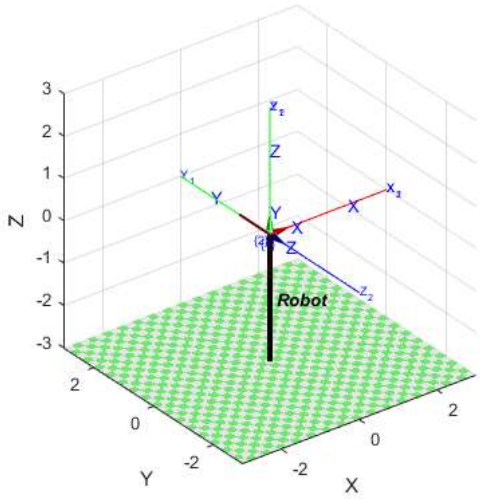
```
%Punto 2

clear all
clc

q1=0;
q2=0;

P2=[0    0      q1    0    0    1;
    0    pi/2   q2    0    0    1];


[Robot,L,A0N] = ForKin(P2);

fprintf('\n\n Evaluacion de A0N con vector  de variables Q = [q1 q2 ... qn] \n')
Q = zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q, 3, 3, 7)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%%pause ()
```




```
%Punto 3

clear all
clc

q1=0;
q2=0;
q3=0;

P3=[0    0       0     q1    0    0;
    0    pi/2    q2    0     0    1;
    0    pi/2    0     q3    0    0];


[Robot,L,A0N] = ForKin(P3);

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q = zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q, 3, 3, 7)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%pause ()
```

```
Robot =

Robot (3 axis, RPR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          0|          0|          0|          0|
|  2|          0|         q2|          0|      1.571|          0|
|  3|         q3|          0|          0|      1.571|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1
```

A01 =

```
[ cos(q1), -1.0*sin(q1),   0,   0]
[ sin(q1),      cos(q1),   0,   0]
[       0,            0, 1.0,   0]
[       0,            0,   0, 1.0]
```

A12 =

```
[ 1.0,   0,    0,       0]
[   0,   0, -1.0, -1.0*q2]
[   0, 1.0,    0,       0]
[   0,   0,    0,     1.0]
```

A23 =

```
[ cos(q3), -1.0*sin(q3),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q3),      cos(q3),    0,   0]
[       0,            0,    0, 1.0]
```

A03 =

```
[ cos(q1 - 1.0*q3),      sin(q1 - 1.0*q3),    0,      q2*sin(q1)]
[ sin(q1 - 1.0*q3), -1.0*cos(q1 - 1.0*q3),    0, -1.0*q2*cos(q1)]
[                0,                     0, -1.0,              0]
[                0,                     0,    0,            1.0]
```

Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

```
      q2*sin(q1)
 -1.0*q2*cos(q1)
               0
```

Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

0

r11 =

cos(q1 - 1.0*q3)

r21 =

sin(q1 - 1.0*q3)

r32 =

0

r33 =

-1.0

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))

Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

0

r23 =

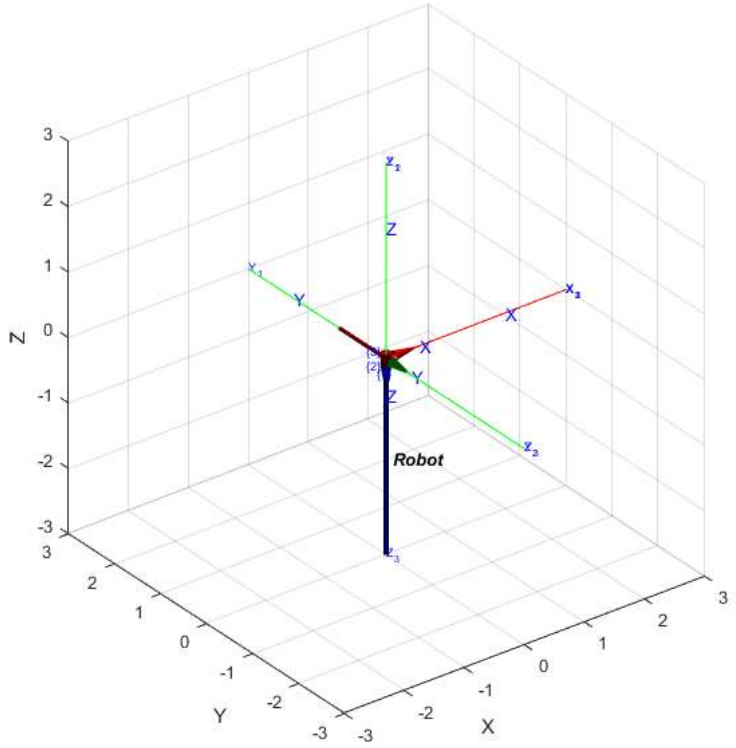
0

r33 =

-1.0

r12 =

```
sin(q1 - 1.0*q3)


r11 =

cos(q1 - 1.0*q3)

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))


 Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn]

Q =

     0     0     0


T =

     1     0     0     0
     0    -1     0     0
     0     0    -1     0
     0     0     0     1


Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[ 0, 0, 0]


Matriz de rotación:

r =

     1     0     0
     0    -1     0
     0     0    -1


Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
    3.1416         0         0


Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
   -3.1416         0         0


Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
         0    3.1416    3.1416

PULSA CUALQUIER TECLA PARA CONTINUAR
```




```
%Punto 4

clear all
clc

q1=0;
q2=0;
q3=0;
q4=0;
q5=0;

P4=[0           0      1          q1    0      0;
    0           pi/2   q2         0     1      1;
    0           pi/2   1          q3    pi/2   0;
    0          -pi/2   0          q4    pi/4   0;
    sqrt(2)/2   pi/2   sqrt(2)/2  q5   -pi/2   0;
    0          -pi/2   sqrt(2)    0     0     -1];


[Robot,L,A0N] = ForKin(P4);

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q = zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q, 5, 3, 8)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%pause ()
```

```
Robot =

Robot (5 axis, RPRRR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          1|          0|          0|          0|
|  2|          0|         q2|          0|      1.571|          1|
|  3|         q3|          1|          0|      1.571|      1.571|
|  4|         q4|          0|          0|     -1.571|     0.7854|
|  5|         q5|     0.7071|     0.7071|      1.571|     -1.571|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1          0          0          0
          0          0  1  0  0           0          0          1     1.4142
       9.81          0  0  1  0           0         -1          0          0
                     0  0  0  1           0          0          0          1


A01 =

[ cos(q1), -1.0*sin(q1),   0,   0]
```

```
[ sin(q1),      cos(q1),   0,   0]
[       0,            0, 1.0, 1.0]
[       0,            0,   0, 1.0]
```

A12 =

```
[ 1.0,   0,    0,              0]
[   0,   0, -1.0, - 1.0*q2 - 1.0]
[   0, 1.0,    0,              0]
[   0,   0,    0,            1.0]
```

A23 =

```
[ -1.0*sin(q3), -1.0*cos(q3),    0,    0]
[            0,            0, -1.0, -1.0]
[      cos(q3), -1.0*sin(q3),    0,    0]
[            0,           0,    0,  1.0]
```

A34 =

```
[      cos(q4 + 0.7854), -1.0*sin(q4 + 0.7854),   0,   0]
[                     0,                     0, 1.0,   0]
[ -1.0*sin(q4 + 0.7854), -1.0*cos(q4 + 0.7854),   0,   0]
[                     0,                     0,   0, 1.0]
```

A45 =

```
[      sin(q5), cos(q5),    0,  0.70711]
[            0,       0, -1.0, -0.70711]
[ -1.0*cos(q5), sin(q5),    0,        0]
[            0,       0,    0,      1.0]
```

A56 =

```
[ 1.0,    0,   0,      0]
[   0,    0, 1.0, 1.4142]
[   0, -1.0,   0,      0]
[   0,    0,   0,    1.0]
```

A06 =

```
[     cos(q1 - 1.0*q3)*cos(q5) + sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*sin(q5), -1.0*sin(q1 - 1.0*q3)*sin(q4 + 0.7854),       sin(q1 - 1.0*q3)*cos(q4 +
0.7854)*cos(q5) - 1.0*cos(q1 - 1.0*q3)*sin(q5),   0.70711*sin(q1 - 1.0*q3)*cos(q4 + 0.7854) + 1.0*sin(q1)*(q2 + 1.0) + 0.70711*sin(q1 - 1.0*q3)*sin(
q4 + 0.7854) - 1.4142*cos(q1 - 1.0*q3)*sin(q5) + 1.4142*sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5)]
[ sin(q1 - 1.0*q3)*cos(q5) - 1.0*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*sin(q5),      cos(q1 - 1.0*q3)*sin(q4 + 0.7854), - 1.0*sin(q1 - 1.0*q3)*sin(q5) -
 1.0*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5), - 0.70711*cos(q1 - 1.0*q3)*cos(q4 + 0.7854) - 1.0*cos(q1)*(q2 + 1.0) - 0.70711*cos(q1 - 1.0*q3)*sin(
q4 + 0.7854) - 1.4142*sin(q1 - 1.0*q3)*sin(q5) - 1.4142*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5)]
[                                          sin(q4 + 0.7854)*sin(q5),                      cos(q4 + 0.7854),
                    sin(q4 + 0.7854)*cos(q5),
                                      1.4142*sin(q4 + 0.7854)*cos(q5) - 1.0*cos(q4 + 1.5708)]
[                                                 0,                                     0,
                                                  0,
                                                                                      1.0]
```

Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

```
   0.70711*sin(q1 - 1.0*q3)*cos(q4 + 0.7854) + 1.0*sin(q1)*(q2 + 1.0) + 0.70711*sin(q1 - 1.0*q3)*sin(q4 + 0.7854) - 1.4142*cos(q1 - 1.0*q3)*sin(q5)
+ 1.4142*sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5)
 - 0.70711*cos(q1 - 1.0*q3)*cos(q4 + 0.7854) - 1.0*cos(q1)*(q2 + 1.0) - 0.70711*cos(q1 - 1.0*q3)*sin(q4 + 0.7854) - 1.4142*sin(q1 - 1.0*q3)*sin(q5)
- 1.4142*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5)
                                                                                                                                                1.41
42*sin(q4 + 0.7854)*cos(q5) - 1.0*cos(q4 + 1.5708)
```

Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

```
sin(q4 + 0.7854)*sin(q5)
```

r11 =

```
cos(q1 - 1.0*q3)*cos(q5) + sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*sin(q5)
```

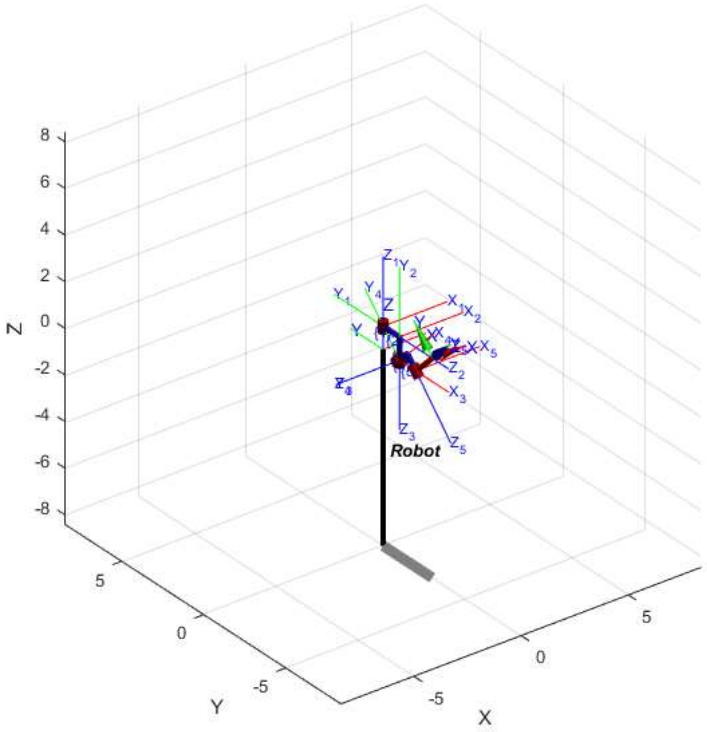
r21 =

```
sin(q1 - 1.0*q3)*cos(q5) - 1.0*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*sin(q5)
```

r32 =

```
cos(q4 + 0.7854)
```

```
r33 =

sin(q4 + 0.7854)*cos(q5)

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))


Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5) - 1.0*cos(q1 - 1.0*q3)*sin(q5)


r23 =

- 1.0*sin(q1 - 1.0*q3)*sin(q5) - 1.0*cos(q1 - 1.0*q3)*cos(q4 + 0.7854)*cos(q5)


r33 =

sin(q4 + 0.7854)*cos(q5)


r12 =

-1.0*sin(q1 - 1.0*q3)*sin(q4 + 0.7854)


r11 =

cos(q1 - 1.0*q3)*cos(q5) + sin(q1 - 1.0*q3)*cos(q4 + 0.7854)*sin(q5)

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))


 Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn]

Q =

     0     0     0     0     0


T =

    1.0000         0         0         0
         0    0.7071   -0.7071   -3.0000
         0    0.7071    0.7071    1.0000
         0         0         0    1.0000


Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[ 0, -3.0, 1.0]


Matriz de rotación:

r =

    1.0000         0         0
         0    0.7071   -0.7071
         0    0.7071    0.7071


Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
    0.7854         0         0


Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
    0.7854         0         0


Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
   -1.5708    0.7854    1.5708

PULSA CUALQUIER TECLA PARA CONTINUAR
```




```
%Punto 5

clear all
clc

syms L1 L2 L3 L4 as real
syms q1 q2 q3 q4 q5 q6 as real


P5=[0    0      L1     q1   0     0;
    0    pi/2   0      q2   0     0;
    L2   0      0      q3   pi/2  0;
    0    pi/2   L3     q4   pi/2  0;
    0    pi/2   0      q5   0     0;
    0   -pi/2   0      q6   0     0;
    0    0      L4     0    0    -1];


[Robot,L,A0N] = ForKin(P5);

q1=0;
q2=0;
q3=0;
q4=0;
q5=0;
q6=0;

L1=40;
L2=30;
L3=50;
L4=30;

P5=eval(P5);

[Robot,L,A0N] = ForKin(P5);

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q= zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q, 5, 20, 10)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%pause ()

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q=[pi/6, -pi/6, -pi/12, -pi/2, pi, pi/2]
evaluar(A0N, Q);
graficar (Robot,L, Q, 5, 20, 10)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%pause ()
```

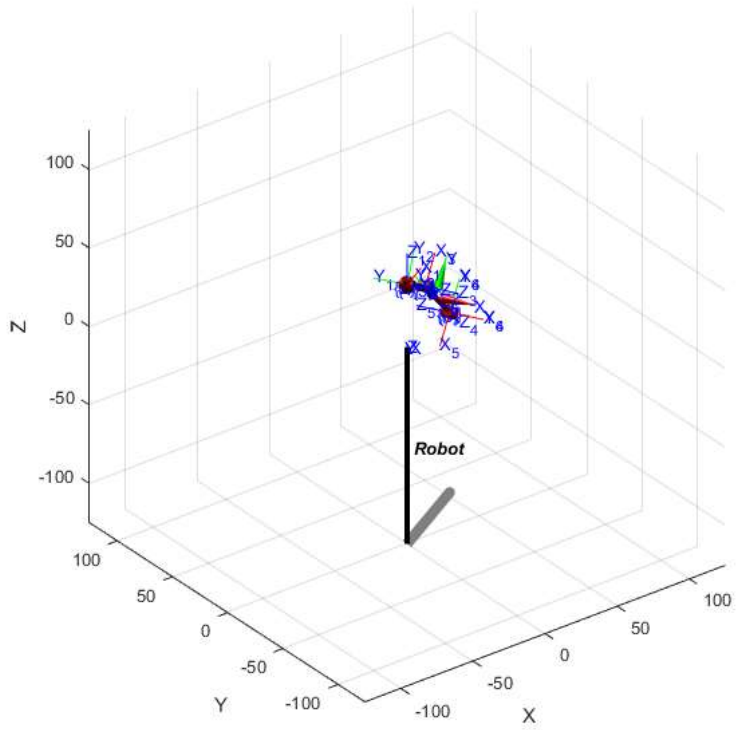
```
Robot =

Robot (6 axis, RRRRRR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|         L1|          0|          0|          0|
|  2|         q2|          0|          0|       pi/2|          0|
|  3|         q3|          0|         L2|          0|       pi/2|
|  4|         q4|         L3|          0|       pi/2|       pi/2|
|  5|         q5|          0|          0|       pi/2|          0|
|  6|         q6|          0|          0|      -pi/2|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =     0  base = 1  0  0  0   tool =   1  0  0  0
          0         0  1  0  0            0  1  0  0
       9.81         0  0  1  0            0  0  1  L4
                    0  0  0  1            0  0  0  1


A01 =

[ cos(q1), -1.0*sin(q1),   0,   0]
[ sin(q1),      cos(q1),   0,   0]
[       0,            0, 1.0,  L1]
[       0,            0,   0, 1.0]


A12 =

[ cos(q2), -1.0*sin(q2),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q2),      cos(q2),    0,   0]
[       0,            0,    0, 1.0]


A23 =

[ -1.0*sin(q3), -1.0*cos(q3),   0,  L2]
[      cos(q3), -1.0*sin(q3),   0,   0]
[            0,            0, 1.0,   0]
[            0,            0,   0, 1.0]


A34 =

[ -1.0*sin(q4), -1.0*cos(q4),    0,       0]
[            0,            0, -1.0, -1.0*L3]
[      cos(q4), -1.0*sin(q4),    0,       0]
[            0,            0,    0,     1.0]


A45 =

[ cos(q5), -1.0*sin(q5),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q5),      cos(q5),    0,   0]
[       0,            0,    0, 1.0]


A56 =

[      cos(q6), -1.0*sin(q6),   0,   0]
[            0,            0, 1.0,   0]
[ -1.0*sin(q6), -1.0*cos(q6),   0,   0]
[            0,            0,   0, 1.0]


A67 =

[ 1.0,   0,   0,   0]
[   0, 1.0,   0,   0]
[   0,   0, 1.0,  L4]
[   0,   0,   0, 1.0]


A07 =

[                          cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*s
in(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)),                   - 1.0*cos(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)) - 1.0*sin(q6)*(cos(
q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)),         1.0*cos(q2 + q3)*cos(q1)*cos(q5) - 1.0*sin(q5)*(cos(q4
)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)),     cos(q1)*(L3*cos(q2 + q3) + L2*cos(q2)) - 1.0*L4*(1.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*
sin(q4)) - 1.0*cos(q2 + q3)*cos(q1)*cos(q5))]
[ 1.0*sin(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)) - 1.0*cos(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*si
n(q4)) - 1.0*cos(q2 + q3)*sin(q1)*sin(q5)), 1.0*sin(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) - 1.0*cos(q2 + q3)*sin
(q1)*sin(q5)) + 1.0*cos(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)), 1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*si
n(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q1), L4*(1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q
1)) + sin(q1)*(L3*cos(q2 + q3) + L2*cos(q2))]
[                                                                                                  cos(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*s
in(q4)) - 1.0*cos(q2 + q3)*cos(q4)*sin(q6),                                                                                              - 1.0*sin(q6)*(sin(q2
+ q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*cos(q6),                                                                sin(q2 + q
3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5),                                                        L1 + L3*sin(q2 + q3) + L2*sin(q2) + L4*(sin(q2 +
 q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5))]
[
                                          0,
                                                                                     0,
```

```
                                        0,
                                         1.0]


Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

     cos(q1)*(L3*cos(q2 + q3) + L2*cos(q2)) - 1.0*L4*(1.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q1)*cos(q5
))
 L4*(1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q1)) + sin(q1)*(L3*cos(q2 + q3) + L2*cos(q2
))
                                                                 L1 + L3*sin(q2 + q3) + L2*sin(q2) + L4*(sin(q2 + q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5
))



Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

cos(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*sin(q6)


r11 =

cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*
cos(q1)*cos(q4))


r21 =

1.0*sin(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)) - 1.0*cos(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(
q4)) - 1.0*cos(q2 + q3)*sin(q1)*sin(q5))


r32 =

- 1.0*sin(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*cos(q6)


r33 =

sin(q2 + q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5)

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

1.0*cos(q2 + q3)*cos(q1)*cos(q5) - 1.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4))


r23 =

1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q1)


r33 =

sin(q2 + q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5)


r12 =

- 1.0*cos(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)) - 1.0*sin(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2
+ q3)*cos(q1)*sin(q5))


r11 =

cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*
cos(q1)*cos(q4))

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))


Robot =

Robot (6 axis, RRRRRR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|         40|          0|          0|          0|
```

```
|  2|         q2|          0|          0|      1.571|          0|
|  3|         q3|          0|         30|          0|      1.571|
|  4|         q4|         50|          0|      1.571|      1.571|
|  5|         q5|          0|          0|      1.571|          0|
|  6|         q6|          0|          0|     -1.571|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =     0  base = 1  0  0  0   tool =  1   0   0   0
           0          0  1  0  0           0   1   0   0
        9.81          0  0  1  0           0   0   1  30
                      0  0  0  1           0   0   0   1


A01 =

[ cos(q1), -1.0*sin(q1),   0,    0]
[ sin(q1),      cos(q1),   0,    0]
[       0,            0, 1.0, 40.0]
[       0,            0,   0,  1.0]


A12 =

[ cos(q2), -1.0*sin(q2),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q2),      cos(q2),    0,   0]
[       0,            0,    0, 1.0]


A23 =

[ -1.0*sin(q3), -1.0*cos(q3),   0, 30.0]
[      cos(q3), -1.0*sin(q3),   0,    0]
[            0,            0, 1.0,    0]
[            0,            0,   0,  1.0]


A34 =

[ -1.0*sin(q4), -1.0*cos(q4),    0,     0]
[            0,            0, -1.0, -50.0]
[      cos(q4), -1.0*sin(q4),    0,     0]
[            0,            0,    0,   1.0]


A45 =

[ cos(q5), -1.0*sin(q5),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q5),      cos(q5),    0,   0]
[       0,            0,    0, 1.0]


A56 =

[      cos(q6), -1.0*sin(q6),   0,   0]
[            0,            0, 1.0,   0]
[ -1.0*sin(q6), -1.0*cos(q6),   0,   0]
[            0,            0,   0, 1.0]


A67 =

[ 1.0,   0,   0,    0]
[   0, 1.0,   0,    0]
[   0,   0, 1.0, 30.0]
[   0,   0,   0,  1.0]


A07 =

[                          cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*s
in(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)),                   - 1.0*cos(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)) - 1.0*sin(q6)*(cos(
q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)),         1.0*cos(q2 + q3)*cos(q1)*cos(q5) - 1.0*sin(q5)*(cos(q4
)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)),         10.0*cos(q1)*(5.0*cos(q2 + q3) + 3.0*cos(q2)) - 30.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(
q1)*sin(q4)) + 30.0*cos(q2 + q3)*cos(q1)*cos(q5)]
[ 1.0*sin(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)) - 1.0*cos(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*si
n(q4)) - 1.0*cos(q2 + q3)*sin(q1)*sin(q5)), 1.0*sin(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) - 1.0*cos(q2 + q3)*sin
(q1)*sin(q5)) + 1.0*cos(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)), 1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*si
n(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q1), 30.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 10.0*sin(q1)*(5.0*cos(q2 + q3) +
3.0*cos(q2)) + 30.0*cos(q2 + q3)*cos(q5)*sin(q1)]
[                                                                                                  cos(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*s
in(q4)) - 1.0*cos(q2 + q3)*cos(q4)*sin(q6),                                                                                          - 1.0*sin(q6)*(sin(q2
+ q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*cos(q6),                                                               sin(q2 + q
3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5),                                                     50.0*sin(q2 + q3) + 30.0*sin(q2) + 30.0*sin(q2 + q3)*co
s(q5) + 30.0*cos(q2 + q3)*sin(q4)*sin(q5) + 40.0]
[
                                        0,
                                                                                0,
                                       0,
                                            1.0]


Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =
```

```
         10.0*cos(q1)*(5.0*cos(q2 + q3) + 3.0*cos(q2)) - 30.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + 30.0*cos(q2 + q3)*cos(q1)*c
os(q5)
 30.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 10.0*sin(q1)*(5.0*cos(q2 + q3) + 3.0*cos(q2)) + 30.0*cos(q2 + q3)*cos(q5)*s
in(q1)
                                                    50.0*sin(q2 + q3) + 30.0*sin(q2) + 30.0*sin(q2 + q3)*cos(q5) + 30.0*cos(q2 + q3)*sin(q4)*sin(q5)
+ 40.0



Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

cos(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*sin(q6)


r11 =

cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*
cos(q1)*cos(q4))


r21 =

1.0*sin(q6)*(1.0*cos(q1)*sin(q4) + 1.0*sin(q2 + q3)*cos(q4)*sin(q1)) - 1.0*cos(q6)*(1.0*cos(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(
q4)) - 1.0*cos(q2 + q3)*sin(q1)*sin(q5))


r32 =

- 1.0*sin(q6)*(sin(q2 + q3)*sin(q5) - 1.0*cos(q2 + q3)*cos(q5)*sin(q4)) - 1.0*cos(q2 + q3)*cos(q4)*cos(q6)


r33 =

sin(q2 + q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5)

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

1.0*cos(q2 + q3)*cos(q1)*cos(q5) - 1.0*sin(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4))


r23 =

1.0*sin(q5)*(1.0*cos(q1)*cos(q4) - 1.0*sin(q2 + q3)*sin(q1)*sin(q4)) + 1.0*cos(q2 + q3)*cos(q5)*sin(q1)


r33 =

sin(q2 + q3)*cos(q5) + cos(q2 + q3)*sin(q4)*sin(q5)


r12 =

- 1.0*cos(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*cos(q1)*cos(q4)) - 1.0*sin(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2
+ q3)*cos(q1)*sin(q5))


r11 =

cos(q6)*(cos(q5)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q2 + q3)*cos(q1)*sin(q5)) - 1.0*sin(q6)*(sin(q1)*sin(q4) - 1.0*sin(q2 + q3)*
cos(q1)*cos(q4))

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))



 Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn]

Q =

     0     0     0     0     0     0


T =

     0     0     1   110
    -1     0     0     0
     0    -1     0    40
     0     0     0     1
```

```
Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[ 110.0, 0, 40.0]



Matriz de rotación:

r =

     0     0     1
    -1     0     0
     0    -1     0



Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
   -1.5708         0   -1.5708



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
         0    1.5708   -1.5708



Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
         0    1.5708   -1.5708

PULSA CUALQUIER TECLA PARA CONTINUAR


 Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn]

Q =

    0.5236   -0.5236   -0.2618   -1.5708    3.1416    1.5708


T =

    0.5000    0.6124   -0.6124   34.7474
   -0.8660    0.3536   -0.3536   20.0614
   -0.0000    0.7071    0.7071   10.8579
         0         0         0    1.0000



Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[ 34.747, 20.061, 10.858]



Matriz de rotación:

r =

    0.5000    0.6124   -0.6124
   -0.8660    0.3536   -0.3536
   -0.0000    0.7071    0.7071



Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
    0.7854    0.0000   -1.0472



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
    0.4636   -0.6591   -0.8861



Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
   -2.6180    0.7854    1.5708

PULSA CUALQUIER TECLA PARA CONTINUAR
```




```
%Punto 6

clear all
clc

syms L1 L2 L3 L4 L5 as real
syms q1 q2 q3 q4 q5 q6 as real

P6=[0    0      L1     q1    0     0;
    0   -pi/2   L2     q2   -pi/2  0;
    0   -pi/2   L3     q3    pi/2  0;
    L2   pi/2   0      q4    pi/2  0;
    L4   pi/2   q5     pi/2  L5    1;
    0    pi/2   0      q6    0     0;];

[Robot,L,A0N] = ForKin(P6);

q1=0;
q2=0;
q3=0;
q4=0;
q5=0;
q6=0;

L1=10;
L2=5;
L3=10;
L4=5;
L5=5;

P6=eval(P6);

[Robot,L,A0N] = ForKin(P6);

[N,~]=size(P6);

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q=zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q , 0, 20,10)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
%%pause ()
```

```
Robot =

Robot (6 axis, RRRRPR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|         L1|          0|          0|          0|
```

```
|  2|         q2|         L2|          0|      -pi/2|      -pi/2|
|  3|         q3|         L3|          0|      -pi/2|       pi/2|
|  4|         q4|          0|         L2|       pi/2|       pi/2|
|  5|       pi/2|         q5|         L4|       pi/2|         L5|
|  6|         q6|          0|          0|       pi/2|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1


A01 =

[ cos(q1), -1.0*sin(q1),   0,   0]
[ sin(q1),      cos(q1),   0,   0]
[       0,            0, 1.0,  L1]
[       0,            0,   0, 1.0]


A12 =

[ sin(q2),      cos(q2),   0,   0]
[       0,            0, 1.0,  L2]
[ cos(q2), -1.0*sin(q2),   0,   0]
[       0,            0,   0, 1.0]


A23 =

[ -1.0*sin(q3), -1.0*cos(q3),   0,   0]
[            0,            0, 1.0,  L3]
[ -1.0*cos(q3),      sin(q3),   0,   0]
[            0,            0,   0, 1.0]


A34 =

[ -1.0*sin(q4), -1.0*cos(q4),    0,  L2]
[            0,            0, -1.0,   0]
[      cos(q4), -1.0*sin(q4),    0,   0]
[            0,            0,    0, 1.0]


A45 =

[   0, -1.0,    0,                L4]
[   0,    0, -1.0, - 1.0*L5 - 1.0*q5]
[ 1.0,    0,    0,                 0]
[   0,    0,    0,               1.0]


A56 =

[ cos(q6), -1.0*sin(q6),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q6),      cos(q6),    0,   0]
[       0,            0,    0, 1.0]


A06 =

[                      cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) +
 1.0*cos(q1)*cos(q2)*sin(q4)),                 cos(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) -
 1.0*sin(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)),     1.0*cos(q1)*cos(q2)*cos(q4) - 1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q
3)),         1.0*L2*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) - 1.0*L4*(1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) - 1.0*cos(
q1)*cos(q2)*cos(q4)) + 1.0*(1.0*L5 + 1.0*q5)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) - 1.0*L2*si
n(q1) + 1.0*L3*cos(q1)*cos(q2)]
[ - 1.0*cos(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*sin(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) -
 1.0*cos(q2)*sin(q1)*sin(q4)), 1.0*sin(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*cos(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0
*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q1)*sin(q4)), 1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + 1.0*cos(q2)*cos(q4)*sin(
q1), 1.0*L4*(1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + cos(q2)*cos(q4)*sin(q1)) + 1.0*L2*cos(q1) - 1.0*L2*(1.0*cos(q1)*cos(q
3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q1)*sin(q4))*(1.0*L5 + 1.
0*q5) + 1.0*L3*cos(q2)*sin(q1)]
[                                                                                                                                  cos(q2)*cos(q3)*cos(q6) - 1.0*sin(q6)*(1.0*sin(q2)*sin(q4) +
 1.0*cos(q2)*cos(q4)*sin(q3)),                                                                                                                                  - 1.0*cos(q6)*(1.0*sin(q2)*sin(q4) + 1.
0*cos(q2)*cos(q4)*sin(q3)) - 1.0*cos(q2)*cos(q3)*sin(q6),                                                                                                 cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(
q2),
 1.0*L1 - 1.0*(1.0*L5 + 1.0*q5)*(sin(q2)*sin(q4) + cos(q2)*cos(q4)*sin(q3)) - 1.0*L3*sin(q2) - 1.0*L4*(1.0*cos(q4)*sin(q2) - 1.0*cos(q2)*sin(q3)*sin
(q4)) - 1.0*L2*cos(q2)*sin(q3)]
[
                            0,
                                                    0,
  0,

                            1.0]



Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

        1.0*L2*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) - 1.0*L4*(1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) - 1.0*cos(q1)*
cos(q2)*cos(q4)) + 1.0*(1.0*L5 + 1.0*q5)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) - 1.0*L2*sin(q1
) + 1.0*L3*cos(q1)*cos(q2)
```

```
 1.0*L4*(1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + cos(q2)*cos(q4)*sin(q1)) + 1.0*L2*cos(q1) - 1.0*L2*(1.0*cos(q1)*cos(q3) +
 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q1)*sin(q4))*(1.0*L5 + 1.0*q5
) + 1.0*L3*cos(q2)*sin(q1)
                                                                                                                                                1.0
*L1 - 1.0*(1.0*L5 + 1.0*q5)*(sin(q2)*sin(q4) + cos(q2)*cos(q4)*sin(q3)) - 1.0*L3*sin(q2) - 1.0*L4*(1.0*cos(q4)*sin(q2) - 1.0*cos(q2)*sin(q3)*sin(q4)
) - 1.0*L2*cos(q2)*sin(q3)



Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

cos(q2)*cos(q3)*cos(q6) - 1.0*sin(q6)*(1.0*sin(q2)*sin(q4) + 1.0*cos(q2)*cos(q4)*sin(q3))


r11 =

cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin
(q4))


r21 =

- 1.0*cos(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*sin(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1
.0*cos(q2)*sin(q1)*sin(q4))


r32 =

- 1.0*cos(q6)*(1.0*sin(q2)*sin(q4) + 1.0*cos(q2)*cos(q4)*sin(q3)) - 1.0*cos(q2)*cos(q3)*sin(q6)


r33 =

cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(q2)

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

1.0*cos(q1)*cos(q2)*cos(q4) - 1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3))


r23 =

1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + 1.0*cos(q2)*cos(q4)*sin(q1)


r33 =

cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(q2)


r12 =

cos(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) - 1.0*sin(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)
*sin(q2))


r11 =

cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin
(q4))

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))


Robot =

Robot (6 axis, RRRRPR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|         10|          0|          0|          0|
|  2|         q2|          5|          0|     -1.571|     -1.571|
|  3|         q3|         10|          0|     -1.571|      1.571|
|  4|         q4|          0|          5|      1.571|      1.571|
|  5|      1.571|         q5|          5|      1.571|          5|
|  6|         q6|          0|          0|      1.571|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
```

```
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1


A01 =

[ cos(q1), -1.0*sin(q1),   0,    0]
[ sin(q1),      cos(q1),   0,    0]
[       0,            0, 1.0, 10.0]
[       0,            0,   0,  1.0]


A12 =

[ sin(q2),      cos(q2),   0,   0]
[       0,            0, 1.0, 5.0]
[ cos(q2), -1.0*sin(q2),   0,   0]
[       0,            0,   0, 1.0]


A23 =

[ -1.0*sin(q3), -1.0*cos(q3),   0,    0]
[            0,            0, 1.0, 10.0]
[ -1.0*cos(q3),      sin(q3),   0,    0]
[            0,            0,   0,  1.0]


A34 =

[ -1.0*sin(q4), -1.0*cos(q4),    0, 5.0]
[            0,            0, -1.0,   0]
[      cos(q4), -1.0*sin(q4),    0,   0]
[            0,            0,    0, 1.0]


A45 =

[   0, -1.0,    0,            5.0]
[   0,    0, -1.0, - 1.0*q5 - 5.0]
[ 1.0,    0,    0,              0]
[   0,    0,    0,            1.0]


A56 =

[ cos(q6), -1.0*sin(q6),    0,   0]
[       0,            0, -1.0,   0]
[ sin(q6),      cos(q6),    0,   0]
[       0,            0,    0, 1.0]


A06 =

[                      cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) +
 1.0*cos(q1)*cos(q2)*sin(q4)),                 cos(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) -
 1.0*sin(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)),     1.0*cos(q1)*cos(q2)*cos(q4) - 1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q
3)),         10.0*cos(q1)*cos(q2) - 5.0*sin(q1) + 5.0*cos(q3)*sin(q1) + 1.0*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q
1)*cos(q2)*sin(q4))*(1.0*q5 + 5.0) - 5.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 5.0*cos(q1)*cos(q2)*cos(q4) - 5.0*cos(q1)*sin(q2)
*sin(q3)]
[ - 1.0*cos(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*sin(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) -
 1.0*cos(q2)*sin(q1)*sin(q4)), 1.0*sin(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*cos(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0
*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q1)*sin(q4)), 1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + 1.0*cos(q2)*cos(q4)*sin(
q1), 5.0*cos(q1) - 5.0*cos(q1)*cos(q3) + 10.0*cos(q2)*sin(q1) - 1.0*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*s
in(q1)*sin(q4))*(1.0*q5 + 5.0) + 5.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 5.0*sin(q1)*sin(q2)*sin(q3) + 5.0*cos(q2)*cos(q4)
*sin(q1)]
[                                                                                                                                  cos(q2)*cos(q3)*cos(q6) - 1.0*sin(q6)*(1.0*sin(q2)*sin(q4) +
 1.0*cos(q2)*cos(q4)*sin(q3)),                                                                                                                - 1.0*cos(q6)*(1.0*sin(q2)*sin(q4) + 1.
0*cos(q2)*cos(q4)*sin(q3)) - 1.0*cos(q2)*cos(q3)*sin(q6),                                                                                  cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(
q2),
  5.0*cos(q2)*sin(q3)*sin(q4) - 5.0*cos(q2)*sin(q3) - 5.0*cos(q4)*sin(q2) - 1.0*(q5 + 5.0)*(sin(q2)*sin(q4) + cos(q2)*cos(q4)*sin(q3)) - 10.0*sin(q2
) + 10.0]
[
                            0,
                                                        0,
  0,

     1.0]



Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

         10.0*cos(q1)*cos(q2) - 5.0*sin(q1) + 5.0*cos(q3)*sin(q1) + 1.0*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*c
os(q2)*sin(q4))*(1.0*q5 + 5.0) - 5.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 5.0*cos(q1)*cos(q2)*cos(q4) - 5.0*cos(q1)*sin(q2)*sin
(q3)
 5.0*cos(q1) - 5.0*cos(q1)*cos(q3) + 10.0*cos(q2)*sin(q1) - 1.0*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q
1)*sin(q4))*(1.0*q5 + 5.0) + 5.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 5.0*sin(q1)*sin(q2)*sin(q3) + 5.0*cos(q2)*cos(q4)*sin
(q1)
                                                                                                                                                  5.
0*cos(q2)*sin(q3)*sin(q4) - 5.0*cos(q2)*sin(q3) - 5.0*cos(q4)*sin(q2) - 1.0*(q5 + 5.0)*(sin(q2)*sin(q4) + cos(q2)*cos(q4)*sin(q3)) - 10.0*sin(q2) +
10.0
```

Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

cos(q2)*cos(q3)*cos(q6) - 1.0*sin(q6)*(1.0*sin(q2)*sin(q4) + 1.0*cos(q2)*cos(q4)*sin(q3))

r11 =

cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4))

r21 =

- 1.0*cos(q6)*(1.0*cos(q1)*sin(q3) - 1.0*cos(q3)*sin(q1)*sin(q2)) - 1.0*sin(q6)*(1.0*cos(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) - 1.0*cos(q2)*sin(q1)*sin(q4))

r32 =

- 1.0*cos(q6)*(1.0*sin(q2)*sin(q4) + 1.0*cos(q2)*cos(q4)*sin(q3)) - 1.0*cos(q2)*cos(q3)*sin(q6)

r33 =

cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(q2)

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))

Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

1.0*cos(q1)*cos(q2)*cos(q4) - 1.0*sin(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3))

r23 =

1.0*sin(q4)*(1.0*cos(q1)*cos(q3) + 1.0*sin(q1)*sin(q2)*sin(q3)) + 1.0*cos(q2)*cos(q4)*sin(q1)

r33 =

cos(q2)*sin(q3)*sin(q4) - 1.0*cos(q4)*sin(q2)

r12 =

cos(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4)) - 1.0*sin(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2))

r11 =

cos(q6)*(sin(q1)*sin(q3) + cos(q1)*cos(q3)*sin(q2)) + sin(q6)*(1.0*cos(q4)*(cos(q3)*sin(q1) - 1.0*cos(q1)*sin(q2)*sin(q3)) + 1.0*cos(q1)*cos(q2)*sin(q4))

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))

Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn]

Q =

0	0	0	0	0	0

T =

0	0	1	15
0	-1	0	-5
1	0	0	10
0	0	0	1

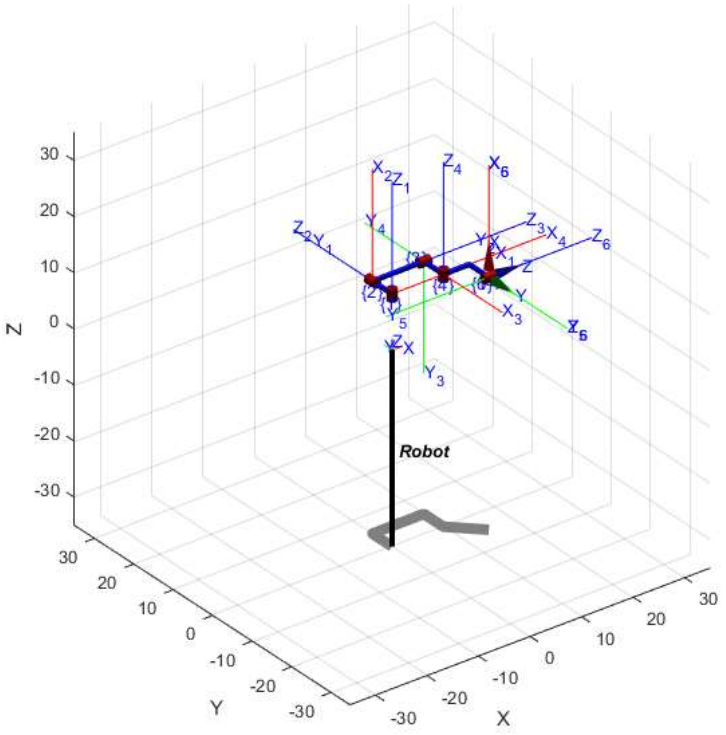
Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[15.0, -5.0, 10.0]

Matriz de rotación:

```
r =

     0     0     1
     0    -1     0
     1     0     0


Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
         0   -1.5708   -3.1416


Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
         0    1.5708   -3.1416


Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
         0    1.5708   -3.1416

PULSA CUALQUIER TECLA PARA CONTINUAR
```




```
%Punto 7

clear all
clc

syms L1 L2 L3 L6 as real
syms q1 q2 q3 q4 q5 q6 as real

P7=[0    0      L1     q1    0     0;
    0   -pi/2   L2     q2   -pi/2  0;
    0    pi/2   q3     pi/2  L3    1;
    0    0      0      q4    0     0;
    0    pi/2   0      q5    0     0;
    0   -pi/2   0      q6    0     0;
    0    0      L6     0     0    -1];


[Robot,L,A0N] = ForKin(P7);

L1=20;
L2=15;
L3=0;
L6=20;


q1=0;
q2=0;
q3=0;
q4=0;
q5=0;
```

```
q6=0;

P7=eval(P7);

[Robot,L,A0N] = ForKin(P7);

[N,~]=size(P7);


fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q= zeros(1,Robot.n)
evaluar(A0N, Q);
graficar (Robot,L, Q , 30, 20, 10)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
pause (5)

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q=[pi/6, pi/12, 20, 0, pi, pi/2]
evaluar(A0N, Q);
graficar (Robot,L, Q, 30, 20, 10)

disp('PULSA CUALQUIER TECLA PARA CONTINUAR');
pause (5)

fprintf('\n\n Evaluacion de A0N con vector de variables Q = [q1 q2 ... qn] \n')
Q=[pi/6, -pi/12, 35, 0, 0, pi/2]
evaluar(A0N, Q);
graficar (Robot,L, Q, 30, 20, 10)
```

```
function [Rob,l,T] = ForKin(DHpar)

    [N,~] = size(DHpar);
    proof=0;

    for i = 1 : N

        if DHpar(i,6) == 0
            L(i) = Link('revolute','alpha',DHpar(i,2),'a',DHpar(i,1),'d',DHpar(i,3),'offset',DHpar(i,5),'qlim',[-pi pi],'modified');
        elseif DHpar(i,6) == 1
            L(i) = Link('prismatic','alpha',DHpar(i,2),'a',DHpar(i,1),'theta',DHpar(i,4),'offset',DHpar(i,5),'qlim',[0, 40],'modified');
        else
            Robot = SerialLink(L,'name','Robot');

            Twt = trotx(DHpar(i,2))*transl(DHpar(i,1),0,0)*...
                  trotz(DHpar(i,4))*transl(0,0,DHpar(i,3));

            Twt(isAlways(abs(Twt) <= 1e-10,'Unknown','false') ) = 0;

            Robot.tool = trotx(DHpar(i,2))*transl(DHpar(i,1),0,0)*...
                         trotz(DHpar(i,4))*transl(0,0,DHpar(i,3));

            proof=1;
            break;
        end
    end

    if proof==0
        Robot = SerialLink(L,'name','Robot');
    end

    Robot

    Rob = Robot;

    l=L;

    d1 = digits(5);
    eval(sprintf('A%d%d = eye(4);', 0, N));

    for i = 1 : N

        if i==N && proof==1
            eval(sprintf('A%d%d = vpa(simplify(A%d%d*Twt));', 0, N, 0, N));
            break
        end
        eval(sprintf('syms q%d as real', i));
        eval(sprintf('A%d%d = vpa(simplify(L(i).A(q%d)));', i-1, i, i));
        eval(sprintf('A%d%d(isAlways(abs(A%d%d) <= 1e-10,''Unknown'',''false'')) = 0;'...
                     , i-1, i, i-1, i))
        if L(i).sigma
            eval(sprintf('Ainter = A%d%d;', i-1, i));

            eval(sprintf('q%d = 1;', i));

            symb = symvar(Ainter);

            for j=1 : length(symb)
                eval(strcat(char(symb(1,j)),'=1;'))
            end

            Ainter = round(eval(Ainter), 6, 'significant');

            for j=1 : 4
                for k=1 : 4
                    if abs(Ainter(j,k)) <= 1e-10
```

```
                        Ainter(j,k) = 0;
                    else
                        Ainter(j,k) = 1;
                    end
                end
            end

            eval(sprintf('A%d%d = vpa(simplify(Ainter.*A%d%d));', i-1, i, i-1, i));
            eval(sprintf('syms q%d as real', i));

            for j=1 : length(symb)
                eval(['syms ', char(symb(1,j)),' as real'])
            end
        end

        eval(sprintf('A%d%d', i-1, i))

        eval(sprintf('A%d%d = vpa(simplify(A%d%d*A%d%d));', 0, N, 0, N, i-1, i));
    end

    if proof
        eval(sprintf('A%d%d = vpa(Twt);', N-1, N));
        eval(sprintf('A%d%d', N-1, N))
    end

    eval(sprintf('T = vpa(simplify(A%d%d));', 0, N))
    eval(sprintf('A%d%d', 0, N))

    %fkine(Robot)

    fprintf('\n\nPosición en coordenadas Cartesianas: Pos = [x y z]''\n')

    Pos = vpa([T(1,4); T(2,4); T(3,4)])

    fprintf('\n\nOrientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]\n')

    r = T(1:3,1:3);

    r31 = r(3,1)
    r11 = r(1,1)
    r21 = r(2,1)
    r32 = r(3,2)
    r33 = r(3,3)

    fprintf('beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))\n\n')
    fprintf('alpha = atan2(r32/cos(B),r33/cos(B))\n\n')
    fprintf('gamma = atan2(r21/cos(B),r11/cos(B))\n\n')


    fprintf('\n\nOrientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]\n')

    r = T(1:3,1:3);

    r13 = r(1,3)
    r23 = r(2,3)
    r33 = r(3,3)
    r12 = r(1,2)
    r11 = r(1,1)

    fprintf('beta = atan2(r13,sqrt((r23^2)+(r(33^2))))\n\n')
    fprintf('alpha = atan2(-r23/cos(B),r33/cos(B))\n\n')
    fprintf('gamma = atan2(-r12/cos(B),r11/cos(B))\n\n')


end


function T = evaluar(A0N, Q)

    for i=1 : length(Q)
        eval(sprintf('q%d = Q(i);', i));
    end

    T = eval(A0N)

    fprintf('\n\nPosición en coordenadas Cartesianas: Pos = [x y z]\n')

    Pos = vpa([T(1,4), T(2,4), T(3,4)])

    fprintf('\n\nMatriz de rotación:\n')

    r = T(1:3,1:3)

    fprintf('\n\nOrientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]\n')
    fijXYZ = ones(1,3);
    fijXYZ(2) = atan2(-r(3,1),sqrt((r(1,1)^2)+(r(2,1)^2)));

    if fijXYZ(2) == pi/2
        fijXYZ(1) = 0;
        fijXYZ(3) = atan2(r(1,2),r(2,2));
    elseif fijXYZ(2) == -pi/2
        fijXYZ(1) = 0;
        fijXYZ(3) = -atan2(r(1,2),r(2,2));
    else
        fijXYZ(1) = atan2(r(3,2)/cos(fijXYZ(2)),r(3,3)/cos(fijXYZ(2)));
        fijXYZ(3) = atan2(r(2,1)/cos(fijXYZ(2)),r(1,1)/cos(fijXYZ(2)));
    end
```

```
    disp(fijXYZ)


    fprintf('\n\nOrientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]\n')

    eulXYZ = ones(1,3);
    eulXYZ(2) = atan2(r(1,3),sqrt((r(2,3)^2)+(r(3,3)^2)));

    if eulXYZ(2) == pi/2
        eulXYZ(1) = 0;
        eulXYZ(3) = atan2(r(2,1),r(2,2));
    elseif eulXYZ(2) == -pi/2
        eulXYZ(1) = 0;
        eulXYZ(3) = atan2(r(2,1),r(2,2));
    else
        eulXYZ(1) = atan2(-r(2,3)/cos(eulXYZ(2)),r(3,3)/cos(eulXYZ(2)));
        eulXYZ(3) = atan2(-r(1,2)/cos(eulXYZ(2)),r(1,1)/cos(eulXYZ(2)));
    end

    disp(eulXYZ)



    fprintf('\n\nOrientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]\n')

    eulZYZ = ones(1,3);
    eulZYZ(2) = atan2(sqrt((r(3,1)^2)+(r(3,2)^2)),r(3,3));

    if eulZYZ(2) == 0
        eulZYZ(1) = 0;
        eulZYZ(3) = atan2(-r(1,2),r(1,1));
    elseif abs(eulZYZ(2)) == pi || abs(eulZYZ(2)) == -pi
        eulZYZ(1) = 0;
        eulZYZ(3) = atan2(r(1,2),-r(1,1));
    else
        eulZYZ(1) = atan2(r(2,3)/sin(eulZYZ(2)),r(1,3)/sin(eulZYZ(2)));
        eulZYZ(3) = atan2(r(3,2)/sin(eulZYZ(2)),-r(3,1)/sin(eulZYZ(2)));
    end

    disp(eulZYZ)

end


function graficar (Robot, L, Q, escala1, escala2, escala3)

    cla();


    tam = (sum(Robot.d)+sum(Robot.a))+escala1;

    trplot(eye(4),'length',2,'rgb');
    hold on
    axis([-tam tam -tam tam -tam tam])
    Robot.plot(Q,'workspace',[-tam tam -tam tam -tam tam],'scale',0.25);
    %Robot.teach()

    q_alt = Q;
    M = eye(4);
    Mp = ones(1, Robot.n);
    for i=1:Robot.n
        M = M * L(i).A(q_alt(i));
        Mp(i)=trplot(M,'rgb','frame',num2str(i),'length',escala2,'text_opts', {'FontSize', escala3});
    end


end
```

```
Robot =

Robot (2 axis, PP, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|          0|         q1|          0|          0|          0|
|  2|          0|         q2|          0|      1.571|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1


A01 =

[ 1.0,   0,   0,   0]
[   0, 1.0,   0,   0]
[   0,   0, 1.0,  q1]
[   0,   0,   0, 1.0]


A12 =
```

```
[ 1.0,   0,    0,       0]
[   0,   0, -1.0, -1.0*q2]
[   0, 1.0,    0,       0]
[   0,   0,    0,     1.0]


A02 =

[ 1.0,   0,    0,       0]
[   0,   0, -1.0, -1.0*q2]
[   0, 1.0,    0,      q1]
[   0,   0,    0,     1.0]



Posición en coordenadas Cartesianas: Pos = [x y z]'

Pos =

       0
 -1.0*q2
      q1



Orientación en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]

r31 =

0


r11 =

1.0


r21 =

0


r32 =

1.0


r33 =

0

beta = atan2(-r31,sqrt((r11^2)+(r(21^2))))

alpha = atan2(r32/cos(B),r33/cos(B))

gamma = atan2(r21/cos(B),r11/cos(B))



Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]

r13 =

0


r23 =

-1.0


r33 =

0


r12 =

0


r11 =

1.0

beta = atan2(r13,sqrt((r23^2)+(r(33^2))))

alpha = atan2(-r23/cos(B),r33/cos(B))

gamma = atan2(-r12/cos(B),r11/cos(B))



 Evaluacion de A0N con vector  de variables Q = [q1 q2 ... qn]

Q =
```

```
     0     0


T =

     1     0     0     0
     0     0    -1     0
     0     1     0     0
     0     0     0     1


Posición en coordenadas Cartesianas: Pos = [x y z]

Pos =

[ 0, 0, 0]


Matriz de rotación:

r =

     1     0     0
     0     0    -1
     0     1     0


Orientacion en angulos fijos tipo XYZ: fijXYZ = [alpha beta gamma]
    1.5708         0         0


Orientacion en angulos de Euler tipo XYZ: eulXYZ = [alpha beta gamma]
    1.5708         0         0


Orientacion en angulos de Euler tipo ZYZ: eulZYZ = [alpha beta gamma]
   -1.5708    1.5708    1.5708

PULSA CUALQUIER TECLA PARA CONTINUAR
```

Published with MATLAB® R2017a