta = ângulo da junta (rad) d = deslocamento do elo (m) a = comprimento do elo (m) alpha = torção do elo (rad) sigma = tipo de junta (0: rotativa ou 1: prismática)

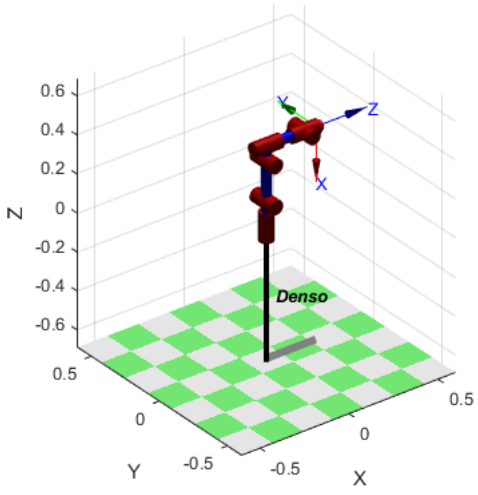
```
%Criação dos Links para o braço de 6-juntas

L(1) = Revolute('a', 0, 'alpha', pi/2, 'd', 0.125, 'qlim', [-2.79, 2.79]);
L(2) = Revolute('a', 0.21, 'alpha', 0, 'd', 0, 'offset', pi/2, 'qlim', [-2.09, 2.09]);
L(3) = Revolute('a', -0.075, 'alpha', -pi/2, 'd', 0, 'offset', -pi/2, 'qlim', [0.33, 2.79]);
L(4) = Revolute('a', 0, 'alpha', pi/2, 'd', 0.21, 'qlim', [-2.79, 2.79]);
L(5) = Revolute('a', 0, 'alpha', -pi/2, 'd', 0, 'qlim', [-2.09, 2.09]);
L(6) = Revolute('a', 0, 'alpha', 0, 'd', 0.07, 'qlim', [-6.28, 6.28]);

robot = SerialLink(L, 'name', 'Denso')
% Config inicial para o Denso
q = [0 0 -pi/2 0 0 0];

% Matriz de transformação por cinemática direta(Config Inicial)
Kd = robot.fkine(q)
% q()s gerados por cinemática inversa(Config Inicial)
Ki = robot.ikine(Kd)

%Ponto teste 1
t2 = transl(0.398, 0.0, 0.05);
t2(1:3,1:3) = roty(90);
%Ponto teste 2
t3 = transl(0.326,-0.230, -0.088);
t3(1:3,1:3) = roty(90);
%Cinematica inversa
Ki2 = robot.ikine(t2);
Ki3 = robot.ikine(t3);
robot.plot(Ki);
pause(3)
robot.plot(Ki2);
pause(3)
robot.plot(Ki);
pause(3)
robot.plot(Ki3);

%robot.teach()
```

```
robot =

Denso:: 6 axis, RRRRRR, stdDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|      0.125|          0|     1.5708|          0|
|  2|         q2|          0|       0.21|          0|     1.5708|
|  3|         q3|          0|     -0.075|    -1.5708|    -1.5708|
|  4|         q4|       0.21|          0|     1.5708|          0|
|  5|         q5|          0|          0|    -1.5708|          0|
|  6|         q6|       0.07|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+


Kd =
         0         0         1      0.28
         0         1         0         0
        -1         0         0      0.41
         0         0         0         1

Ki =

   -0.0000   -0.0000   -1.5708    0.0000   -0.0000   -0.0000
```



Essas trajetórias serão enviadas para o denso como angulos por cinemática inversa

```
t = 7;
traj = jtraj(Ki, Ki2, t);
traj2 = jtraj(Ki2, Ki, t);

% Aqui vai entrar a coordenada que o usr inserir
traj3 = jtraj(Ki, Ki3, t);
traj4 = jtraj(Ki3, Ki, t);
%Kd
%Kd2
%traj
%Trajetória inicial=>cubo
for i = 1:1:7
    [traj(i,1) traj(i,2) traj(i,3) traj(i,4) traj(i,5) traj(i,6)];
    robot.plot([traj(i,1) traj(i,2) traj(i,3) traj(i,4) traj(i,5) traj(i,6)]);
    pause(1);
end
%Trajetória cubo=>inicial
for i = 1:1:7
    [traj2(i,1) traj2(i,2) traj2(i,3) traj2(i,4) traj2(i,5) traj2(i,6)];
    robot.plot([traj2(i,1) traj2(i,2) traj2(i,3) traj2(i,4) traj2(i,5) traj2(i,6)]);
    pause(1);
end
%Trajetória inicial=>destino
for i = 1:1:7
    [traj3(i,1) traj3(i,2) traj3(i,3) traj3(i,4) traj3(i,5) traj3(i,6)];
    robot.plot([traj3(i,1) traj3(i,2) traj3(i,3) traj3(i,4) traj3(i,5) traj3(i,6)]);
    pause(1);
end
%Trajetória destino=>inicial
for i = 1:1:7
    [traj4(i,1) traj4(i,2) traj4(i,3) traj4(i,4) traj4(i,5) traj4(i,6)];
    robot.plot([traj4(i,1) traj4(i,2) traj4(i,3) traj4(i,4) traj4(i,5) traj4(i,6)]);
    pause(1);
end
```



Published with MATLAB® R2015a