```
clear, clc

syms px py pz az ax ay
syms q1 q2 q3

L1 = 12
L2 = L1
L3 = L1

DHParam = [q1 L1 0 0; -pi/2 q2 0 -pi/2; pi/2 q3 0 0];

[A, T, Q, Rot, Tra] = DH(DHParam, px, py, pz, az);

    EqX = px == (Tra(1));
    EqY = py == (Tra(2));
    EqZ = pz == (Tra(3));

    Eqs =  [EqX; EqY; EqZ];
    % n, s, a
    %EqAux = az == Rot(2,3);
    %EqAux = az == Rot(3,3);

    EqAux = az == Rot(3,1);

    jointVar = symvar(T);

Q = solve([EqX EqY EqZ], jointVar, "Real",true)%, "PrincipalValue",false)

%%%%%%%%5
```

```
L1 =

    12


L2 =

    12


L3 =

    12

     3

[q1, 12, 0, 0]

[-pi/2, q2, 0, -pi/2]

[pi/2, q3, 0, 0]
```

```
(:,:,1) =

[cos(q1), -sin(q1), 0,  0]
[sin(q1),  cos(q1), 0,  0]
[      0,        0, 1, 12]
[      0,        0, 0,  1]

(:,:,2) =

[      0,        0, 1,  0]
[     -1,        0, 0,  0]
[      0,       -1, 0, q2]
[      0,        0, 0,  1]

(:,:,3) =

[      0,       -1, 0,  0]
[      1,        0, 0,  0]
[      0,        0, 1, q3]
[      0,        0, 0,  1]

[ 0, -sin(q1), cos(q1), q3*cos(q1)]
[ 0,  cos(q1), sin(q1), q3*sin(q1)]
[-1,        0,       0,    q2 + 12]
[ 0,        0,       0,          1]

Warning: Solutions are only valid under certain conditions. To include
parameters and conditions in the solution, specify the 'ReturnConditions'
 value
as 'true'.

Q =

  struct with fields:

    q1: [2×1 sym]
    q2: [2×1 sym]
    q3: [2×1 sym]
```

assume(q1>-pi & q1<pi & q2>0 & q3>0)

```
   EqX = px == expand(Tra(1));
   EqY = py == expand(Tra(2));
   EqZ = pz == expand(Tra(3));

   EqAux = az == Rot(3,1);
   EqAux2 = ay == Rot(2,1);
   EqAux3 = ax == Rot(1,1);

   jointVar = symvar(T);

   %Q = solve([EqX EqY EqZ EqAux1 EqAux1b EqAux2 EqAux2b EqAux3 EqAux3b], jointVar, "Real"
```

```
    Q = solve([EqX EqY EqZ EqAux EqAux2 EqAux3], jointVar(2:end), "Real",true) % Se puede j
    Q2 = solve([EqX EqY EqZ EqAux EqAux2 EqAux3], jointVar, "Real",true) % Se puede jugar c

     %Q.q1 = solve

     %%%%%%%%%%55

% A es el arreglo conteniendo las matrices de transformación de i = 1 a i =
% n. T es la matriz de transformación simplificada, la matriz de cinemática
% directa. Q es un vector conteniendo los resultados para las variables de
% q1 a qn. Rot es la matriz de rotación extraída de T. Y Tra es el ventor
% posición extraído de T.

% syms n_x s_x a_x p_x n_y s_y a_y p_y n_z s_z a_z p_z
% T_syms = [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];
%
% T1 = (A(:,:,1))\T_syms == A(:,:,2)*A(:,:,3)




%inverseKinematics()

%     ia = inv(A(:, :, 1));
%     ter2 = A(:, :, 2) * A(:, :, 3);
%     eq1 = ter2(3,4) == ia (3,1)*px + ia(3,2) *py;
%     %q1 = solve(eq1,q1, 'real',true);
%
%     ib = inv(A(:, :, 2))*ia;
%     ter1 = A(:, :, 3);
%
%     eq2 = ter1(2,4) == ib(2,1)*px + ib(2,2)*py + ib(2,3)*pz + ib(2,4)*1;
%     eq3 = ter1(1,4) == ib(1,1)*px + ib(1,2)*py + ib(1,3)*pz + ib(1,4)*1;
%
%     [q2,q3] = solve([eq2,eq3],[q2 q3], 'real', true);
%
%     q1 = atan2(py,px);
%
%     q2 = eval(q2(1));
%     q3 = eval(q3(1));
%
%
%     th = [q1,q2,q3];
%     th2 = rad2deg(th);
%
%     th2(1)
%     th2(2)
%     th2(3)
```

Published with MATLAB® R2022b