UA

AutomatizaciónyRobótica

DanielAsensiRochDNI : **48776120C**

6 de abril de 2023

Índice

1. Ejercicio 1

1.1. Enunciado

Ejercicio práctico 1: 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. Resolución

```
% Se define una matriz de transformaci n
T = transl(-3, 10, 10) * trotx(-90) * troty(90);

% Redondeamos la matriz
T_rounded = round(T);

% Mostramos la matriz redondeada en un gr fico
trplot(T_rounded);
```

1.3. Resultado Obtenido

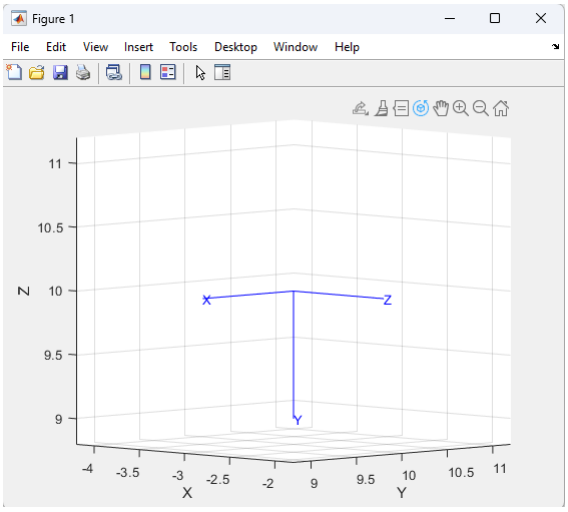


Figura 1: Ejercicio 1

2. Ejercicio 2

2.1. Enunciado

Ejercicio práctico 2: modelado del robot PA10 de 6GDL a partir de la siguiente tabla de sus parámetros DH estándar y los límites articulares. Para introducir los límites articulares y el offset de la articulación, mira en la web siguiente o teclea el comando "help SerialLink"

2.2. Resolución

```
% Definici n de los objetos Link
L(1) = Link('d', 0.317, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-177, 177]));
L(2) = Link('d', 0, 'a', 0.45, 'alpha', 0, 'offset', -pi/2, 'qlim', deg2rad([-64, 124]));
L(3) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/2, 'qlim', deg2rad([-107, 158]));
L(4) = Link('d', 0.48, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-255, 255]));
```

```
L(5) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', 0, 'qlim', deg2rad([-165, 165]));
L(6) = Link('d', 0.07, 'a', 0, 'alpha', 0, 'offset', 0, 'qlim', deg2rad([-255, 255]));

% Definici n del objeto SerialLink
PA10 = SerialLink([L(1) L(2) L(3) L(4) L(5) L(6)], 'name', 'PA10-6GDL');
PA10
```

2.3. Resultado obtenido

```
    PA10 =

PA10-6GDL (6 axis, RRRRRR, stdDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|     0.317|          0|     -1.571|          0|
|  2|         q2|          0|       0.45|          0|     -1.571|
|  3|         q3|          0|          0|      1.571|      1.571|
|  4|         q4|      0.48|          0|     -1.571|          0|
|  5|         q5|          0|          0|      1.571|          0|
|  6|         q6|      0.07|          0|          0|          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
```

3. Ejercicio 3

3.1. Enunciado

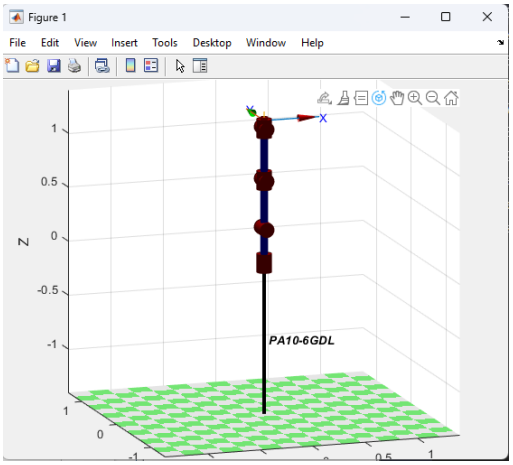
Ejercicio práctico 3: definir las siguientes posiciones articulares para el PA10 (las posiciones se indican en grados, pero en Matlab hay que introducirlas en radianes), calcular la cinemática directa (matriz T) para cada uno de ellos y realizar un plot en esa posición. Posición de home: qh = [0, 0, 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. Resolución

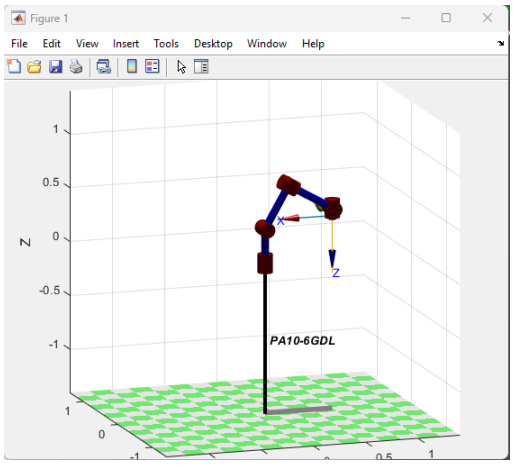
```
% EJERCICIO PRACTICO 3%

% Datos
%Primero, se definen los par metros del robot y sus l mites articulares
L(1) = Link('d', 0.317, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-177, 177]));
L(2) = Link('d', 0, 'a', 0.45, 'alpha', 0, 'offset', -pi/2, 'qlim', deg2rad([-64, 124]));
L(3) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/2, 'qlim', deg2rad([-107, 158]));
L(4) = Link('d', 0.48, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-255, 255]));
L(5) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', 0, 'qlim', deg2rad([-165, 165]));
L(6) = Link('d', 0.07, 'a', 0, 'alpha', 0, 'offset', 0, 'qlim', deg2rad([-255, 255]));


pa10 = SerialLink(L,'name', 'PA10-6GDL');

% Posicion de home
figure(1);
qh = deg2rad([0 0 0 0 0 0]);
qhQ = pa10.fkine(qh);
pa10.plot(qh);

```

```
% Posicion de escape
figure(2);
qe =deg2rad([0 30 90 0 60 0]);
qeQ = pa10.fkine(qe);
pa10.plot(qe);

% Posicion de escape
figure(6);
qe =deg2rad([0 30 90 0 60 0]);
qeQ = pa10.fkine(qe);
pa10.plot(qe);

% Posicion de seguridad
figure(3);
qs = deg2rad([0 45 90 0 -45 0]);
qsQ = pa10.fkine(qs);
pa10.plot(qs);

% Posicion q1
figure(4);
q1 = deg2rad([0 45 45 0 90 0]);
q1Q = pa10.fkine(q1);
pa10.plot(q1);

% Posicion q2
figure(5);
q2 = deg2rad([20 90 45 -22.5 60 0]);
q2Q = pa10.fkine(q2);
pa10.plot(q2);
```

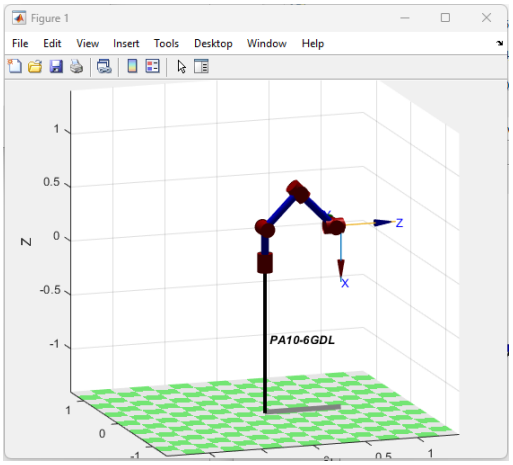
3.3. Resultado Obtenido



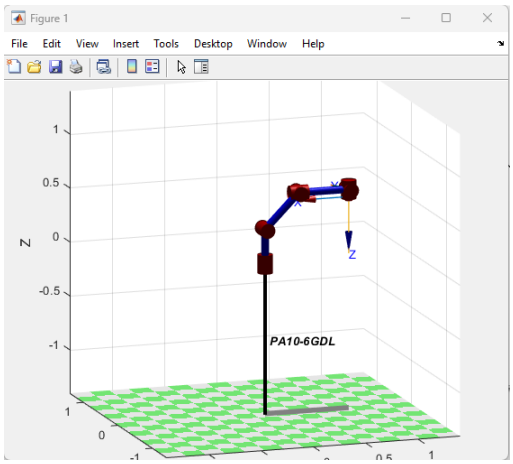
(a) Posición Home



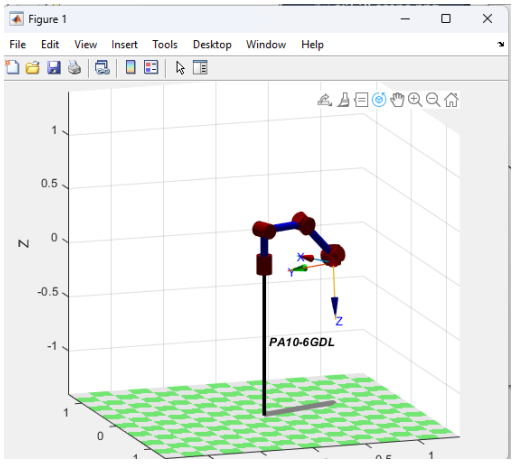
(b) Posición Escape



(c) Posición Seguridad



(d) Posición Q1



(e) Posición Q2

Figura 2: Posiciones

```
qhQ =

    1.0000         0         0    0.0000
         0    1.0000         0   -0.0000
         0         0    1.0000    1.3170
         0         0         0    1.0000


qeQ =

   -1.0000   -0.0000    0.0000    0.6407
    0.0000    1.0000    0.0000    0.0000
   -0.0000    0.0000   -1.0000    0.3967
         0         0         0    1.0000


qsQ =

         0   -0.0000    1.0000    0.7276
    0.0000    1.0000    0.0000    0.0000
   -1.0000    0.0000    0.0000    0.2958
         0         0         0    1.0000


q1Q =

   -1.0000   -0.0000    0.0000    0.7982
    0.0000    1.0000    0.0000    0.0000
   -0.0000    0.0000   -1.0000    0.5652
         0         0         0    1.0000


q2Q =

   -0.8169   -0.5703   -0.0861    0.7358
   -0.5010    0.7756   -0.3840    0.2431
    0.2857   -0.2706   -0.9193   -0.0868
         0         0         0    1.0000
```

4. Ejercicio 4

4.1. Enunciado

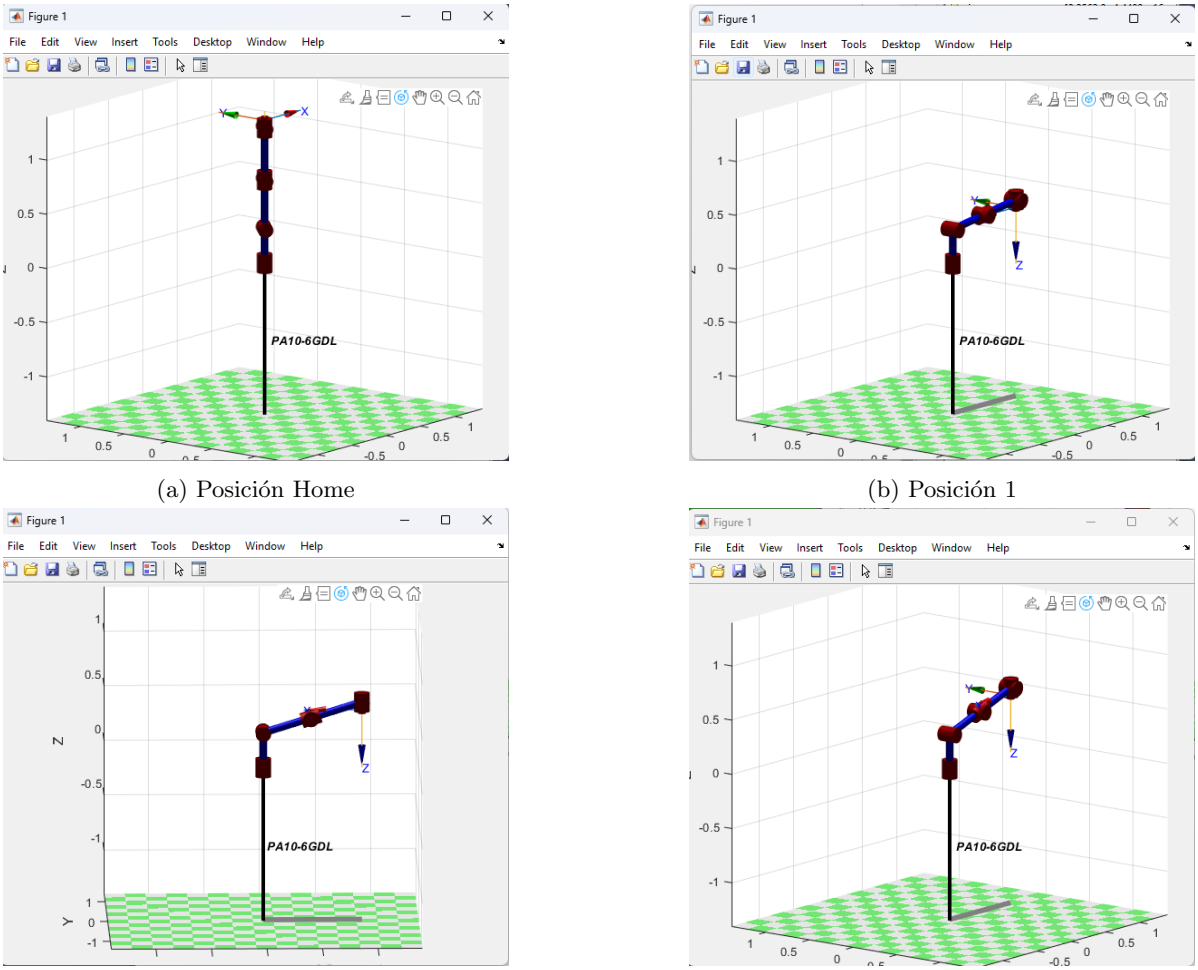
Ejercicio práctico 4: realizar la resolución de la cinemática inversa para el resto de posiciones del PA10 (qe, qs, q1, q2) siguiendo el mismo procedimiento que en el ejemplo mostrado utilizando las funciones ikine6s e ikunc. Para más información de los métodos, se puede acceder mediante el comando "help ikine6s" y "help ikunc" en Matlab.

4.2. Resolución

```
%Ejercicio 4%

% Definici n del robot PA10-6GDL
L(1) = Link('d', 0.317, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-177, 177]));
L(2) = Link('d', 0, 'a', 0.45, 'alpha', 0, 'offset', -pi/2, 'qlim', deg2rad([-64, 124]));
L(3) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/2, 'qlim', deg2rad([-107, 158]));
L(4) = Link('d', 0.48, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-255, 255]));
L(5) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', 0, 'qlim', deg2rad([-165, 165]));
L(6) = Link('d', 0.07, 'a', 0, 'alpha', 0, 'offset', 0, 'qlim', deg2rad([-255, 255]));

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

qh = [0 0 0 0 0 0];
```

```
qhQ = r.fkine(qh);

qe = [0 0.5236 1.5708 0 1.0472 0];
qeQ = r.fkine(qe);

qs = [0 0.7854 1.5708 0 -0.7854 0];
qsQ = r.fkine(qs);

q1 = [0 0.7854 0.7854 0 1.5708 0];
q1Q = r.fkine(q1);

q2 = [0.3491 1.5708 0.7854 -0.39270 1.0472 0];
q2Q = r.fkine(q2);


T = r.fkine(qh);
qinversa = r.ikine6s(T);
r.plot(qinversa);
Tinversa = r.fkine(qinversa);
T
qinversa
Tinversa

T1 = r.fkine(qe);
qinversa1 = r.ikine6s(T1);
r.plot(qinversa1);
Tinversa1 = r.fkine(qinversa1);
T1
qinversa1
Tinversa1

T2 = r.fkine(qs);
qinversa2 = r.ikine6s(T2);
r.plot(qinversa2);
Tinversa2 = r.fkine(qinversa2);
T2
qinversa2
Tinversa2

T3 = r.fkine(q1);
qinversa3 = r.ikine6s(T3);
r.plot(qinversa3);
Tinversa3 = r.fkine(qinversa3);
T3
qinversa3
Tinversa3

T4 = r.fkine(q2);
qinversa4 = r.ikine6s(T4);
r.plot(qinversa4);
Tinversa4 = r.fkine(qinversa4);
T4
qinversa4
Tinversa4
```

4.3. Resultado Obtenido



(a) Posición Home

(b) Posición 1

(c) Posición 2

(d) Posición 3

Figura 3: Posiciones

5. Ejercicio 5

5.1. Enunciado

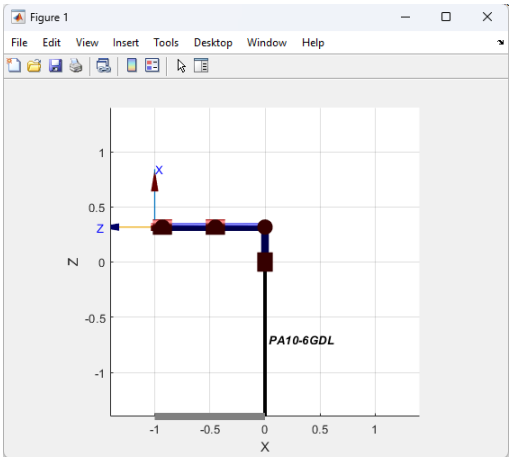
Ejercicio práctico 5: evalúa al robot PA10 y al robot planar en otras posiciones al límite de su espacio de trabajo o donde existan alineaciones de ejes (puedes emplear la función rand para probar diferentes posiciones). Para el robot planar, sólo ten en cuenta las dos primeras filas y la última de la matriz Jacobiana, ya que el resultado no es una matriz cuadrada, y sólo es necesario evaluar el espacio cartesiano plano y uno de los vectores de orientación del robot en el plano (el robot planar sólo puede posicionarse y orientarse en el plano)

5.2. Resolución

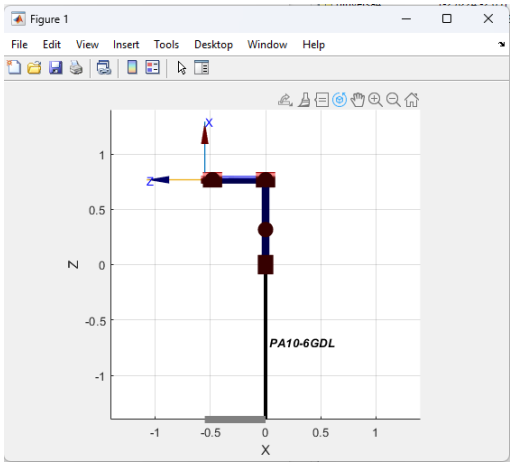
```
% Definici n de los objetos Link
L(1) = Link('d', 0.317, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-177, 177]));
L(2) = Link('d', 0, 'a', 0.45, 'alpha', 0, 'offset', -pi/2, 'qlim', deg2rad([-64, 124]));
L(3) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/2, 'qlim', deg2rad([-107, 158]));
L(4) = Link('d', 0.48, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-255, 255]));
```

```
L(5) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', 0, 'qlim', deg2rad([-165, 165]));
L(6) = Link('d', 0.07, 'a', 0, 'alpha', 0, 'offset', 0, 'qlim', deg2rad([-255, 255]));

% Definici n del objeto SerialLink
PA10 = SerialLink([L(1) L(2) L(3) L(4) L(5) L(6)], 'name', 'PA10-6GDL');

% Pruebas del robot en diferentes posiciones
q1 = [0 -pi/2 0 0 0 0];
q2 = [0 0 -pi/2 0 0 0];
q3 = [0 0 0 pi/2 0 0];
q4 = [0 0 0 0 0 pi/2];

% Imprimir matriz jacobiana, su determinante y graficar el robot en la posici n
    correspondiente
disp("Prueba 1:");
jacobian1 = jacob0(PA10, q1);
disp(jacobian1);
disp(det(jacobian1));
PA10.plot(q1);

disp("Prueba 2:");
jacobian2 = jacob0(PA10, q2);
disp(jacobian2);
disp(det(jacobian2));
PA10.plot(q2);

disp("Prueba 3:");
jacobian3 = jacob0(PA10, q3);
disp(jacobian3);
disp(det(jacobian3));
PA10.plot(q3);

disp("Prueba 4:");
jacobian4 = jacob0(PA10, q4);
disp(jacobian4);
disp(det(jacobian4));
PA10.plot(q4);
```

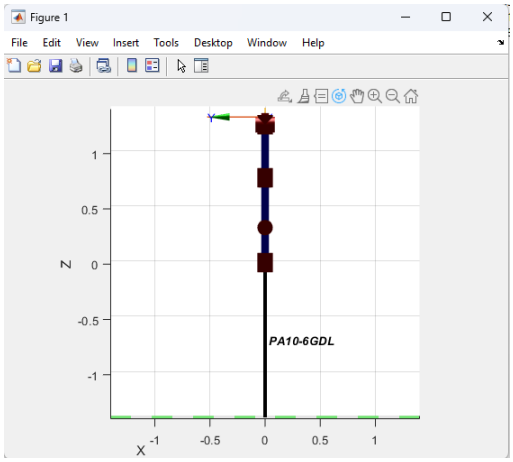
5.3. Resultado Obtenido



(a) Posición Home



(b) Posición 1



(c) Posición 2

Figura 4: Posiciones

```
    Prueba 1:
   -0.0000    0.0000    0.0000         0    0.0000         0
   -1.0000   -0.0000   -0.0000         0   -0.0000         0
   -0.0000    1.0000    0.5500         0    0.0700         0
         0         0         0   -1.0000         0   -1.0000
    0.0000    1.0000    1.0000    0.0000    1.0000    0.0000
    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000

     0

Prueba 2:
   -0.0000    0.4500    0.0000         0    0.0000         0
   -0.5500   -0.0000   -0.0000         0   -0.0000         0
    0.0000    0.5500    0.5500         0    0.0700         0
         0         0         0   -1.0000         0   -1.0000
    0.0000    1.0000    1.0000    0.0000    1.0000    0.0000
    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000

     0
```

```
Prueba 3:
    0.0000    1.0000    0.5500         0    0.0000         0
    0.0000         0    0.0000         0    0.0700         0
         0   -0.0000         0         0         0         0
   -0.0000         0         0         0   -1.0000         0
    0.0000    1.0000    1.0000         0    0.0000         0
    1.0000    0.0000    0.0000    1.0000    0.0000    1.0000

     0

Prueba 4:
    0.0000    1.0000    0.5500         0    0.0700         0
    0.0000         0    0.0000         0    0.0000         0
         0   -0.0000         0         0         0         0
   -0.0000         0         0         0         0         0
    0.0000    1.0000    1.0000         0    1.0000         0
    1.0000    0.0000    0.0000    1.0000    0.0000    1.0000

     0
```

6. Ejercicio 6

6.1. Enunciado

Ejercicio práctico 6: calcula los pares articulares del resto de posiciones del robot PA10 (qs, q1 y q2) utilizando el comando robot.rne(q0, v0, a0). Nota: por defecto, la RT calcula la dinámica inversa de un robot empleando un método rápido llamado fastRNE. Si el comando robot.rne no funcionara correctamente (da un error que cierra Matlab), será necesario cambiarlo mediante la opción robot.fast al método slowRNE. Para ello se pone el valor de robot.fast a 0 (robot.fast=0).

6.2. Resolución

```
% Definici n del robot PA10-6GDL y carga de par metros din micos
PA10 = DynamicParams(loadPA10Params());

% Definici n de posiciones articulares
qs = [0, deg2rad(45), deg2rad(90), 0, deg2rad(-45), 0]
q1 = [0, deg2rad(45), deg2rad(45), 0, deg2rad(90), 0]
q2 = [deg2rad(20), deg2rad(90), deg2rad(45), deg2rad(-22.5), deg2rad(60), 0]

% C lculo de torques utilizando la funci n rne() de Robot Toolbox
% Primera posici n
tau = PA10.rne(qs, [0 0 0 0 0 0], [0 0 0 0 0 0])

% Segunda posici n
tau = PA10.rne(q1, [0 0 0 0 0 0], [0 0 0 0 0 0])

% Tercera posici n
tau = PA10.rne(q2, [0 0 0 0 0 0], [0 0 0 0 0 0])
```

6.3. Resultado Obtenido

```
qs =

         0    0.7854    1.5708         0   -0.7854         0


q1 =

         0    0.7854    0.7854         0    1.5708         0

```

```
q2 =

    0.3491    1.5708    0.7854   -0.3927    1.0472         0


tau =

   -0.0000  -62.3155  -19.2553    0.0000    0.6263         0


tau =

   -0.0000  -71.1769  -28.1168    0.0000    0.0000         0


tau =

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

7. Ejercicio 7

7.1. Enunciado

Ejercicio práctico 7: 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). Justifica los resultados.

7.2. Resolución

```
% EJERCICIO PRACTICO 7%

% Creamos una instancia de la clase DynamicParams y cargamos los par metros del robot PA10
r = DynamicParams(loadPA10Params())

% Definimos tres posiciones articulares (en radianes) del robot
qs = [0, deg2rad(45), deg2rad(90), 0, deg2rad(-45),0]
q1 = [0, deg2rad(45), deg2rad(45), 0, deg2rad(90),0]
q2 = [deg2rad(20), deg2rad(90), deg2rad(45),deg2rad(-22.5), deg2rad(60), 0]

% Definimos la gravedad en el eje z (en m/s^2)
r.gravity = [0 0 1.62]

% Calculamos la carga gravitatoria en las tres posiciones articulares
G0 = r.gravload(qs)
G1 = r.gravload(q1)
G2 = r.gravload(q2)

% Calculamos los torques inerciales en la posici n inicial (qs) para 3 configuraciones
    distintas de la matriz de inercia
M0_0 = r.itorque(qs,[1 0 0 0 0 0])
M0_1 = r.itorque(qs,[1 1 0 0 0 0])
M0_2 = r.itorque(qs,[1 1 1 0 0 0])

% Calculamos los torques inerciales en la posici n q1 para 3 configuraciones distintas de
    la matriz de inercia
M1_0 = r.itorque(q1,[1 0 0 0 0 0])
M1_1 = r.itorque(q1,[1 1 0 0 0 0])
M1_2 = r.itorque(q1,[1 1 1 0 0 0])

% Calculamos los torques de Coriolis en la posici n inicial (qs) para 2 configuraciones
    distintas de las velocidades articulares
C0_1 = r.coriolis(qs, [0 pi 0 0 0 0])
C0_2 = r.coriolis(qs, [pi pi 0 0 0 0])
```

```
32
33 % Calculamos los torques de Coriolis en la posici n q1 para 2 configuraciones distintas de
       las velocidades articulares
34 C1_1 = r.coriolis(q1, [0 pi 0 0 0 0])
35 C1_2 = r.coriolis(q1, [pi pi 0 0 0 0])
```

7.3. Resultado Obtenido

```
1  r =
2
3  PA10 (6 axis, RRRRRR, stdDH, fastRNE)
4
5  +---+-----------+-----------+-----------+-----------+-----------+
6  | j |     theta |         d |         a |     alpha |    offset |
7  +---+-----------+-----------+-----------+-----------+-----------+
8  |  1|         q1|      0.317|          0|     -1.571|          0|
9  |  2|         q2|          0|       0.45|          0|     -1.571|
10 |  3|         q3|          0|          0|      1.571|      1.571|
11 |  4|         q4|       0.48|          0|     -1.571|          0|
12 |  5|         q5|          0|          0|      1.571|          0|
13 |  6|         q6|       0.07|          0|          0|          0|
14 +---+-----------+-----------+-----------+-----------+-----------+
15
16 grav =    0   base = 1   0   0   0    tool =  1   0   0   0
17           0           0   1   0   0            0   1   0   0
18         9.81          0   0   1   0            0   0   1   0
19                       0   0   0   1            0   0   0   1
20
21
22 qs =
23
24           0     0.7854     1.5708          0    -0.7854          0
25
26
27 q1 =
28
29           0     0.7854     0.7854          0     1.5708          0
30
31
32 q2 =
33
34      0.3491     1.5708     0.7854    -0.3927     1.0472          0
35
36
37 r =
38
39 PA10 (6 axis, RRRRRR, stdDH, fastRNE)
40
41 +---+-----------+-----------+-----------+-----------+-----------+
42 | j |     theta |         d |         a |     alpha |    offset |
43 +---+-----------+-----------+-----------+-----------+-----------+
44 |  1|         q1|      0.317|          0|     -1.571|          0|
45 |  2|         q2|          0|       0.45|          0|     -1.571|
46 |  3|         q3|          0|          0|      1.571|      1.571|
47 |  4|         q4|       0.48|          0|     -1.571|          0|
48 |  5|         q5|          0|          0|      1.571|          0|
49 |  6|         q6|       0.07|          0|          0|          0|
50 +---+-----------+-----------+-----------+-----------+-----------+
51
52 grav =    0   base = 1   0   0   0    tool =  1   0   0   0
53           0           0   1   0   0            0   1   0   0
54         1.62          0   0   1   0            0   0   1   0
55                       0   0   0   1            0   0   0   1
56
57
58 G0 =
```

```
    0.0000   -10.2906   -3.1798    0.0000    0.1034         0


G1 =

   -0.0000   -11.7540   -4.6431    0.0000    0.0000         0


G2 =

    0.0000   -13.9655   -3.3051    0.0242   -0.0296         0


M0_0 =

    3.3207    0.0009    0.0013    0.0129   -0.0000    0.0000


M0_1 =

    3.3216    3.9875    1.1747    0.0129   -0.0295    0.0000


M0_2 =

    3.3229    5.1610    2.3685    0.0129   -0.0387    0.0000


M1_0 =

    4.5375   -0.0004   -0.0000   -0.0510    0.0000   -0.0006


M1_1 =

    4.5371    5.9348    2.1694   -0.0510    0.0328   -0.0006


M1_2 =

    4.5371    8.1042    3.4065   -0.0510    0.0453   -0.0006


C0_1 =

    1.4263    0.0014    0.0040   -0.0258   -0.0000   -0.0009
    0.0000   -0.0000   -3.9881    0.0000   -0.0043    0.0000
    0.0000    3.9881    0.0000    0.0000   -0.0681         0
   -0.0094   -0.0000    0.0000   -0.0000   -0.0000    0.0006
   -0.0000    0.0043    0.0681   -0.0000         0         0
   -0.0009         0         0   -0.0006         0         0


C0_2 =

    1.4263    1.4277   -3.8636   -0.0258   -0.0001   -0.0009
   -1.4263    0.0000   -3.9881    0.0094   -0.0043    0.0009
    3.8677    3.9881    0.0000    0.0545   -0.0681    0.0009
   -0.0094   -0.0094   -0.0545   -0.0000   -0.0035    0.0006
    0.0001    0.0043    0.0681    0.0035         0    0.0009
   -0.0009   -0.0009   -0.0009   -0.0006   -0.0009         0


C1_1 =

    6.3571    0.0031    0.0057   -0.0365    0.0000   -0.0000
    0.0000    0.0000   -2.8013   -0.0000    0.1601   -0.0000
```

```
    0.0000    2.8013         0         0    0.0963         0
   -0.1021   -0.0000   -0.0000    0.0000    0.0000   -0.0009
    0.0000   -0.1601   -0.0963    0.0000         0         0
   -0.0000         0         0    0.0009         0         0


C1_2 =

    6.3571    6.3602    0.1658   -0.0365    0.1601   -0.0000
   -6.3571    0.0000   -2.8013    0.1021    0.1601   -0.0000
   -0.1601    2.8013    0.0000    0.0383    0.0963         0
   -0.1021   -0.1021   -0.0383    0.0000   -0.0343   -0.0009
   -0.1601   -0.1601   -0.0963    0.0343         0         0
   -0.0000   -0.0000   -0.0000    0.0009   -0.0000         0
```

8. Ejercicio 8

8.1. Enunciado

Ejercicio práctico 8: ¿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. Resolución

```
% Se cargan los par metros din micos del robot PA-10
r = DynamicParams(loadPA10Params());

% Se definen las posiciones articulares en radianes para tres casos distintos
% [Base, Shoulder, Elbow, Wrist1, Wrist2, Wrist3]
qs = [0, deg2rad(45), deg2rad(90), 0, deg2rad(-45),0];
q1 = [0, deg2rad(45), deg2rad(45), 0, deg2rad(90),0];
q2 = [deg2rad(20), deg2rad(90), deg2rad(45),deg2rad(-22.5), deg2rad(60), 0];

% Se a ade una carga al robot en su extremo y se obtiene la fuerza gravitacional ejercida
    en cada caso
r.payload(2.5, [0, 0, 0.1]);
Res0 = r.gravload(qs);
Res1 = r.gravload(q1);
Res2 = r.gravload(q2);

% Se calcula el torque necesario para contrarrestar la gravedad en cada posici n, con una
    fuerza en la base de 1N y cero en el resto
Vqs_0 = r.itorque(qs,[1 0 0 0 0 0]);
Vq1_0 = r.itorque(q1,[1 0 0 0 0 0]);
Vq2_0 = r.itorque(q2,[1 0 0 0 0 0]);

% Se a ade una carga en una posici n ligeramente desplazada y se obtiene la fuerza
    gravitacional en cada caso
r.payload(2.5,[0.5, 0, 0.1]);
Res0 = r.gravload(qs)
Res1 = r.gravload(q1)
Res2 = r.gravload(q2)

% Se calcula el torque necesario para contrarrestar la gravedad en cada posici n, con una
    fuerza en la base de 1N y cero en el resto
Vqs_1 = r.itorque(qs,[1 0 0 0 0 0])
Vq1_1 = r.itorque(q1,[1 0 0 0 0 0])
Vq2_1 = r.itorque(q2,[1 0 0 0 0 0])
```

8.3. Resultado Obtenido

```
Res0 =

    0.0000   -75.6123   -28.0259    0.0000   -3.3164    0.0000


Res1 =

    0.0000   -70.2684   -22.6820    0.0000   12.2625    0.0000


Res2 =

   -0.0000   -83.4466   -12.4911   -2.4363   12.2207   -3.3182


Vqs_1 =

    4.5481    0.0009    0.0013    0.5067    0.0000    1.0345


Vq1_1 =

    4.0908   -0.0004   -0.0000    0.0573   -0.0000    0.3722


Vq2_1 =

    4.6529   -0.1244   -0.1235    0.1522    0.0840    0.2758
```

9. Ejercicio 9

9.1. Enunciado

Ejercicio práctico 9: 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. Resolución

```
% EJERCICIO PRACTICO 9%

% Definici n del robot PA10-6GDL
L(1) = Link('d', 0.317, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-177, 177]));
L(2) = Link('d', 0, 'a', 0.45, 'alpha', 0, 'offset', -pi/2, 'qlim', deg2rad([-64, 124]));
L(3) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', pi/2, 'qlim', deg2rad([-107, 158]));
L(4) = Link('d', 0.48, 'a', 0, 'alpha', -pi/2, 'offset', 0, 'qlim', deg2rad([-255, 255]));
L(5) = Link('d', 0, 'a', 0, 'alpha', pi/2, 'offset', 0, 'qlim', deg2rad([-165, 165]));
L(6) = Link('d', 0.07, 'a', 0, 'alpha', 0, 'offset', 0, 'qlim', deg2rad([-255, 255]));

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

qh = [0 0 0 0 0 0];
qhQ = r.fkine(qh);

qe = [0 0.5236 1.5708 0 1.0472 0];
qeQ = r.fkine(qe);

qs = [0 0.7854 1.5708 0 -0.7854 0];
qsQ = r.fkine(qs);

```

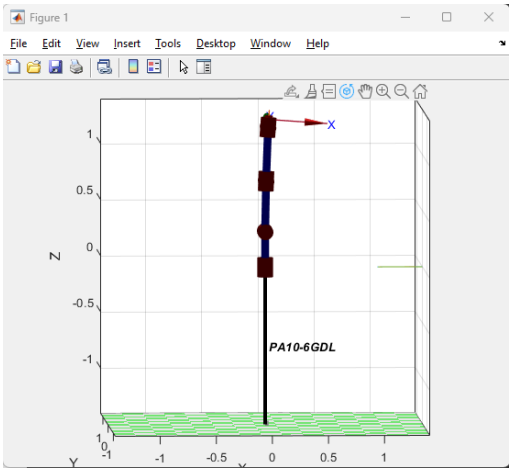
```
q1 = [0 0.7854 0.7854 0 1.5708 0];
q1Q = r.fkine(q1);

q2 = [0.3491 1.5708 0.7854 -0.39270 1.0472 0];
q2Q = r.fkine(q2);

% Pedazo plots. %
TR1 = jtraj(qh, q1, 50)
r.plot(TR1);

TR2 = jtraj(qh, q2, 50)
r.plot(TR2);

TR3 = jtraj(qh, qs, 50)
r.plot(TR3);


% [q , qd, qdd] = jtraj(qh, q2, 50); %
% [q , qd, qdd] = jtraj(qh, qs, 50); %
[q , qd, qdd] = jtraj(qh, qs, 50);

plot(qd);)
```

9.3. Resultado Obtenido

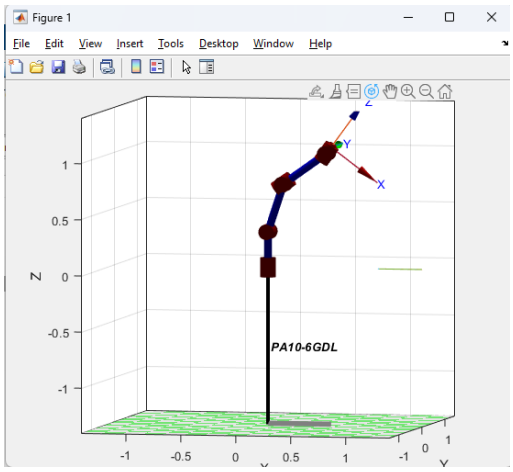
```
TR1 =

         0         0         0         0         0         0
         0    0.0001    0.0001         0    0.0001         0
         0    0.0005    0.0005         0    0.0010         0
         0    0.0016    0.0016         0    0.0033         0
         0    0.0038    0.0038         0    0.0075         0
         0    0.0071    0.0071         0    0.0142         0
         0    0.0119    0.0119         0    0.0238         0
         0    0.0183    0.0183         0    0.0365         0
         0    0.0264    0.0264         0    0.0527         0
         0    0.0362    0.0362         0    0.0725         0
         0    0.0480    0.0480         0    0.0960         0
         0    0.0616    0.0616         0    0.1232         0
         0    0.0771    0.0771         0    0.1543         0
         0    0.0945    0.0945         0    0.1890         0
         0    0.1136    0.1136         0    0.2273         0
         0    0.1345    0.1345         0    0.2690         0
         0    0.1570    0.1570         0    0.3140         0
         0    0.1810    0.1810         0    0.3620         0
         0    0.2063    0.2063         0    0.4126         0
         0    0.2329    0.2329         0    0.4657         0
         0    0.2605    0.2605         0    0.5209         0
         0    0.2889    0.2889         0    0.5779         0
         0    0.3181    0.3181         0    0.6362         0
         0    0.3477    0.3477         0    0.6955         0
         0    0.3777    0.3777         0    0.7554         0
         0    0.4077    0.4077         0    0.8154         0
         0    0.4377    0.4377         0    0.8753         0
         0    0.4673    0.4673         0    0.9346         0
         0    0.4965    0.4965         0    0.9929         0
         0    0.5249    0.5249         0    1.0499         0
         0    0.5525    0.5525         0    1.1051         0
         0    0.5791    0.5791         0    1.1582         0
         0    0.6044    0.6044         0    1.2088         0
         0    0.6284    0.6284         0    1.2568         0
         0    0.6509    0.6509         0    1.3018         0
         0    0.6718    0.6718         0    1.3435         0
         0    0.6909    0.6909         0    1.3818         0
         0    0.7083    0.7083         0    1.4165         0
         0    0.7238    0.7238         0    1.4476         0
```

```
         0    0.7374    0.7374         0    1.4748         0
         0    0.7492    0.7492         0    1.4983         0
         0    0.7590    0.7590         0    1.5181         0
         0    0.7671    0.7671         0    1.5343         0
         0    0.7735    0.7735         0    1.5470         0
         0    0.7783    0.7783         0    1.5566         0
         0    0.7816    0.7816         0    1.5633         0
         0    0.7838    0.7838         0    1.5675         0
         0    0.7849    0.7849         0    1.5698         0
         0    0.7853    0.7853         0    1.5707         0
         0    0.7854    0.7854         0    1.5708         0


TR2 =

         0         0         0         0         0         0
    0.0000    0.0001    0.0001   -0.0000    0.0001         0
    0.0002    0.0010    0.0005   -0.0003    0.0007         0
    0.0007    0.0033    0.0016   -0.0008    0.0022         0
    0.0017    0.0075    0.0038   -0.0019    0.0050         0
    0.0032    0.0142    0.0071   -0.0036    0.0095         0
    0.0053    0.0238    0.0119   -0.0060    0.0159         0
    0.0081    0.0365    0.0183   -0.0091    0.0244         0
    0.0117    0.0527    0.0264   -0.0132    0.0351         0
    0.0161    0.0725    0.0362   -0.0181    0.0483         0
    0.0213    0.0960    0.0480   -0.0240    0.0640         0
    0.0274    0.1232    0.0616   -0.0308    0.0822         0
    0.0343    0.1543    0.0771   -0.0386    0.1028         0
    0.0420    0.1890    0.0945   -0.0472    0.1260         0
    0.0505    0.2273    0.1136   -0.0568    0.1515         0
    0.0598    0.2690    0.1345   -0.0673    0.1794         0
    0.0698    0.3140    0.1570   -0.0785    0.2093         0
    0.0804    0.3620    0.1810   -0.0905    0.2413         0
    0.0917    0.4126    0.2063   -0.1032    0.2751         0
    0.1035    0.4657    0.2329   -0.1164    0.3105         0
    0.1158    0.5209    0.2605   -0.1302    0.3473         0
    0.1284    0.5779    0.2889   -0.1445    0.3852         0
    0.1414    0.6362    0.3181   -0.1590    0.4241         0
    0.1546    0.6955    0.3477   -0.1739    0.4636         0
    0.1679    0.7554    0.3777   -0.1888    0.5036         0
    0.1812    0.8154    0.4077   -0.2039    0.5436         0
    0.1945    0.8753    0.4377   -0.2188    0.5836         0
    0.2077    0.9346    0.4673   -0.2337    0.6231         0
    0.2207    0.9929    0.4965   -0.2482    0.6620         0
    0.2333    1.0499    0.5249   -0.2625    0.6999         0
    0.2456    1.1051    0.5525   -0.2763    0.7367         0
    0.2574    1.1582    0.5791   -0.2895    0.7721         0
    0.2687    1.2088    0.6044   -0.3022    0.8059         0
    0.2793    1.2568    0.6284   -0.3142    0.8379         0
    0.2893    1.3018    0.6509   -0.3254    0.8678         0
    0.2986    1.3435    0.6718   -0.3359    0.8957         0
    0.3071    1.3818    0.6909   -0.3455    0.9212         0
    0.3148    1.4165    0.7083   -0.3541    0.9444         0
    0.3217    1.4476    0.7238   -0.3619    0.9650         0
    0.3278    1.4748    0.7374   -0.3687    0.9832         0
    0.3330    1.4983    0.7492   -0.3746    0.9989         0
    0.3374    1.5181    0.7590   -0.3795    1.0121         0
    0.3410    1.5343    0.7671   -0.3836    1.0228         0
    0.3438    1.5470    0.7735   -0.3867    1.0313         0
    0.3459    1.5566    0.7783   -0.3891    1.0377         0
    0.3474    1.5633    0.7816   -0.3908    1.0422         0
    0.3484    1.5675    0.7838   -0.3919    1.0450         0
    0.3489    1.5698    0.7849   -0.3924    1.0465         0
    0.3491    1.5707    0.7853   -0.3927    1.0471         0
    0.3491    1.5708    0.7854   -0.3927    1.0472         0


TR3 =
```

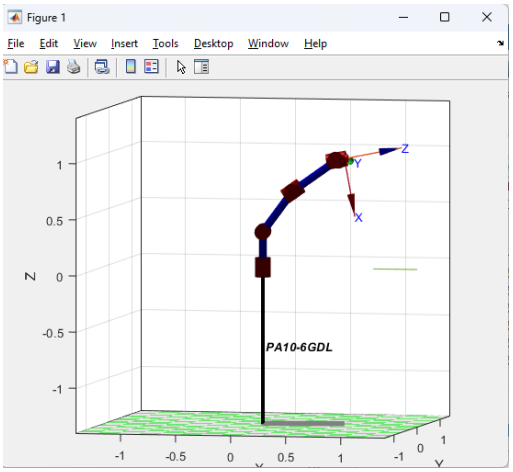
```
110
111         0         0         0         0         0         0
112         0    0.0001    0.0001         0   -0.0001         0
113         0    0.0005    0.0010         0   -0.0005         0
114         0    0.0016    0.0033         0   -0.0016         0
115         0    0.0038    0.0075         0   -0.0038         0
116         0    0.0071    0.0142         0   -0.0071         0
117         0    0.0119    0.0238         0   -0.0119         0
118         0    0.0183    0.0365         0   -0.0183         0
119         0    0.0264    0.0527         0   -0.0264         0
120         0    0.0362    0.0725         0   -0.0362         0
121         0    0.0480    0.0960         0   -0.0480         0
122         0    0.0616    0.1232         0   -0.0616         0
123         0    0.0771    0.1543         0   -0.0771         0
124         0    0.0945    0.1890         0   -0.0945         0
125         0    0.1136    0.2273         0   -0.1136         0
126         0    0.1345    0.2690         0   -0.1345         0
127         0    0.1570    0.3140         0   -0.1570         0
128         0    0.1810    0.3620         0   -0.1810         0
129         0    0.2063    0.4126         0   -0.2063         0
130         0    0.2329    0.4657         0   -0.2329         0
131         0    0.2605    0.5209         0   -0.2605         0
132         0    0.2889    0.5779         0   -0.2889         0
133         0    0.3181    0.6362         0   -0.3181         0
134         0    0.3477    0.6955         0   -0.3477         0
135         0    0.3777    0.7554         0   -0.3777         0
136         0    0.4077    0.8154         0   -0.4077         0
137         0    0.4377    0.8753         0   -0.4377         0
138         0    0.4673    0.9346         0   -0.4673         0
139         0    0.4965    0.9929         0   -0.4965         0
140         0    0.5249    1.0499         0   -0.5249         0
141         0    0.5525    1.1051         0   -0.5525         0
142         0    0.5791    1.1582         0   -0.5791         0
143         0    0.6044    1.2088         0   -0.6044         0
144         0    0.6284    1.2568         0   -0.6284         0
145         0    0.6509    1.3018         0   -0.6509         0
146         0    0.6718    1.3435         0   -0.6718         0
147         0    0.6909    1.3818         0   -0.6909         0
148         0    0.7083    1.4165         0   -0.7083         0
149         0    0.7238    1.4476         0   -0.7238         0
150         0    0.7374    1.4748         0   -0.7374         0
151         0    0.7492    1.4983         0   -0.7492         0
152         0    0.7590    1.5181         0   -0.7590         0
153         0    0.7671    1.5343         0   -0.7671         0
154         0    0.7735    1.5470         0   -0.7735         0
155         0    0.7783    1.5566         0   -0.7783         0
156         0    0.7816    1.5633         0   -0.7816         0
157         0    0.7838    1.5675         0   -0.7838         0
158         0    0.7849    1.5698         0   -0.7849         0
159         0    0.7853    1.5707         0   -0.7853         0
160         0    0.7854    1.5708         0   -0.7854         0
```



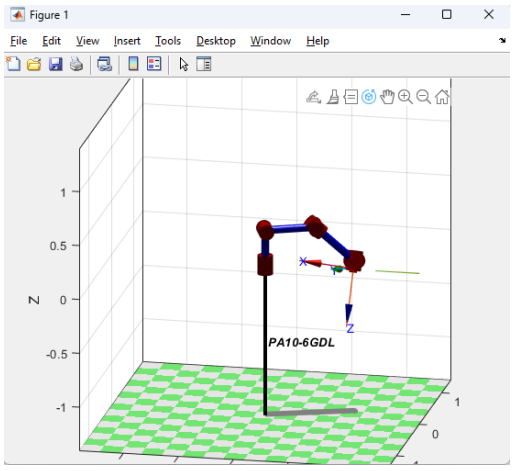
(a) Posición Home



(b) Posición 1



(c) Posición 2



(d) Posición 3

Figura 5: Posiciones

10. Ejercicio 10

10.1. Enunciado

Ejercicio práctico 10: Inserta el comando sl_lanechange en la línea de comandos de Matlab para abrir el archivo Simulink. Ejecuta dicho archivo y visualiza la entrada de dirección (Steering angle), así como el valor del ángulo (theta). Cambia los valores máximos/mínimos de la dicha entrada y visualiza los cambios en el visor XY. ¿Qué es lo que representa esta gráfica XY? Cambia los parámetros del bloque Bicycle y visualiza los cambios en la posición del vehículo.

10.2. Resolución

```
1 sl_lanechange
```

10.3. Resultado Obtenido

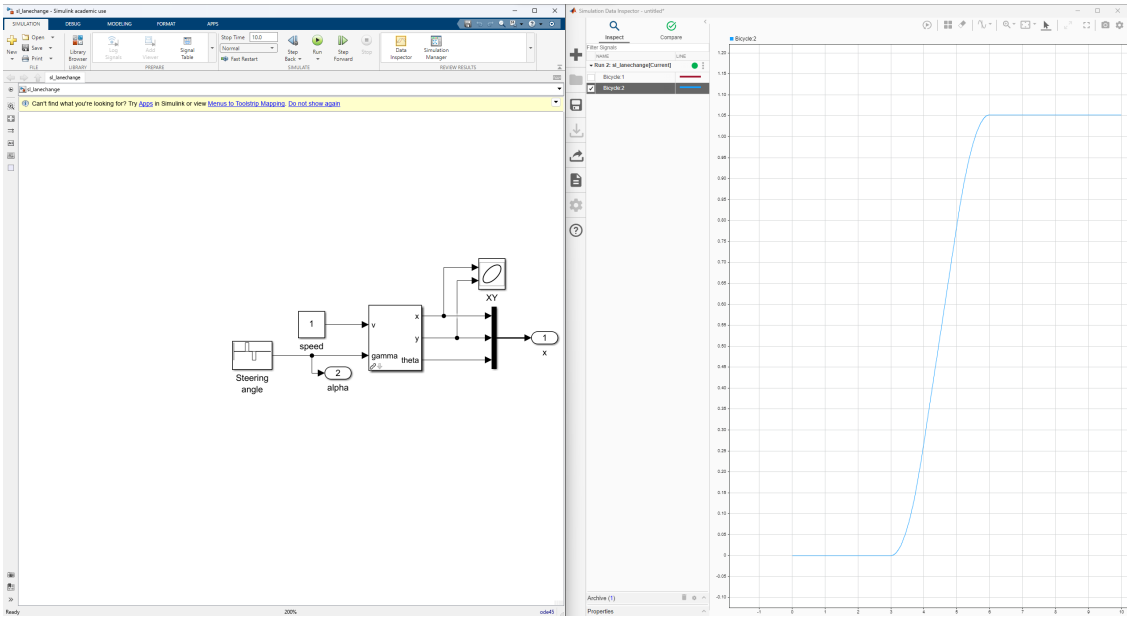


Figura 6: Sin modificar

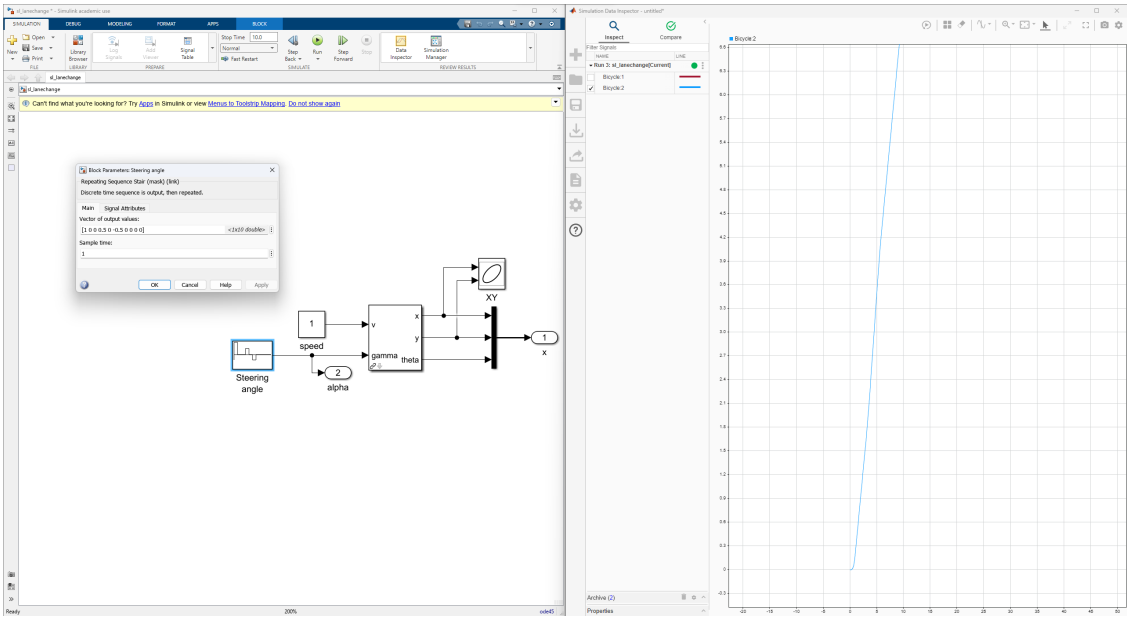


Figura 7: Sin modificar

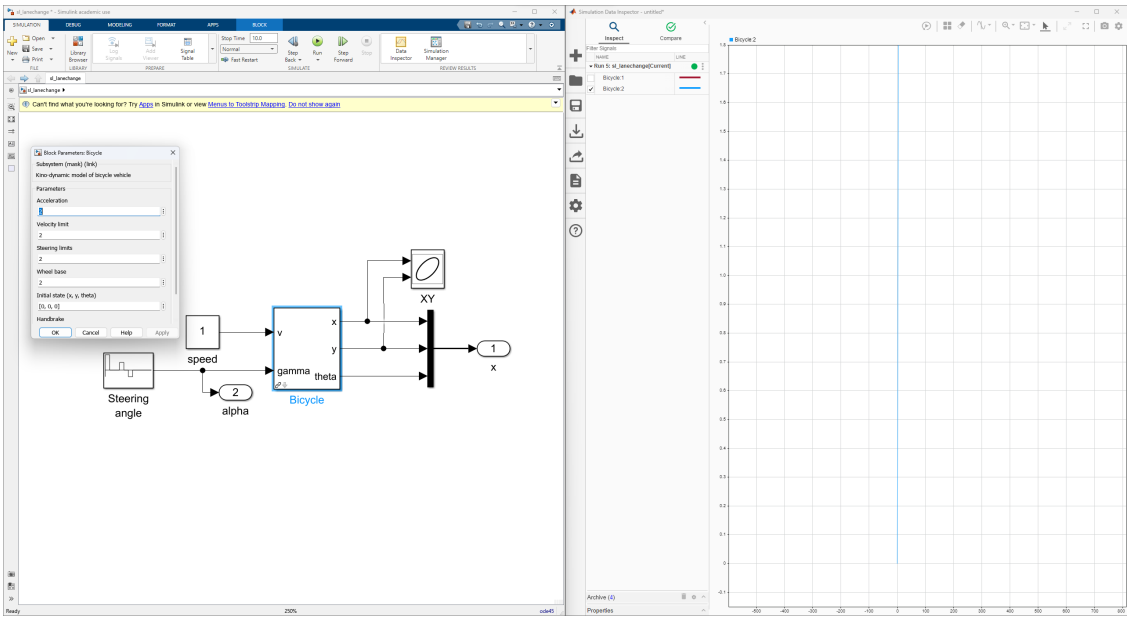


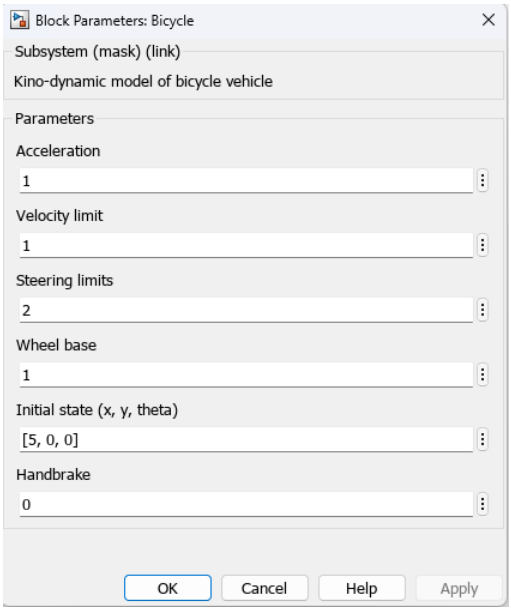
Figura 8: Sin modificar

11. Ejercicio 11

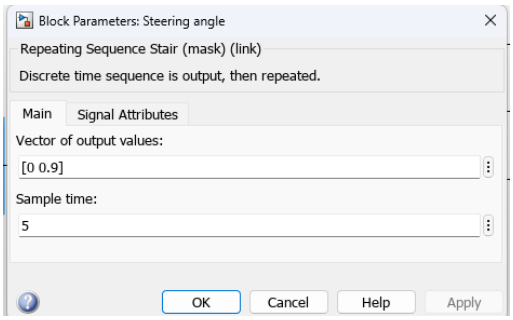
11.1. Enunciado

Ejercicio práctico 11: 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. Resolución



(a) Valores cambiados



(b) Valores cambiados

Figura 9: Posiciones

11.3. Resultado Obtenido

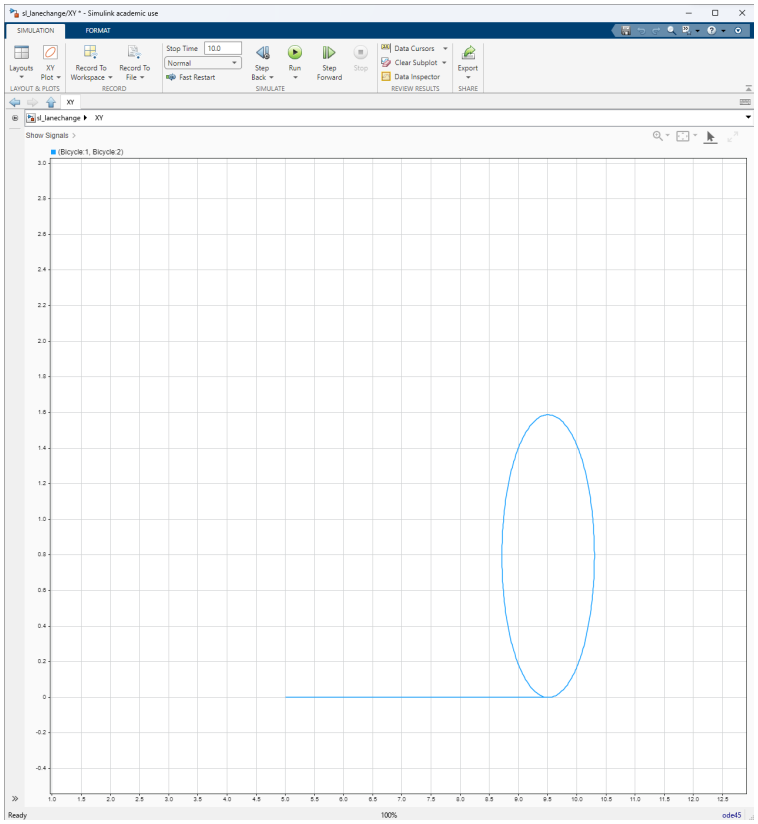


Figura 10: Obtenido