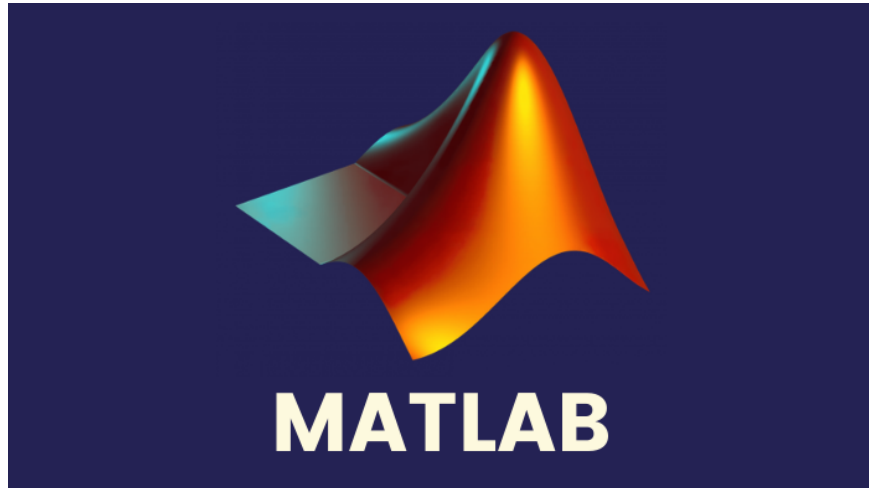


## Ejercicios de Robótica

- Alumno: ELVI MIHAI SABAU SABAU



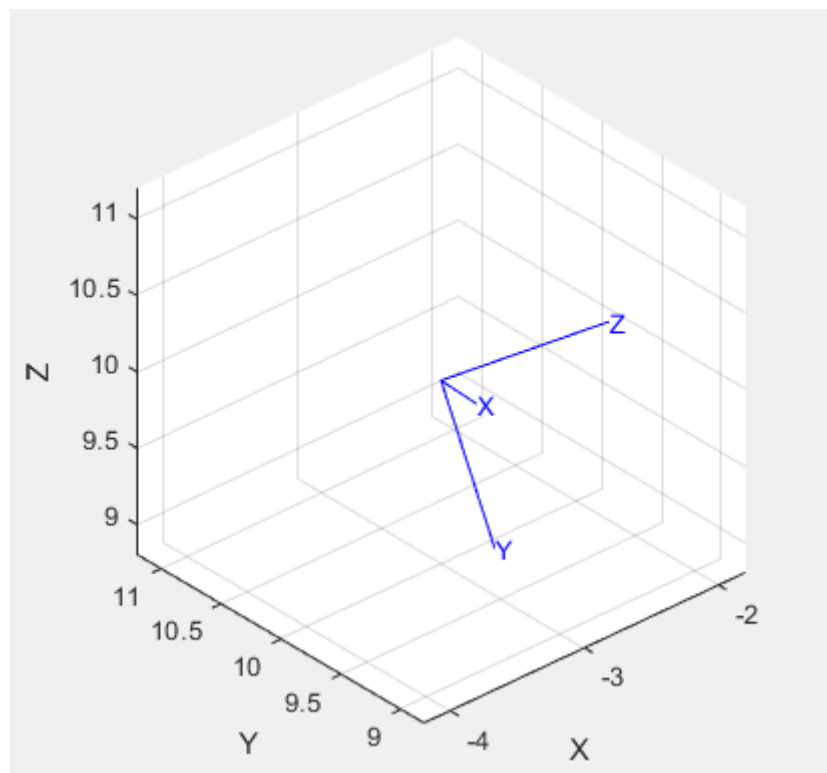
Ejercicio 1.	2
Ejercicio 2.	3
Ejercicio 3.	5
Ejercicio 4.	7
Ejercicio 5.	9
Ejercicio 6.	11
Ejercicio 7.	11

## Ejercicio 1.

**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  $-90^\circ$  sobre el eje O'U del sistema trasladado y giro de  $90^\circ$  sobre el eje O'V' del sistema girado.

ans =

0	0	1	-3
-1	0	0	10
0	-1	0	10
0	0	0	1

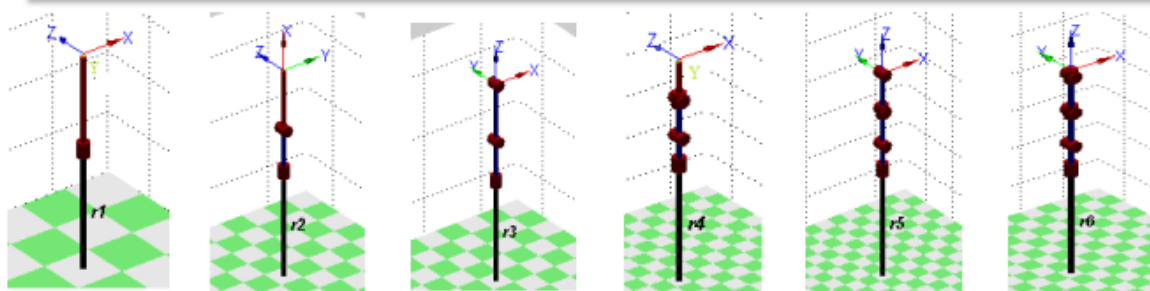


## Ejercicio 2.

**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*” <http://www.petercorke.com/RTB/r9/html/Link.html>.

Transformación	$\Theta$	$d$	$a$	$\alpha$	Límite $q(^{\circ})$	Offset
$0 \rightarrow 1 \quad {}^0A_1$	$q_1$	0.317	0	$-\pi/2$	$[-177,177]$	0
$1 \rightarrow 2 \quad {}^1A_2$	$q_2$	0	0.45	0	$[-64,124]$	$-\pi/2$
$2 \rightarrow 3 \quad {}^2A_3$	$q_3$	0	0	$\pi/2$	$[-107,158]$	$\pi/2$
$3 \rightarrow 4 \quad {}^3A_4$	$q_4$	0.48	0	$-\pi/2$	$[-255,255]$	0
$4 \rightarrow 5 \quad {}^4A_5$	$q_5$	0	0	$\pi/2$	$[-165,165]$	0
$5 \rightarrow 6 \quad {}^5A_6$	$q_6$	0.07	0	0	$[-255,255]$	0

Nota: para poder visualizar el funcionamiento los parámetros DH, podéis ir creando objetos robot con 1, 2, 3,...hasta 6 eslabones e ir visualizando el cambio en los sistemas de referencia.



`r =`

PA10-6GDL:: 6 axis, RRRRRR, stdDH, slowRNE

j	theta	d	a	alpha	offset
1	q1	0.317	0	-1.5708	0
2	q2	0	0.45	0	-1.5708
3	q3	0	0	1.5708	1.5708
4	q4	0.48	0	-1.5708	0
5	q5	0	0	1.5708	0
6	q6	0.07	0	0	0

`ans =`

6

`ans =`

```

Revolute(std): theta=q1 d=0.317 a=0 alpha=-1.571 offset=0
Revolute(std): theta=q2 d=0 a=0.45 alpha=0 offset=-1.571
Revolute(std): theta=q3 d=0 a=0 alpha=1.571 offset=1.571
Revolute(std): theta=q4 d=0.48 a=0 alpha=-1.571 offset=0
Revolute(std): theta=q5 d=0 a=0 alpha=1.571 offset=0
Revolute(std): theta=q6 d=0.07 a=0 alpha=0 offset=0

```

## Ejercicio 3.

**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.

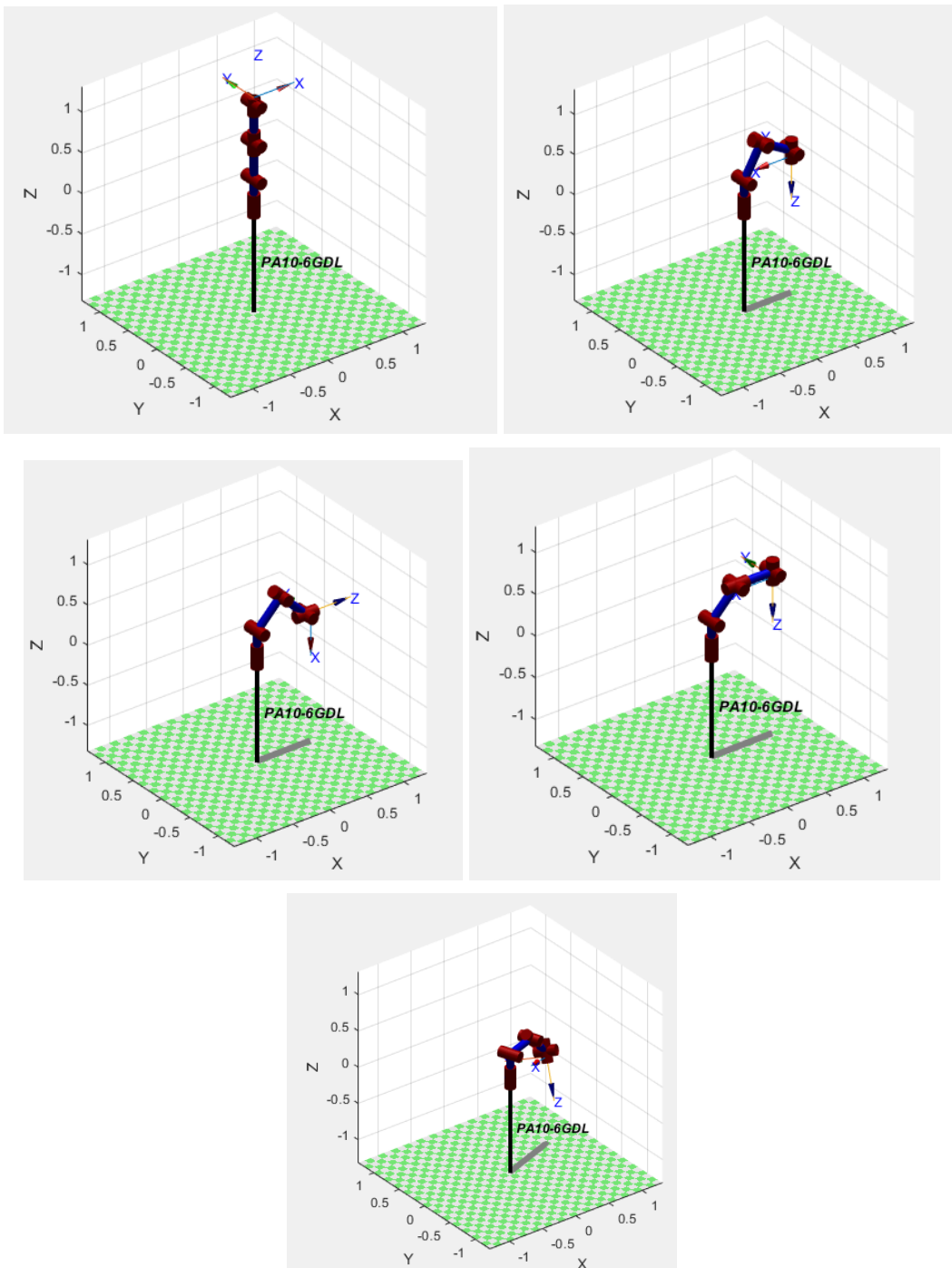
Posición de home:  $q_h = [0, 0, 0, 0, 0, 0]$ .

Posición de escape:  $q_e = [0, 30, 90, 0, 60, 0]$ .

Posición de seguridad:  $q_s = [0, 45, 90, 0, -45, 0]$ .

Posición  $q_1 = [0, 45, 45, 0, 90, 0]$ .

Posición  $q_2 = [20, 90, 45, -22.5, 60, 0]$ .



```

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 =
   -0.0000   -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

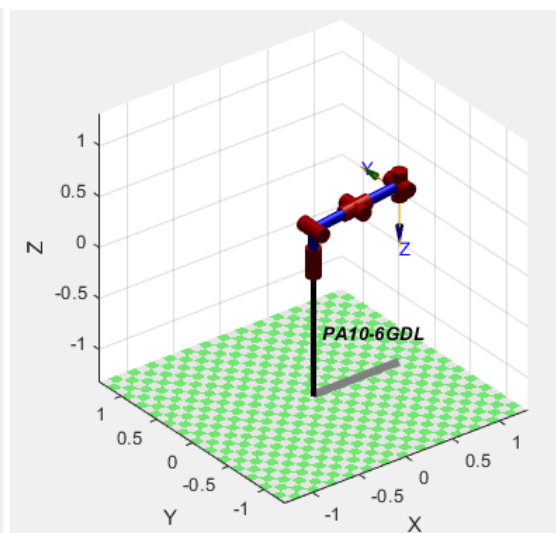
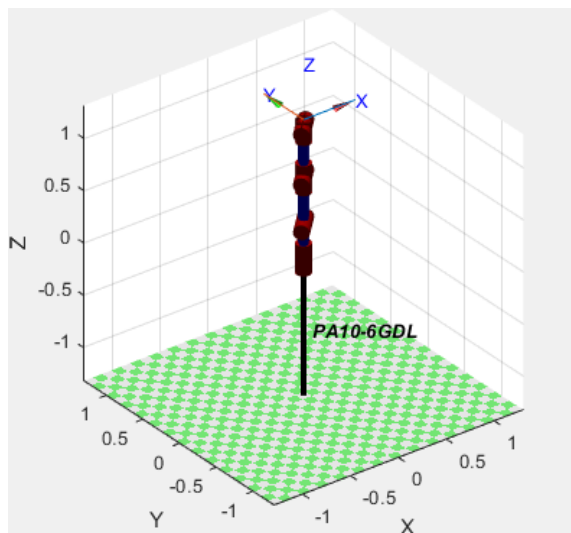
q1Q =
   -1.0000    0.0000   -0.0000    0.7982
   -0.0000    1.0000    0.0000    0.0000
    0.0000    0.0000   -1.0000    0.5652
         0         0         0    1.0000

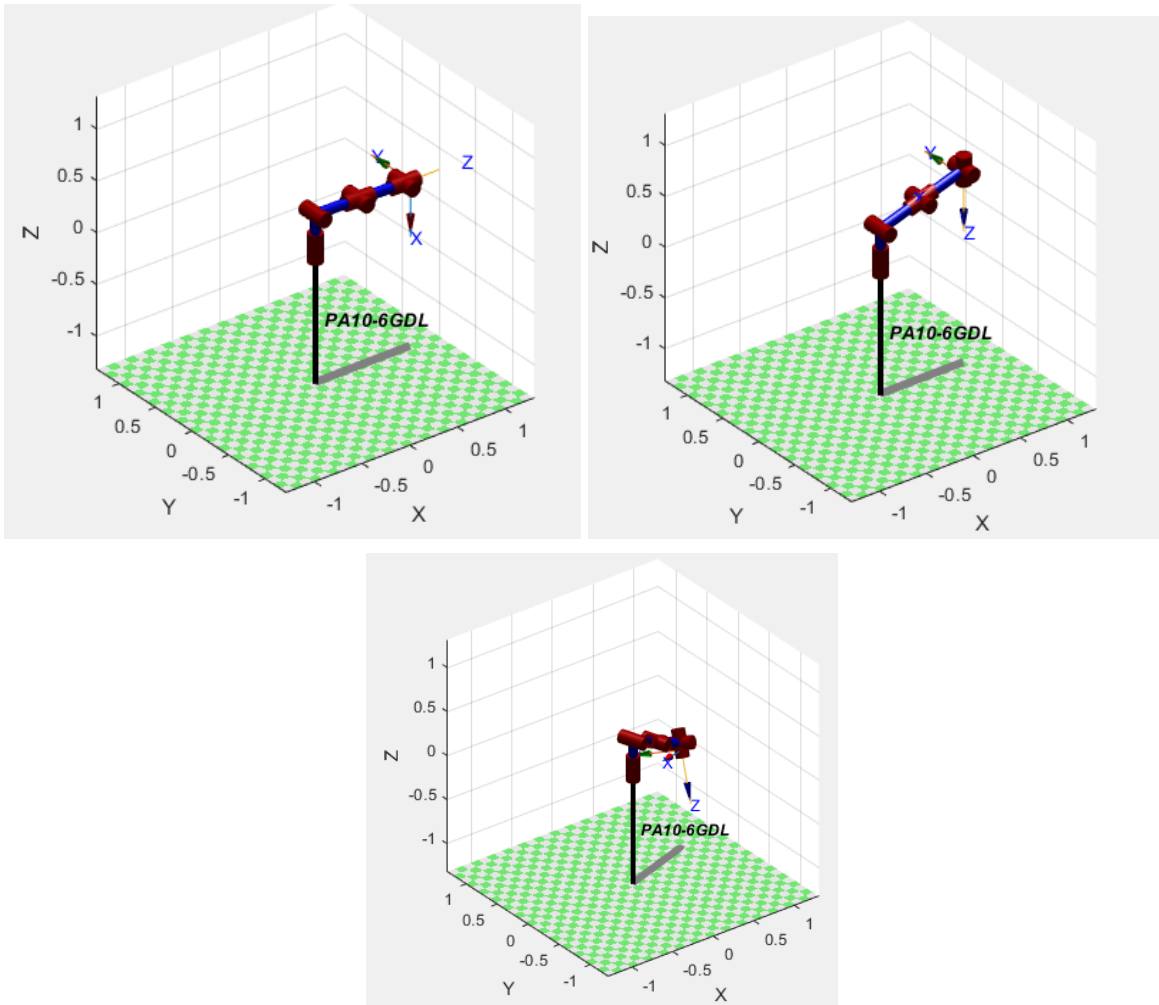
q2Q =
   -0.8169   -0.5703   -0.0861    0.7358
   -0.5010    0.7756   -0.3840    0.2431
    0.2857   -0.2706   -0.9193   -0.0868
         0         0         0    1.0000

```

## Ejercicio 4.

**Ejercicio práctico 4:** realizar la resolución de la cinemática inversa para el resto de posiciones del PA10 ( $q_e$ ,  $q_s$ ,  $q_1$ ,  $q_2$ ) 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.





T =				T1 =			
1.0000	0	0	0.0000	-1.0000	0.0000	-0.0000	0.6407
0	1.0000	0	-0.0000	-0.0000	1.0000	0.0000	0.0000
0	0	1.0000	1.3170	0.0000	0.0000	-1.0000	0.3967
0	0	0	1.0000	0	0	0	1.0000
qinversa =				qinversal =			
2.3562	0	-0.0000	-3.1416	-3.1416	-1.4470	-0.0000	0.0000
			-0.0000			-1.6946	-3.1416
Tinversa =				Tinversal =			
1.0000	0	-0.0000	0.0000	-1.0000	-0.0000	-0.0000	0.9229
0	1.0000	-0.0000	-0.0000	-0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	1.0000	1.3170	0.0000	0.0000	-1.0000	0.3618
0	0	0	1.0000	0	0	0	1.0000

T2 =

-0.0000	-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

T3 =

-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

qinversa2 =

-3.1416	-1.5999	0	-3.1416	-0.0291	0.0000	-3.1416	-1.2693	-0.0000	0.0000	-1.8723	-3.1416
---------	---------	---	---------	---------	--------	---------	---------	---------	--------	---------	---------

qinversa3 =

Tinversa2 =

-0.0000	-0.0000	1.0000	0.9996
0.0000	1.0000	0.0000	0.0000
-1.0000	0.0000	-0.0000	0.2899
0	0	0	1.0000

Tinversa3 =

-1.0000	-0.0000	-0.0000	0.8881
-0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	-1.0000	0.5231
0	0	0	1.0000

T4 =

-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

qinversa4 =

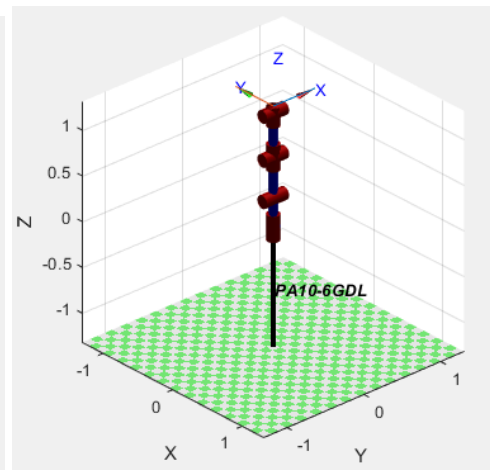
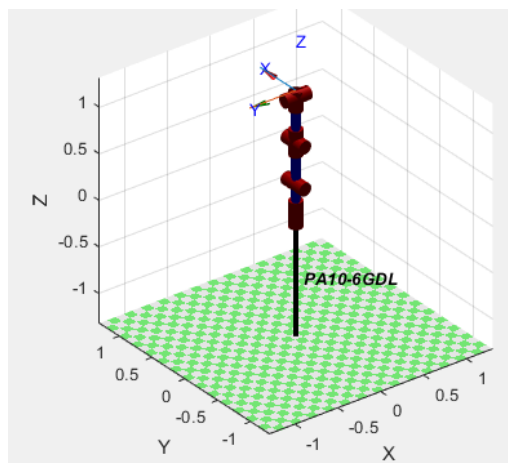
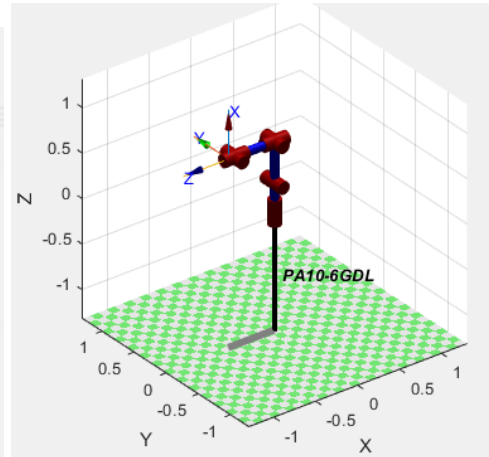
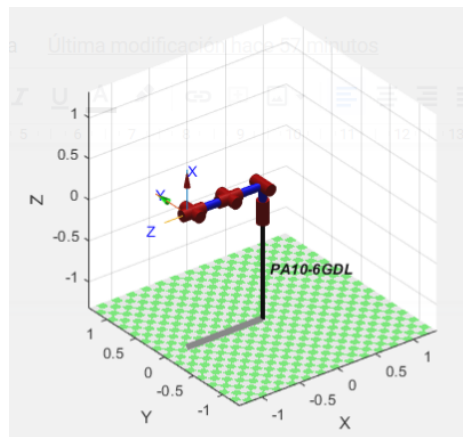
-2.8224	-2.0511	-0.0000	-0.3557	-1.3228	2.9973
---------	---------	---------	---------	---------	--------

Tinversa4 =

-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	1.0000

## Ejercicio 5.

**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).



```

1
-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



2						
	-0.0000	0.4500	0.0000	0	0.0000	0
	-0.5500	-0.0000	-0.0000	0	-0.0000	0
	0	0.5500	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					
3						
	0.0000	1.0000	0.5500	0	0.0000	0
	0.0000	0	0	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
						2.8735e-66
4						
	0	0	0	0	0	1.5708
	0.0000	1.0000	0.5500	0	0.0700	0
	0.0000	0	0	0	0	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
						1.0717e-82

## Ejercicio 6.

**Ejercicio práctico 6:** calcula los pares articulares del resto de posiciones del robot PA10 ( $q_3$ ,  $q_4$  y  $q_5$ ) 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`).

ans =

```
0.1068 -0.0000 -0.0000
-0.0000 0.0528 -0.0000
-0.0000 -0.0000 0.1308
```

ans =

```
0.3783 -0.0201 0.0003
-0.0201 0.0641 0.0009
0.0003 0.0009 0.3758
```

ans =

```
0.0599 0.0000 0.0000
0.0000 0.0577 -0.0018
0.0000 -0.0018 0.0084
```

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
```

ans =

```
0.0283 0.0000 0.0000
0.0000 0.0109 -0.0000
0.0000 -0.0000 0.0199
```

ans =

```
0.0074 0 -0.0000
0 0.0070 0
-0.0000 0 0.0030
```

ans =

```
0.0014 0 0
0 0.0014 0
0 0 0.0006
```

## Ejercicio 7.

**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 ( $g=1,62 \text{ m/s}^2$ ). Justifica los resultados.

$r =$

PA10 (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
        1.62           0  0  1  0           0  0  1  0
                   0  0  0  1           0  0  0  1

```

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

## Ejercicio 8.

**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.

Res0 =

0 -75.6123 -28.0259 -0.0000 -3.3164 0

Res1 =

0 -82.5309 -34.9445 0.0000 -0.0000 0

Res2 =

-0.0000 -94.9612 -24.0058 -0.7772 0.9476 0

Vqs\_0 =

4.5481 0.0009 0.0013 -0.2248 0.0000 0.0000

Vq1\_0 =

5.4613 -0.0004 -0.0000 0.2698 0.0000 -0.0006

Vq2\_0 =

5.8204 -0.0601 -0.0593 0.1832 -0.0739 -0.0005

Como podemos observar a continuación, al agregar peso al robot, cambian los parámetros de gravedad.

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 =

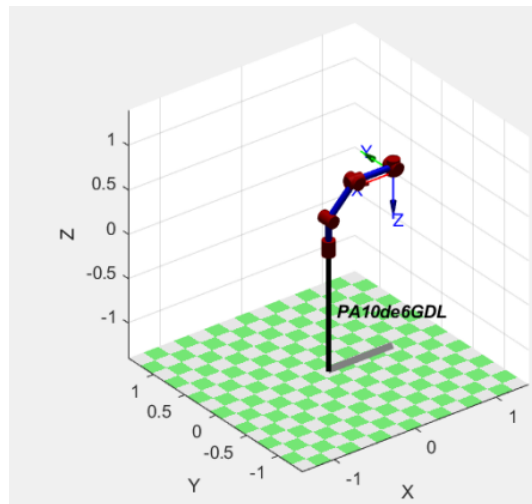
4.0908	-0.0004	-0.0000	0.0573	-0.0000	0.3722
--------	---------	---------	--------	---------	--------

Vq2\_1 =

4.6529	-0.1244	-0.1235	0.1522	0.0840	0.2758
--------	---------	---------	--------	--------	--------

## Ejercicio 9.

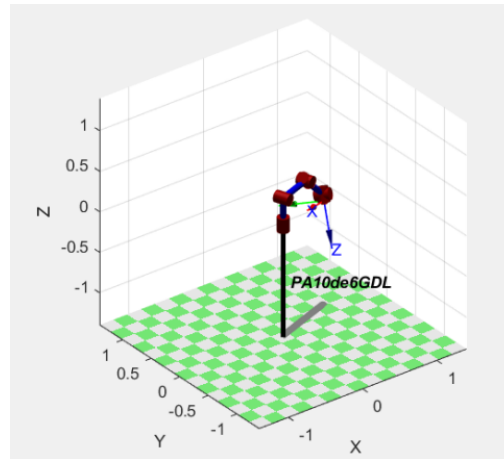
**Ejercicio práctico 9:** Realiza 3 trayectorias articulares con el robot PA10 entre diferentes puntos probando el perfil trapezoidal y polinomial. Para visualizar los valores de velocidad y aceleración puedes emplear el comando `plot(qd)`. Realiza 3 trayectorias cartesianas con el robot PA10 cambiando 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  $J$ ).



v1 =

0	0	0	0	0	0	0	0	0.4377	0.4377	0	0.8753	0
0	0.0001	0.0001	0	0.0001	0	0	0.4673	0.4673	0	0.9346	0	0
0	0.0005	0.0005	0	0.0010	0	0	0.4965	0.4965	0	0.9929	0	0
0	0.0016	0.0016	0	0.0033	0	0	0.5249	0.5249	0	1.0499	0	0
0	0.0038	0.0038	0	0.0075	0	0	0.5525	0.5525	0	1.1051	0	0
0	0.0071	0.0071	0	0.0142	0	0	0.5791	0.5791	0	1.1582	0	0
0	0.0119	0.0119	0	0.0238	0	0	0.6044	0.6044	0	1.2088	0	0
0	0.0183	0.0183	0	0.0365	0	0	0.6284	0.6284	0	1.2568	0	0
0	0.0264	0.0264	0	0.0527	0	0	0.6509	0.6509	0	1.3018	0	0
0	0.0362	0.0362	0	0.0725	0	0	0.6718	0.6718	0	1.3435	0	0
0	0.0480	0.0480	0	0.0960	0	0	0.6909	0.6909	0	1.3818	0	0
0	0.0616	0.0616	0	0.1232	0	0	0.7083	0.7083	0	1.4165	0	0
0	0.0771	0.0771	0	0.1543	0	0	0.7238	0.7238	0	1.4476	0	0
0	0.0945	0.0945	0	0.1890	0	0	0.7374	0.7374	0	1.4748	0	0
0	0.1136	0.1136	0	0.2273	0	0	0.7492	0.7492	0	1.4983	0	0
0	0.1345	0.1345	0	0.2690	0	0	0.7590	0.7590	0	1.5181	0	0
0	0.1570	0.1570	0	0.3140	0	0	0.7671	0.7671	0	1.5343	0	0
0	0.1810	0.1810	0	0.3620	0	0	0.7735	0.7735	0	1.5470	0	0
0	0.2063	0.2063	0	0.4126	0	0	0.7783	0.7783	0	1.5566	0	0
0	0.2329	0.2329	0	0.4657	0	0	0.7816	0.7816	0	1.5633	0	0
0	0.2605	0.2605	0	0.5209	0	0	0.7838	0.7838	0	1.5675	0	0
0	0.2889	0.2889	0	0.5779	0	0	0.7849	0.7849	0	1.5698	0	0
0	0.3181	0.3181	0	0.6362	0	0	0.7853	0.7853	0	1.5707	0	0
0	0.3477	0.3477	0	0.6955	0	0	0.7854	0.7854	0	1.5708	0	0
0	0.3777	0.3777	0	0.7554	0	0						
0	0.4077	0.4077	0	0.8154	0	0						



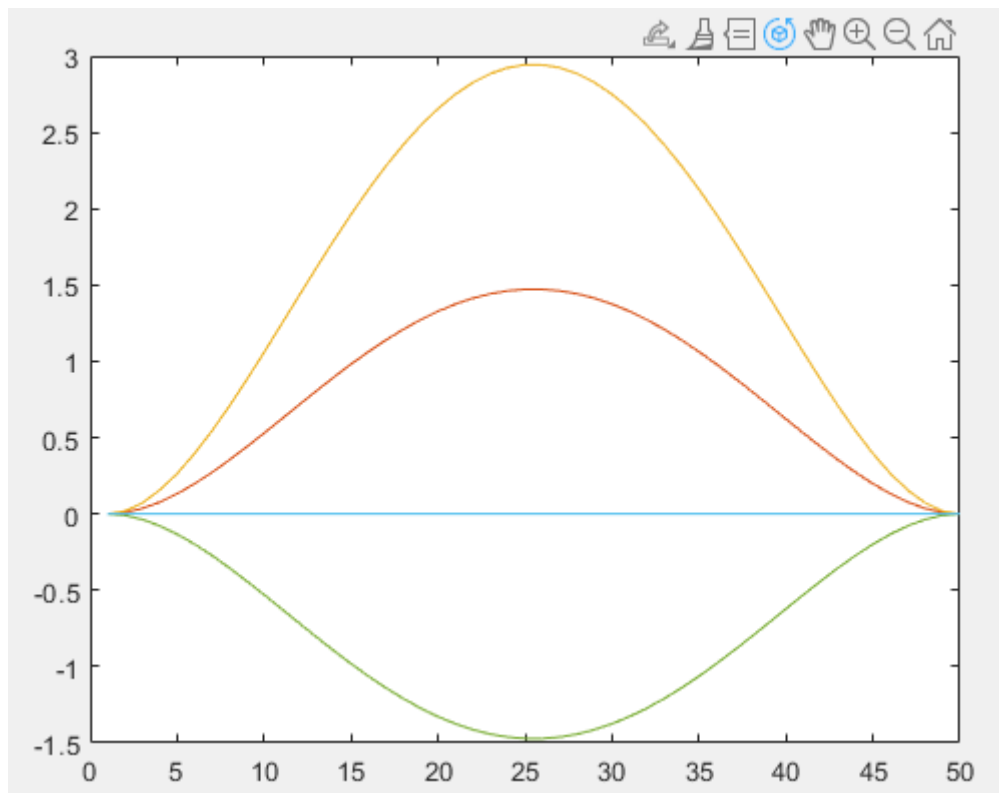


V2 =

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

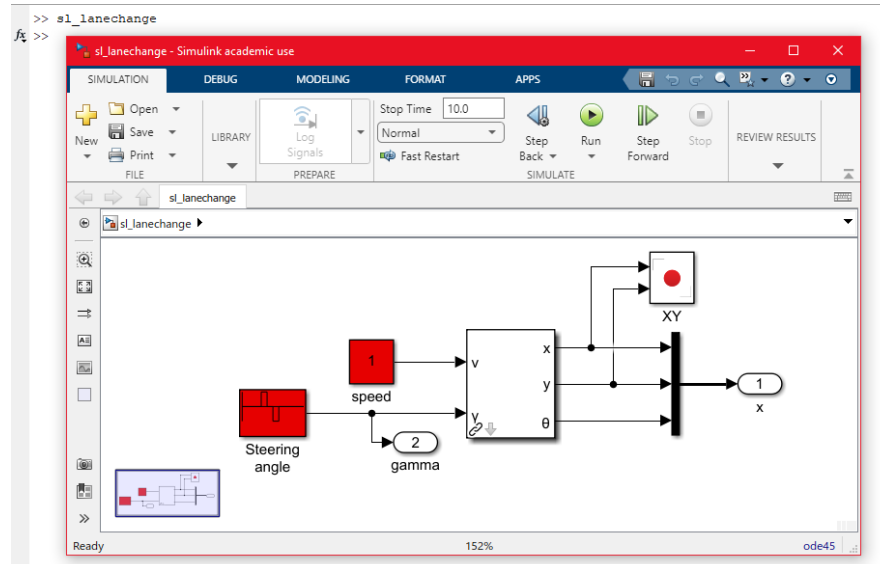


0	0	0	0	0	0	0.4673	0.9346	0	-0.4673	0	
0	0.0001	0.0001	0	-0.0001	0	0	0.4965	0.9929	0	-0.4965	0
0	0.0005	0.0010	0	-0.0005	0	0	0.5249	1.0499	0	-0.5249	0
0	0.0016	0.0033	0	-0.0016	0	0	0.5525	1.1051	0	-0.5525	0
0	0.0038	0.0075	0	-0.0038	0	0	0.5791	1.1582	0	-0.5791	0
0	0.0071	0.0142	0	-0.0071	0	0	0.6044	1.2088	0	-0.6044	0
0	0.0119	0.0238	0	-0.0119	0	0	0.6284	1.2568	0	-0.6284	0
0	0.0183	0.0365	0	-0.0183	0	0	0.6509	1.3018	0	-0.6509	0
0	0.0264	0.0527	0	-0.0264	0	0	0.6718	1.3435	0	-0.6718	0
0	0.0362	0.0725	0	-0.0362	0	0	0.6909	1.3818	0	-0.6909	0
0	0.0480	0.0960	0	-0.0480	0	0	0.7083	1.4165	0	-0.7083	0
0	0.0616	0.1232	0	-0.0616	0	0	0.7238	1.4476	0	-0.7238	0
0	0.0771	0.1543	0	-0.0771	0	0	0.7374	1.4748	0	-0.7374	0
0	0.0945	0.1890	0	-0.0945	0	0	0.7492	1.4983	0	-0.7492	0
0	0.1136	0.2273	0	-0.1136	0	0	0.7590	1.5181	0	-0.7590	0
0	0.1345	0.2690	0	-0.1345	0	0	0.7671	1.5343	0	-0.7671	0
0	0.1570	0.3140	0	-0.1570	0	0	0.7735	1.5470	0	-0.7735	0
0	0.1810	0.3620	0	-0.1810	0	0	0.7783	1.5566	0	-0.7783	0
0	0.2063	0.4126	0	-0.2063	0	0	0.7816	1.5633	0	-0.7816	0
0	0.2329	0.4657	0	-0.2329	0	0	0.7838	1.5675	0	-0.7838	0
0	0.2605	0.5209	0	-0.2605	0	0	0.7849	1.5698	0	-0.7849	0
0	0.2889	0.5779	0	-0.2889	0	0	0.7853	1.5707	0	-0.7853	0
0	0.3181	0.6362	0	-0.3181	0	0	0.7854	1.5708	0	-0.7854	0
0	0.3477	0.6955	0	-0.3477	0	0					
0	0.3777	0.7554	0	-0.3777	0	0					
0	0.4077	0.8154	0	-0.4077	0	0					

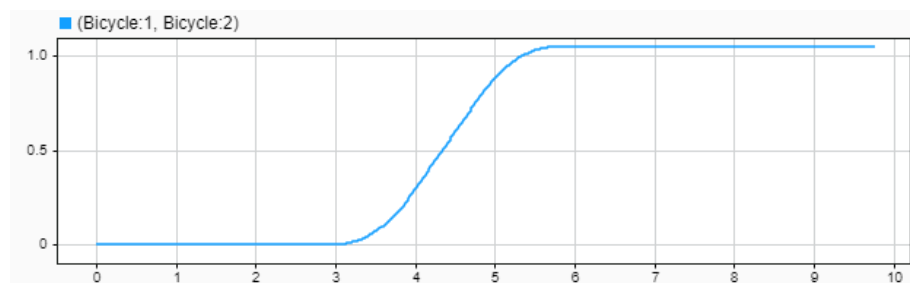


## Ejercicio 10.

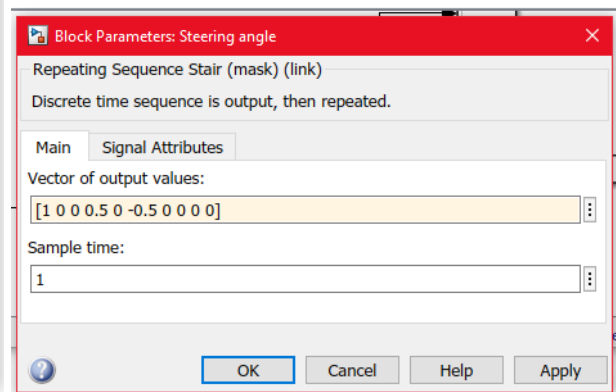
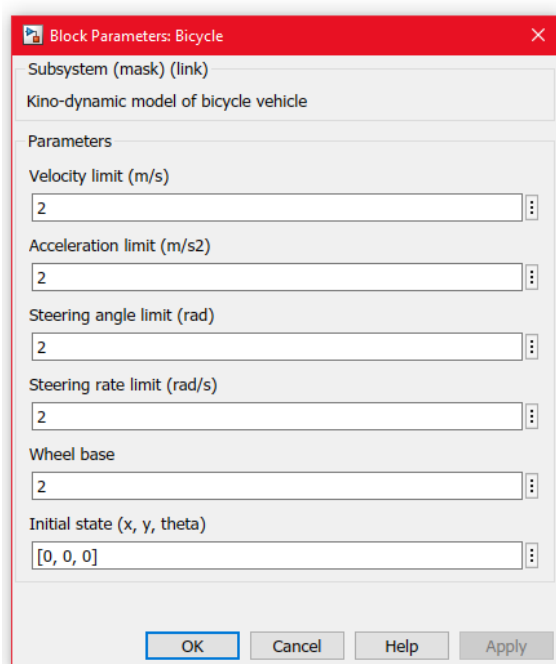
**Ejercicio práctico 10:** Inserta el comando `sl_lanechange` en la línea de comandos de Matlab para abrir el archivo *Simulink*. Ejecuta dicho archivo y visualiza la entrada de dirección (*Steering angle*), así como el valor del ángulo  $\theta$  (*theta*). Cambia los valores máximos/mínimos de la dicha 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.



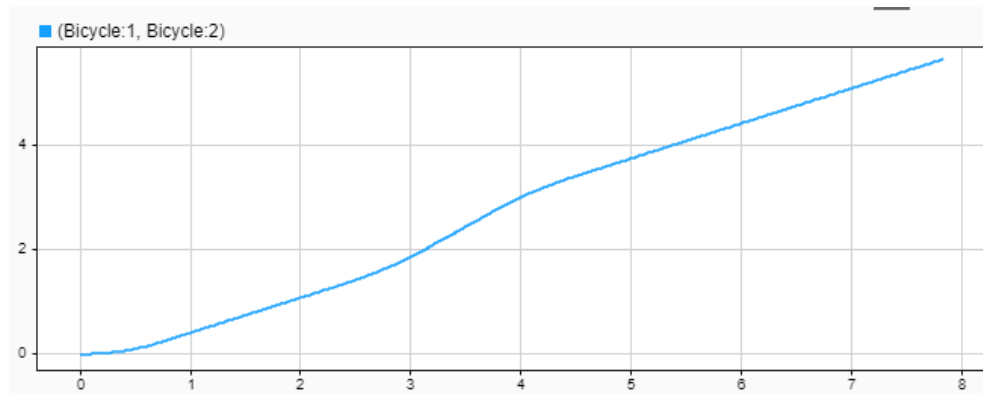
Sin cambiar los valores:



Cambiando los Valores max / min y de Steering Angle:



Resultado:



## Ejercicio 11.

**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?

