Automatización y robótica

Universitat d'Alacant
Universidad de Alicante

Práctica 2 – Modelado y simulación haciendo uso de la librería Robotics Toolbox desarrollada por Peter I. Corke

Francisco Joaquín Murcia Gómez

25 de mayo de 2022

Índice

1. Ejercicio 1

1.1. Enunciado

Mediante las funciones de las herramientas matemáticas, obtener la matriz de transformación y graficar el resultado que representa las siguientes transformaciones sobre un sistema OXYZ fijo de referencia: traslación de (-3,10,10); giro de -90º sobre el eje O'U del sistema trasladado y giro de 90º sobre el eje O'V' del sistema girado.

1.2. Código

```
% EJERCICIO 1%
T = transl(-3,10,10)*trotx(-90)*troty(90)
```

1.3. Resultados

```
>> ej1

T =

   -0.4481         0    0.8940   -3.0000
   -0.7992   -0.4481   -0.4006   10.0000
    0.4006   -0.8940    0.2008   10.0000
         0         0         0    1.0000
```

Figura 1: Matriz de transformación

2. Ejercicio 2

2.1. Enunciado

Modelado del robot PA10 de 6GDL a partir de la siguiente tabla de sus parámetros DH estándar y los límites articulares.

Transformación	Θ	d	a	α	Límite q(º)	Offset
0 → 1 $^0A_1$	q_1	0.317	0	-pi/2	[-177,177]	0
1 → 2 $^1A_2$	q_2	0	0.45	0	[-64,124]	-pi/2
2 → 3 $^2A_3$	q_3	0	0	pi/2	[-107,158]	pi/2
3 → 4 $^3A_4$	q_4	0.48	0	-pi/2	[-255,255]	0
4 → 5 $^4A_5$	q_5	0	0	pi/2	[-165,165]	0
5 → 6 $^5A_6$	q_6	0.07	0	0	[-255,255]	0

Figura 2: Tabla de sus parámetros DH estándar y los límites articulares

2.2. Código

```
%ejercicio 2

%% Datos
L1 = Link([0 0.317 0 -pi/2  0]);
L2 = Link([0 0 0.45 0 0 -pi/2]);
L3 = Link([0 0 0 pi/2 0 pi/2]);
L4 = Link([0 0.48 0 -pi/2 0]);
L5 = Link([0 0 0 pi/2 0]);
L6 = Link([0 0.07 0 0 0]);

%% Limites articulares
L1.qlim=[deg2rad(-177) deg2rad(177)];
L2.qlim=[deg2rad(-64) deg2rad(124)];
L3.qlim=[deg2rad(-107) deg2rad(158)];
L4.qlim=[deg2rad(-255) deg2rad(255)];
L5.qlim=[deg2rad(-165) deg2rad(165)];
L6.qlim=[deg2rad(-255) deg2rad(255)];

L=[L1,L2,L3,L4,L5,L6];

%% creacion del robot
robot = SerialLink(L);
robot.links
robot
```

2.3. Resultados

```
>> ej1

T =

   -0.4481         0    0.8940   -3.0000
   -0.7992   -0.4481   -0.4006   10.0000
    0.4006   -0.8940    0.2008   10.0000
         0         0         0    1.0000
```

Figura 3: Información del robot PA10

3. Ejercicio 3

3.1. Enunciado

Definir las siguientes posiciones articulares para el PA10 (las posiciones se indican en grados, pero en Matlab hay que introducirlas en radianes), calcular la cinemática directa (matriz T) para cada uno de ellos y realizar un plot en esa posición.

- Posición de home: qh = [0, 0, 0, 0, 0, 0].
- Posición de escape: qe = [0, 30, 90, 0, 60, 0].
- Posición de seguridad: qs = [0, 45, 90, 0, -45, 0].
- Posición q1 = [0, 45, 45, 0, 90, 0].
- Posición q2 = [20, 90, 45, -22.5, 60, 0].

3.2. Código

```
%ejercicio 3

%% Datos %
L1 = Link([0 0.317 0 -pi/2  0]);
L2 = Link([0 0 0.45 0 0 -pi/2]);
L3 = Link([0 0 0 pi/2 0 pi/2]);
L4 = Link([0 0.48 0 -pi/2 0]);
L5 = Link([0 0 0 pi/2 0]);
L6 = Link([0 0.07 0 0 0]);
L1.qlim=[deg2rad(-177) deg2rad(177)];
L2.qlim=[deg2rad(-64) deg2rad(124)];
L3.qlim=[deg2rad(-107) deg2rad(158)];
L4.qlim=[deg2rad(-255) deg2rad(255)];
L5.qlim=[deg2rad(-165) deg2rad(165)];
L6.qlim=[deg2rad(-255) deg2rad(255)];

L=[L1,L2,L3,L4,L5,L6];

robot = SerialLink(L,'name', 'PA10-6GDL');
robot;

%% Posición de home
fprintf("\n POSICION HOME: \n")
figure(1);
qh = [0 0 0 0 0 0];
qhT = robot.fkine(qh);
robot.plot(qh);
qhT
fprintf('\n')
pause(2)

%% Posición de escape
fprintf("\n PSICICION ESCAPE: \n")
figure(1);
qe = [0, pi/6, pi/2, 0, pi/3, 0];
qeT = robot.fkine(qe);
robot.plot(qe);
qeT
```

```
fprintf('\n')
pause(2)

%% Posición de seguridad
fprintf("\n POSICION DE SEGURIDAD:\n ")
figure(1);
qs = [0, pi/4, pi/2, 0, -pi/4, 0];
qsT = robot.fkine(qs);
robot.plot(qs);
qsT
fprintf('\n')
pause(2)

%% Posición q1
fprintf("\n POSICION Q1: \n")
figure(1);
q1 = [0, pi/4, pi/4, 0, pi/2, 0];
q1T = robot.fkine(q1);
robot.plot(q1);
q1T
fprintf('\n')
pause(2)

%% Posición q2
fprintf("\n POSICION Q2: \n")
figure(1);
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];
q2T = robot.fkine(q2);
robot.plot(q2);
q2T
fprintf('\n')

close all
```

3.3. Resultados

```
 POSICION HOME:

qhT =
         1         0         0         0
         0         1         0         0
         0         0         1     1.317
         0         0         0         1


 PSICICION ESCAPE:

qeT =
        -1         0         0    0.6407
         0         1         0         0
         0         0        -1    0.3967
         0         0         0         1


 POSICION DE SEGURIDAD:

qsT =
         0         0         1    0.7276
         0         1         0         0
        -1         0         0    0.2958
         0         0         0         1


 POSICION Q1:

q1T =
        -1         0         0    0.7982
         0         1         0         0
         0         0        -1    0.5652
         0         0         0         1


 POSICION Q2:

q2T =
   -0.8169   -0.5703   -0.0861    0.7358
   -0.5010    0.7756   -0.3840    0.2431
    0.2857   -0.2706   -0.9193  -0.08676
         0         0         0         1
```

Figura 4: Posiciones articulares

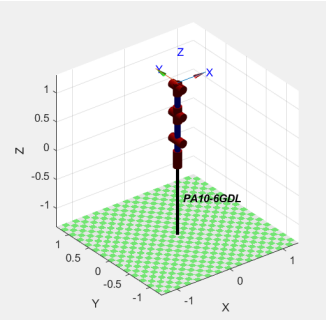

Figura 5: Posición home

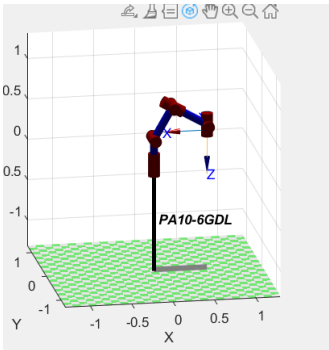

Figura 6: Posición escape

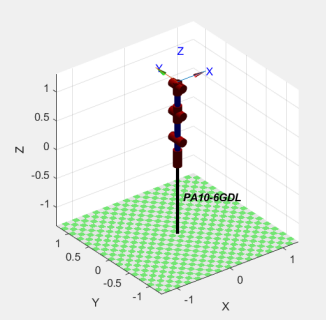

Figura 7: Posición escape

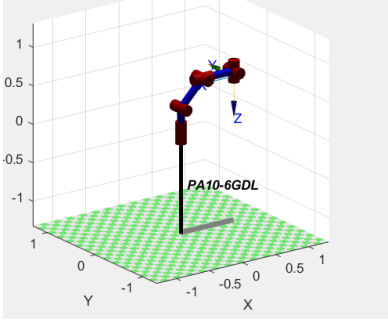

Figura 8: Posición 1

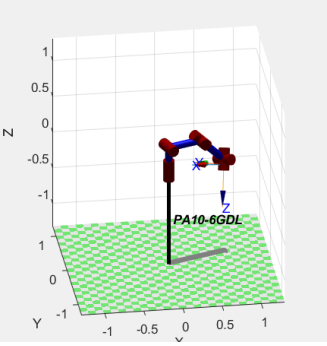

Figura 9: Posición 2

4. Ejercicio 4

4.1. Enunciado

Realizar la resolución de la cinemática inversa para el resto de posiciones del PA10 (qe, qs, q1, q2) siguiendo el mismo procedimiento que en el ejemplo mostrado utilizando las funciones ikine6s e ikunc

4.2. Código

```
%ejercicio 4

%% Datos %
L1 = Link([0 0.317 0 -pi/2  0]);
L2 = Link([0 0 0.45 0 0 -pi/2]);
L3 = Link([0 0 0 pi/2 0 pi/2]);
L4 = Link([0 0.48 0 -pi/2 0]);
L5 = Link([0 0 0 pi/2 0]);
L6 = Link([0 0.07 0 0 0]);
L1.qlim=[deg2rad(-177) deg2rad(177)];
L2.qlim=[deg2rad(-64) deg2rad(124)];
L3.qlim=[deg2rad(-107) deg2rad(158)];
L4.qlim=[deg2rad(-255) deg2rad(255)];
L5.qlim=[deg2rad(-165) deg2rad(165)];
L6.qlim=[deg2rad(-255) deg2rad(255)];

L=[L1,L2,L3,L4,L5,L6];

robot = SerialLink(L);
robot;

%% Posición de home inversa
fprintf("\n POSICION HOME INVERSA: \n")
figure(1);
qh = [0 0 0 0 0 0];
Tqh = robot.fkine(qh);
qinversaQH = robot.ikine6s(Tqh);
robot.plot(qinversaQH);
TinversaQH = robot.fkine(qinversaQH);
Tqh;
qinversaQH
TinversaQH
pause(2)

%% Posición de escape inversa
fprintf("\n PSICICION ESCAPE INVERSA: \n")
figure(1);
qe = [0, pi/6, pi/2, 0, pi/3, 0];
Tqe = robot.fkine(qe);
qinversaQE = robot.ikine6s(Tqe);
robot.plot(qinversaQE);
TinversaQE = robot.fkine(qinversaQE);
Tqe;
qinversaQE
TinversaQE
pause(2)

```

```
%% Posición de seguridad inversa
fprintf("\n POSICION DE SEGURIDAD INVERSA:\n ")
figure(1);
qs = [0, pi/4, pi/2, 0, -pi/4, 0];
Tqs = robot.fkine(qs);
qinversaQS = robot.ikine6s(Tqs);
robot.plot(qinversaQS);
TinversaQS = robot.fkine(qinversaQS);
Tqs;
qinversaQS
TinversaQS
pause(2)

%% Posición q1 inversa
fprintf("\n POSICION Q1 INVERSA: \n")
q1 = [0, pi/4, pi/4, 0, pi/2, 0];
Tq1 = robot.fkine(q1);
qinversaQ1 = robot.ikine6s(Tq1);
robot.plot(qinversaQ1);
TinversaQ1 = robot.fkine(qinversaQ1);
Tq1;
qinversaQ1;
TinversaQ1
pause(2)

%% Posición q2 inversa
fprintf("\n POSICION Q2 INVERSA: \n")
figure(1);
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];
Tq2 = robot.fkine(q2);
qinversaQ2 = robot.ikine6s(Tq2);
robot.plot(qinversaQ2);
TinversaQ2 = robot.fkine(qinversaQ2);
Tq2;
qinversaQ2
TinversaQ2
```

4.3. Resultados

```
 POSICION HOME INVERSA:

qinversaQH =

    2.3562         0   -0.0000   -3.1416   -0.0000    0.7854

TinversaQH =
         1         0         0         0
         0         1         0         0
         0         0         1     1.317
         0         0         0         1

 PSICICION ESCAPE INVERSA:

qinversaQE =

   -3.1416   -1.4470   -0.0000    0.0000   -1.6946   -3.1416

TinversaQE =
        -1         0         0    0.9229
         0         1         0         0
         0         0        -1    0.3618
         0         0         0         1

 POSICION DE SEGURIDAD INVERSA:

qinversaQS =

   -3.1416   -1.5999    0.0000   -3.1416   -0.0291    0.0000

TinversaQS =
         0         0         1    0.9996
         0         1         0         0
        -1         0         0    0.2899
         0         0         0         1

 POSICION Q1 INVERSA:


TinversaQ1 =
        -1         0         0    0.8881
         0         1         0         0
         0         0        -1    0.5231
         0         0         0         1

 POSICION Q2 INVERSA:

qinversaQ2 =

   -2.8225   -2.0511   -0.0000   -0.3557   -1.3228    2.9973

TinversaQ2 =
   -0.8169   -0.5703   -0.0861    0.7771
   -0.5010    0.7756   -0.3840    0.2319
    0.2857   -0.2706   -0.9193   -0.1771
         0         0         0         1
```

Figura 10: Posiciones articulares inversas

5. Ejercicio 5

5.1. Enunciado

Evalúa al robot PA10 y al robot planar en otras posiciones al límite de su espacio de trabajo o donde existan alineaciones de ejes (puedes emplear la función rand para probar diferentes posiciones).

5.2. Código

```
%ejercicio 5

%% robots

%robot PA10
L1 = Link([0 0.317 0 -pi/2  0]);
L2 = Link([0 0 0.45 0 0 -pi/2]);
L3 = Link([0 0 0 pi/2 0 pi/2]);
L4 = Link([0 0.48 0 -pi/2 0]);
L5 = Link([0 0 0 pi/2 0]);
L6 = Link([0 0.07 0 0 0]);
L1.qlim=[deg2rad(-177) deg2rad(177)];
L2.qlim=[deg2rad(-64) deg2rad(124)];
L3.qlim=[deg2rad(-107) deg2rad(158)];
L4.qlim=[deg2rad(-255) deg2rad(255)];
L5.qlim=[deg2rad(-165) deg2rad(165)];
L6.qlim=[deg2rad(-255) deg2rad(255)];

Lpa10=[L1,L2,L3,L4,L5,L6];

robotPA10 = SerialLink(Lpa10,'name', 'PA10-6GDL');
robotPA10;

%robot PLANAR
L10 = Link([0 0 1 0]);
L20 = Link([0 0 1 0]);
L30 = Link([0 0 1 0]);

Lplanar=[L10,L20,L30];

robotPlanar = SerialLink(Lplanar,'name', 'PLANAR');
robotPlanar;


%% calculo de singularidades robot PA10

q = [0 -pi/2 0 0 0 0];
j1=jacob0(robotPA10, q)
disp(det(j1));
figure(1)
robotPA10.plot(q);
pause(2)

q = [0, pi/6, pi/2, deg2rad(-255), pi/3, 0];
j2=jacob0(robotPA10, q)
disp(det(j2));
figure(1)
```

```
48  robotPA10.plot(q);
49  pause(2)
50
51  q = [0 -pi/4 0 0 0 pi/3];
52  j3=jacob0(robotPA10, q)
53  disp(det(j3));
54  figure(1)
55  robotPA10.plot(q);
56  pause(2)
57
58  close all
59
60  %% calculo de singularidades robot Planar
61
62  q = [0 0 0];
63  j4=jacob0(robotPlanar, q)
64  disp(det(j4));
65  figure(1)
66  robotPlanar.plot(q);
67  pause(2)
68
69  q = [1 0 0];
70  j5=jacob0(robotPlanar, q)
71  disp(det(j5));
72  figure(1)
73  robotPlanar.plot(q);
74  pause(2)
75
76  close all
```

5.3. Resultados

```
j1 =

   -0.0000    0.0000    0.0000         0    0.0000         0
   -1.0000   -0.0000   -0.0000         0   -0.0000         0
         0    1.0000    0.5500         0    0.0700         0
    0.0000         0         0   -1.0000         0   -1.0000
         0    1.0000    1.0000    0.0000    1.0000    0.0000
    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000

     0


j2 =

   -0.0586    0.1458   -0.2439    0.0293   -0.0480         0
    0.6788    0.0000    0.0000   -0.0157    0.0338         0
   -0.0000   -0.6788   -0.4538    0.0507    0.0382         0
    0.0000         0         0    0.8660    0.4830    0.5451
    0.0000    1.0000    1.0000   -0.0000   -0.2588    0.8365
    1.0000    0.0000    0.0000   -0.5000    0.8365   -0.0559

   -0.1198


j3 =

    0.0000    0.7071    0.3889         0    0.0495         0
   -0.7071    0.0000    0.0000         0    0.0000         0
   -0.0000    0.7071    0.3889         0    0.0495         0
    0.0000   -0.0000   -0.0000   -0.7071   -0.0000   -0.7071
    0.0000    1.0000    1.0000    0.0000    1.0000    0.0000
    1.0000         0         0    0.7071         0    0.7071

  -1.5979e-34
```

Figura 11: Posiciones articulares inversas

6. Ejercicio 6

6.1. Enunciado

calcula los pares articulares del resto de posiciones del robot PA10 (qs, q1 y q2) utilizando el comando robot.rne(q0, v0, a0)

6.2. Código

```
%ejercicio 6
%% creamos el robot
robot = DynamicParams(loadPA10Params());

%% coordenadas articulares del robot
qs = [0, pi/4, pi/2, 0, -pi/4, 0];%posicion de escape
q1 = [0, pi/4, pi/4, 0, pi/2, 0];%posicion q1
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];%posicion q2

%%  pares articulares suponiendo velocidades articulares 0
fprintf("ESCAPE:")
robot.rne(qs, [0 0 0 0 0 0], [0 0 0 0 0 0])
```

```
fprintf("Q1:")
robot.rne(q1, [0 0 0 0 0 0], [0 0 0 0 0 0])
fprintf("Q2:")
robot.rne(q2, [0 0 0 0 0 0], [0 0 0 0 0 0])
```

6.3. Resultados

```
>> ej6
HOME:
ans =

   -0.0000  -62.3155  -19.2553    0.0000    0.6263         0

Q1:
ans =

   -0.0000  -71.1769  -28.1168    0.0000    0.0000         0

Q2:
ans =

   -0.0000  -84.5689  -20.0145    0.1468   -0.1789         0
```

Figura 12: Pares articulares dadas las posiciones qs, q1 y q2

7. Ejercicio 7

7.1. Enunciado

Calcula los resultados dinámicos (par articular, par de gravedad, par de coriolis, par de inercia) para distintas posiciones con el valor de la gravedad en la Luna (g=1,62 m/s2)

7.2. Código

```
%ejercicio 7
%% creamos el robot
robot = DynamicParams(loadPA10Params());

%% coordenadas articulares del robot
qs = [0, pi/4, pi/2, 0, -pi/4, 0];
q1 = [0, pi/4, pi/4, 0, pi/2, 0];
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];

%% seteamos gravedad Lunar
robot.gravity = [0 0 1.62];

%% posicion qs
fprintf("\n===posicion qs====\n:")
fprintf("Articular:")
disp(robot.rne(qs, [0 0 0 0 0 0], [0 0 0 0 0 0]))
fprintf("Gravedad:")
disp(robot.gravload(qs));
fprintf("Inercia:")
disp(robot.itorque(qs,[1 0 0 0 0 0]));
```

```
fprintf("Coriolis:\n")
disp(robot.coriolis(qs, [1 0 0 0 0 0]));

%% posicion q1
fprintf("\n===posicion q1====\n:")
fprintf("Articular:")
disp(robot.rne(q1, [0 0 0 0 0 0], [0 0 0 0 0 0]))
fprintf("Gravedad:")
disp(robot.gravload(q1));
fprintf("Inercia:")
disp(robot.itorque(q1,[1 0 0 0 0 0]));
fprintf("Coriolis:\n")
disp(robot.coriolis(q1, [1 0 0 0 0 0]));

%% posicion q2
fprintf("\n===posicion q2====\n:")
fprintf("Articular:")
disp(robot.rne(q2, [0 0 0 0 0 0], [0 0 0 0 0 0]))
fprintf("Gravedad:")
disp(robot.gravload(q2));
fprintf("Inercia:")
disp(robot.itorque(q2,[1 0 0 0 0 0]));
fprintf("Coriolis:\n")
disp(robot.coriolis(q2, [1 0 0 0 0 0]));
```

7.3. Resultados

```
===posicion qs====
:Articular:    0.0000  -10.2906   -3.1798    0.0000    0.1034         0

Gravedad:    0.0000  -10.2906   -3.1798    0.0000    0.1034         0

Inercia:    3.3207    0.0009    0.0013    0.0129   -0.0000    0.0000

Coriolis:
    0.0000    0.4540   -1.2311    0.0000   -0.0000    0.0000
   -0.4540    0.0000    0.0000    0.0030   -0.0000    0.0003
    1.2311    0.0000    0.0000    0.0173    0.0000    0.0003
   -0.0000   -0.0030   -0.0173   -0.0000   -0.0011    0.0000
    0.0000   -0.0000    0.0000    0.0011         0    0.0003
         0   -0.0003   -0.0003   -0.0000   -0.0003         0


===posicion q1====
:Articular:   -0.0000  -11.7540   -4.6431    0.0000    0.0000         0

Gravedad:   -0.0000  -11.7540   -4.6431    0.0000    0.0000         0

Inercia:     0     0     0     0     0     0

Coriolis:
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0


===posicion q2====
:Articular:    0.0000  -13.9655   -3.3051    0.0242   -0.0296         0

Gravedad:    0.0000  -13.9655   -3.3051    0.0242   -0.0296         0

Inercia:     0     0     0     0     0     0

Coriolis:
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
```

Figura 13: Pares articulares dadas las posiciones qs, q1 y q2

Se puede observar que los pares articulares al haber menos gravedad se requiero un mar muy inferior en los motores para mover las articulaciones.

8. Ejercicio 8

8.1. Enunciado

¿Cómo afecta añadir una carga de este tipo a la componente gravitacional e inercial? ¿Y si la separamos también 0.3 m en el eje X? ¿Añadir una carga afectará sólo a la componente gravitacional? Justifica las respuestas haciendo uso del robot PA10.

8.2. Código

```
%ejercicio 8
%% creamos el robot
robot = DynamicParams(loadPA10Params());
```

```
4
5   %% coordenadas articulares del robot
6   qs = [0, pi/4, pi/2, 0, -pi/4, 0];
7   q1 = [0, pi/4, pi/4, 0, pi/2, 0];
8   q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];
9
10
11  %% con peso de 3kg
12  robot.payload(3, [0, 0, 0]);
13  fprintf("\n===============\n===Carga de 3Kg====\n===============\n:")
14
15  %%posicion qs
16  fprintf("\n===posicion qs====\n:")
17  fprintf("Articular:")
18  disp(robot.rne(qs, [0 0 0 0 0 0], [0 0 0 0 0 0]))
19  fprintf("Gravedad:")
20  disp(robot.gravload(qs));
21  fprintf("Inercia:")
22  disp(robot.itorque(qs,[1 0 0 0 0 0]));
23  fprintf("Coriolis:\n")
24  disp(robot.coriolis(qs, [1 0 0 0 0 0]));
25
26  %%posicion q1
27  fprintf("\n===posicion q1====\n:")
28  fprintf("Articular:")
29  disp(robot.rne(q1, [0 0 0 0 0 0], [0 0 0 0 0 0]))
30  fprintf("Gravedad:")
31  disp(robot.gravload(q1));
32  fprintf("Inercia:")
33  disp(robot.itorque(q1,[1 0 0 0 0 0]));
34  fprintf("Coriolis:\n")
35  disp(robot.coriolis(q1, [1 0 0 0 0 0]));
36
37  %%posicion q2
38  fprintf("\n===posicion q2====\n:")
39  fprintf("Articular:")
40  disp(robot.rne(q2, [0 0 0 0 0 0], [0 0 0 0 0 0]))
41  fprintf("Gravedad:")
42  disp(robot.gravload(q2));
43  fprintf("Inercia:")
44  disp(robot.itorque(q2,[1 0 0 0 0 0]));
45  fprintf("Coriolis:\n")
46  disp(robot.coriolis(q2, [1 0 0 0 0 0]));
47  %% con peso de 3kg separado en el eje X 0.3m
48  robot.payload(3, [0.3, 0, 0.1]);
49  fprintf("\n===============\n===Carga de 3Kg separada en X====\n===============\n:")
50
51  %%posicion qs
52  fprintf("\n===posicion qs====\n:")
53  fprintf("Articular:")
54  disp(robot.rne(qs, [0 0 0 0 0 0], [0 0 0 0 0 0]))
55  fprintf("Gravedad:")
56  disp(robot.gravload(qs));
57  fprintf("Inercia:")
58  disp(robot.itorque(qs,[1 0 0 0 0 0]));
59  fprintf("Coriolis:\n")
60  disp(robot.coriolis(qs, [1 0 0 0 0 0]));
```

```
61
62  %%posicion q1
63  fprintf("\n===posicion q1====\n:")
64  fprintf("Articular:")
65  disp(robot.rne(q1, [0 0 0 0 0 0], [0 0 0 0 0 0]))
66  fprintf("Gravedad:")
67  disp(robot.gravload(q1));
68  fprintf("Inercia:")
69  disp(robot.itorque(q1,[1 0 0 0 0 0]));
70  fprintf("Coriolis:\n")
71  disp(robot.coriolis(q1, [1 0 0 0 0 0]));
72
73  %%posicion q2
74  fprintf("\n===posicion q2====\n:")
75  fprintf("Articular:")
76  disp(robot.rne(q2, [0 0 0 0 0 0], [0 0 0 0 0 0]))
77  fprintf("Gravedad:")
78  disp(robot.gravload(q2));
79  fprintf("Inercia:")
80  disp(robot.itorque(q2,[1 0 0 0 0 0]));
81  fprintf("Coriolis:\n")
82  disp(robot.coriolis(q2, [1 0 0 0 0 0]));
```

8.3. Resultados

```
===posicion qs====
:Articular:   -0.0000  -76.7287  -27.5815    0.0000   -1.2072         0

Gravedad:   -0.0000  -76.7287  -27.5815    0.0000   -1.2072         0

Inercia:    4.4240    0.0009    0.0013   -0.0841    0.0000    0.0000

Coriolis:
    0.0000    0.4228   -1.7298    0.0000   -0.0000   -0.0000
   -0.4228    0.0000   -0.0000    0.0002    0.0000    0.0003
    1.7298    0.0000         0   -0.0275    0.0000    0.0003
   -0.0000   -0.0002    0.0275   -0.0000   -0.0011    0.0000
    0.0000    0.0000    0.0000    0.0011         0    0.0003
         0   -0.0003   -0.0003   -0.0000   -0.0003         0


===posicion q1====
:Articular:   -0.0000  -86.4461  -37.2989    0.0000   -0.0000         0

Gravedad:   -0.0000  -86.4461  -37.2989    0.0000   -0.0000         0

Inercia:    5.7799   -0.0004   -0.0000    0.0982    0.0000   -0.0006

Coriolis:
    0.0000    2.3696   -0.0982   -0.0000   -0.0982   -0.0000
   -2.3696    0.0000   -0.0000   -0.0128         0         0
    0.0982   -0.0000   -0.0000    0.0264         0         0
    0.0000    0.0128   -0.0264    0.0000   -0.0251   -0.0000
    0.0982   -0.0000   -0.0000    0.0251         0         0
         0   -0.0000   -0.0000    0.0000   -0.0000         0


===posicion q2====
:Articular:   -0.0000  -99.2808  -26.1182   -0.2829    0.3449         0

Gravedad:   -0.0000  -99.2808  -26.1182   -0.2829    0.3449         0

Inercia:    6.1951   -0.0184   -0.0175    0.0616   -0.0258   -0.0005

Coriolis:
    0.0000   -2.0896   -2.1176   -0.0263   -0.0850   -0.0000
    2.0896   -0.0000    0.0000    0.0537   -0.0126   -0.0001
    2.1176    0.0000    0.0000    0.0537   -0.0126   -0.0001
    0.0263   -0.0537   -0.0537   -0.0000   -0.0198    0.0001
    0.0850    0.0126    0.0126    0.0198         0   -0.0001
         0    0.0001    0.0001   -0.0001    0.0001         0
```

Figura 14: 3kg de peso sin desplazar

```
===posicion qs====
:Articular:   -0.0000  -79.6717  -30.5245    0.0000   -4.1502    0.0000

Gravedad:   -0.0000  -79.6717  -30.5245    0.0000   -4.1502    0.0000

Inercia:    4.8906    0.0009    0.0013    0.2522    0.0000    0.7448

Coriolis:
    0.0000   -0.3284   -2.5765   -0.0000   -0.7449    0.0000
    0.3284   -0.0000   -0.0000    0.0919   -0.0000    0.2894
    2.5765   -0.0000   -0.0000    0.1992         0    0.5758
    0.0000   -0.0919   -0.1992   -0.0000   -0.0838    0.0000
    0.7449   -0.0000         0    0.0838    0.0000    0.2703
   -0.0000   -0.2894   -0.5758   -0.0000   -0.2703         0


===posicion q1====
:Articular:    0.0000  -77.6171  -28.4699    0.0000    8.8290    0.0000

Gravedad:    0.0000  -77.6171  -28.4699    0.0000    8.8290    0.0000

Inercia:    4.6131   -0.0004   -0.0000    0.1847   -0.0000    0.4478

Coriolis:
    0.0000    1.9968   -0.1847    0.0000   -0.1847    0.0000
   -1.9968   -0.0000         0   -0.0362   -0.0000   -0.1334
    0.1847   -0.0000    0.0000    0.0984    0.0000    0.1530
   -0.0000    0.0362   -0.0984    0.0000   -0.0971    0.0000
    0.1847    0.0000    0.0000    0.0971    0.0000    0.1530
   -0.0000    0.1334   -0.1530   -0.0000   -0.1530         0


===posicion q2====
:Articular:   -0.0000  -90.3658  -17.2031   -2.1671    9.3025   -2.3891

Gravedad:   -0.0000  -90.3658  -17.2031   -2.1671    9.3025   -2.3891

Inercia:    5.0998   -0.1302   -0.1293    0.2937    0.0718    0.3639

Coriolis:
    0.0000   -1.7314   -1.7594   -0.1918   -0.0918   -0.2094
    1.7314         0    0.0000    0.3236    0.0859    0.3408
    1.7594    0.0000    0.0000    0.3236    0.0859    0.3408
    0.1918   -0.3236   -0.3236    0.0000   -0.0710    0.0001
    0.0918   -0.0859   -0.0859    0.0710    0.0000    0.0634
    0.2094   -0.3408   -0.3408   -0.0001   -0.0634         0
```

Figura 15: 3kg de peso desplazado en el eje X 0,3m

9. Ejercicio 9

9.1. Enunciado

Realiza 3 trayectorias articulares con el robot PA10 entre diferentes puntos probando el perfil trapezoidal y polinomial. Para visualizar los valores de velocidad y aceleración puedes emplear el comando plot(qd). Realiza 3 trayectorias cartesianas con el robot PA10 cambiando los valores de la posición cartesiana del robot. Para todas las trayectorias, representa gráficamente los valores de las posiciones en los tres ejes del espacio cartesiano X Y Z a lo largo de la trayectoria y los valores de su jacobiano (determinante matriz J).

9.2. Código

```
%% creamos el robot
robot = DynamicParams(loadPA10Params());

%% coordenadas articulares del robot
qs = [0, pi/4, pi/2, 0, -pi/4, 0];
q1 = [0, pi/4, pi/4, 0, pi/2, 0];
```

```
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];

qh = [0 0 0 0 0 0];
qe = [0, pi/6, pi/2, 0, pi/3, 0];
qs = [0, pi/4, pi/2, 0, -pi/4, 0];
q1 = [0, pi/4, pi/4, 0, pi/2, 0];
q2 = [deg2rad(20), pi/2, pi/4, deg2rad(-22.5), pi/3, 0];

qhQ = robot.fkine(qh);
qeQ = robot.fkine(qe);
qsQ = robot.fkine(qs);
q1Q = robot.fkine(q1);
q2Q = robot.fkine(q2);

%% trayectorias articulares
[q , qd, qdd] = jtraj(qh, q1, 50);
robot.plot(q);
plot(qd);
pause(2)

[q , qd, qdd] = jtraj(qh, q2, 50);
robot.plot(q);
plot(qd);
pause(2)

[q , qd, qdd] = jtraj(qh, qs, 50);
robot.plot(q);
plot(qd);
pause(2)

close all

%% trayectorias cartesianas

Ts = ctraj(qh, q1, 20)
plot(ts)
```

9.3. Resultados

9.3.1. Posición home a posición q1

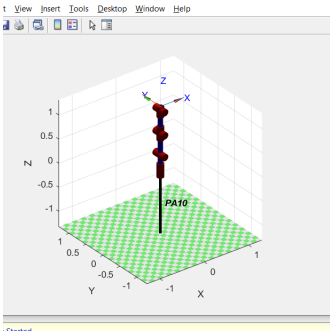

Figura 16: Posición home

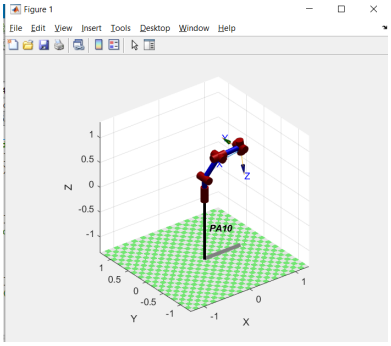

Figura 17: Posición home a q1

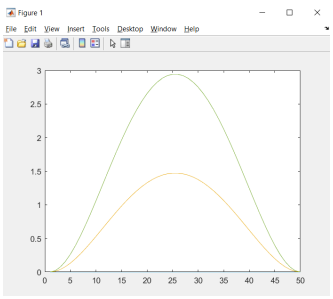

Figura 18: Valores de velocidad y aceleración home - q1

9.3.2. Posición home a posición q2

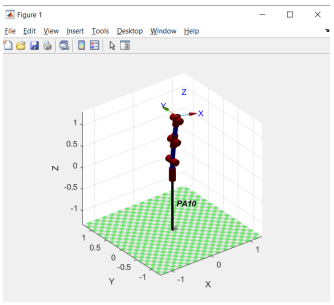

Figura 19: Posición home

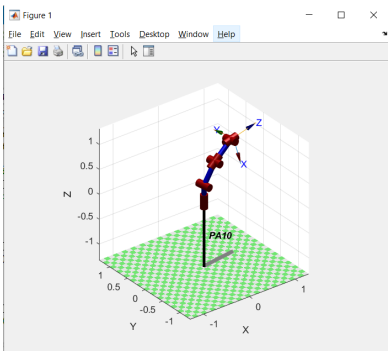

Figura 20: Posición home a q2

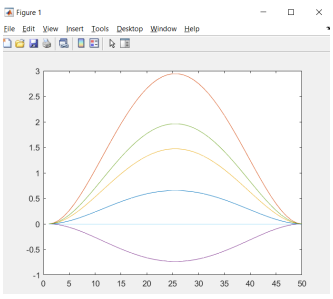

Figura 21: Valores de velocidad y aceleración home - q2

9.3.3. Posición home a posición de seguridad

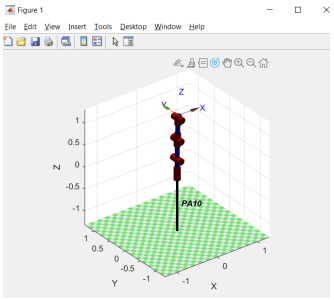

Figura 22: Posición home

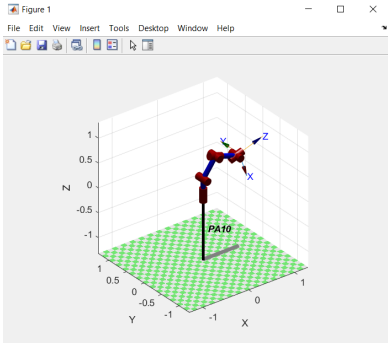

Figura 23: Posición home a posición de seguridad

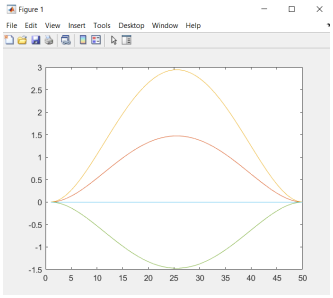

Figura 24: Valores de velocidad y aceleración home - seguridad

10. Ejercicio 10

10.1. Enunciado

Inserta el comando sl*ı anechangeenlalíneadecomandosdeMatlabparaabrirelarchivoSimulink.Ejecutadichoarchivoyvisua*

10.2. Resultados

La gráfica XY representa el plano por donde se mueve la bicicleta. si colocamos un numero positivo la bici girara a la izquierda y si ponemos uno negativo girara a la derecha. Si colocamos [0 0 0 0.5 -0.5 0.5 0 0 0 0] tendriamos lo siguiente:

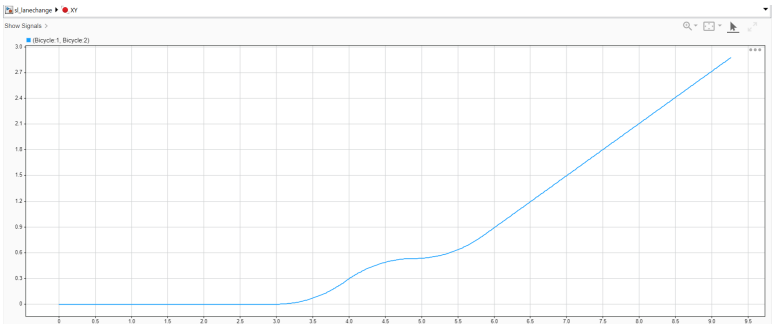

Figura 25: Giro de bicicleta

11. Ejercicio 11

11.1. Enunciado

Sobre el archivo Simulink introduce otras entradas en la dirección del vehículo y visualiza los cambios en la trayectoria. ¿Qué tipo de entrada y qué valor se debe introducir al vehículo para que la trayectoria XY sea una circunferencia en un tiempo de 10 seg?

11.2. Resultados

Para realizar un circulo se ha de colocar un valor constante todo el rato. Si colocamos 0.7 10 veces la bici realiza un giro que dura 10 segundos.

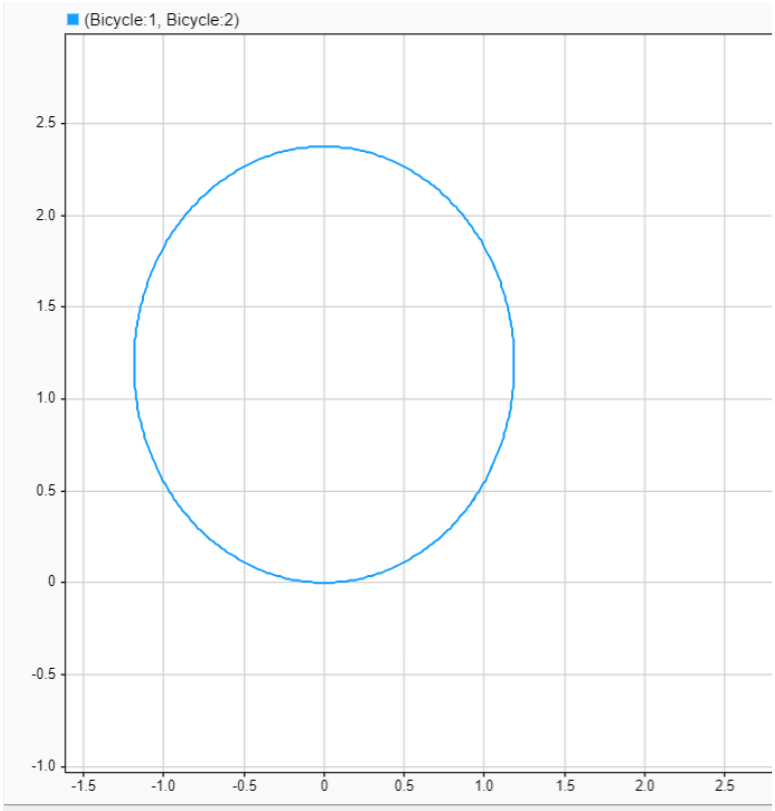

Figura 26: Circunferencia de la bicicleta