

Laboratorio 1 Robótica: Análisis Geométrico Directo de Manipuladores

Universidad Nacional de Colombia

Alejandro Ojeda Olarte, Brayan Calderon, Mateo Rodriguez

Parte I | Robot

El robot asignado es el modelo UR3 de la empresa Universal Robots.

Características y modelo del robot

1. Busque información técnica del robot, obtenga catálogos de fabricante, presente en el informe:

a) Imagen del robot.



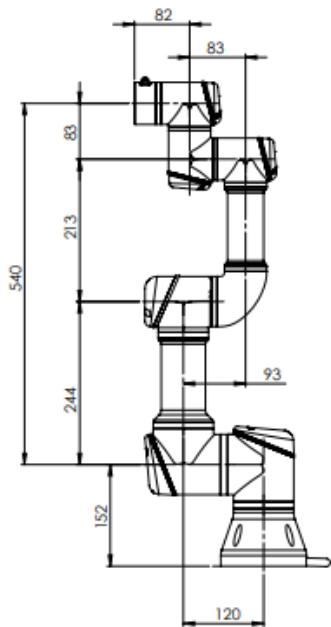
Universal Robot UR3

b) Capacidad de carga.

Carga Util: 3Kg

c) Alcance vertical y horizontal.

Tiene un alcance de 500mm tanto vertical como horizontal

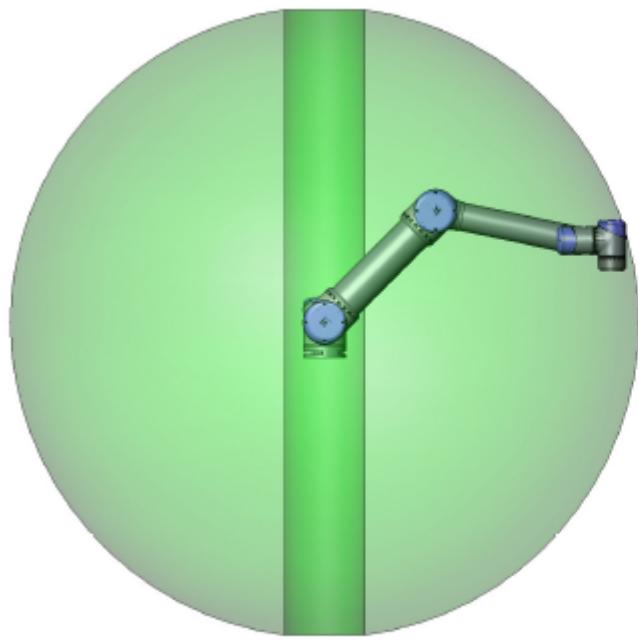


All dimension is in mm

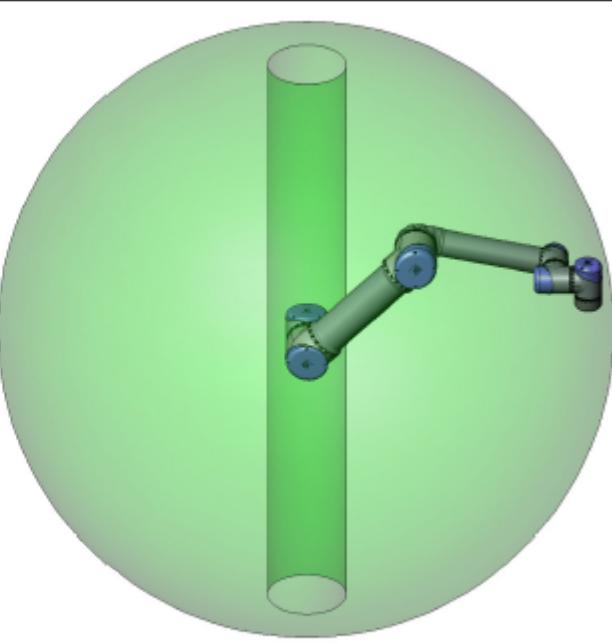
d) Repetibilidad.

Tiene una repetibilidad de +/- 0.1mm

e) Gráfica(s) de espacio alcanzable.



Front



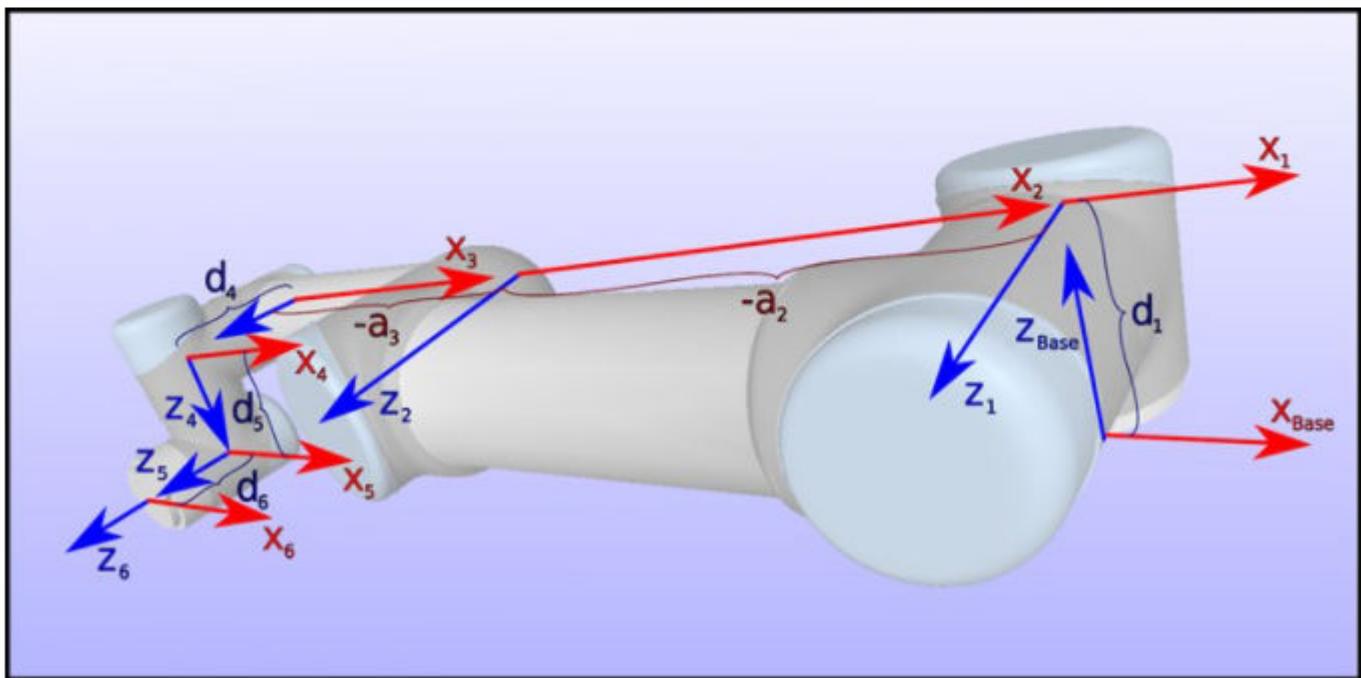
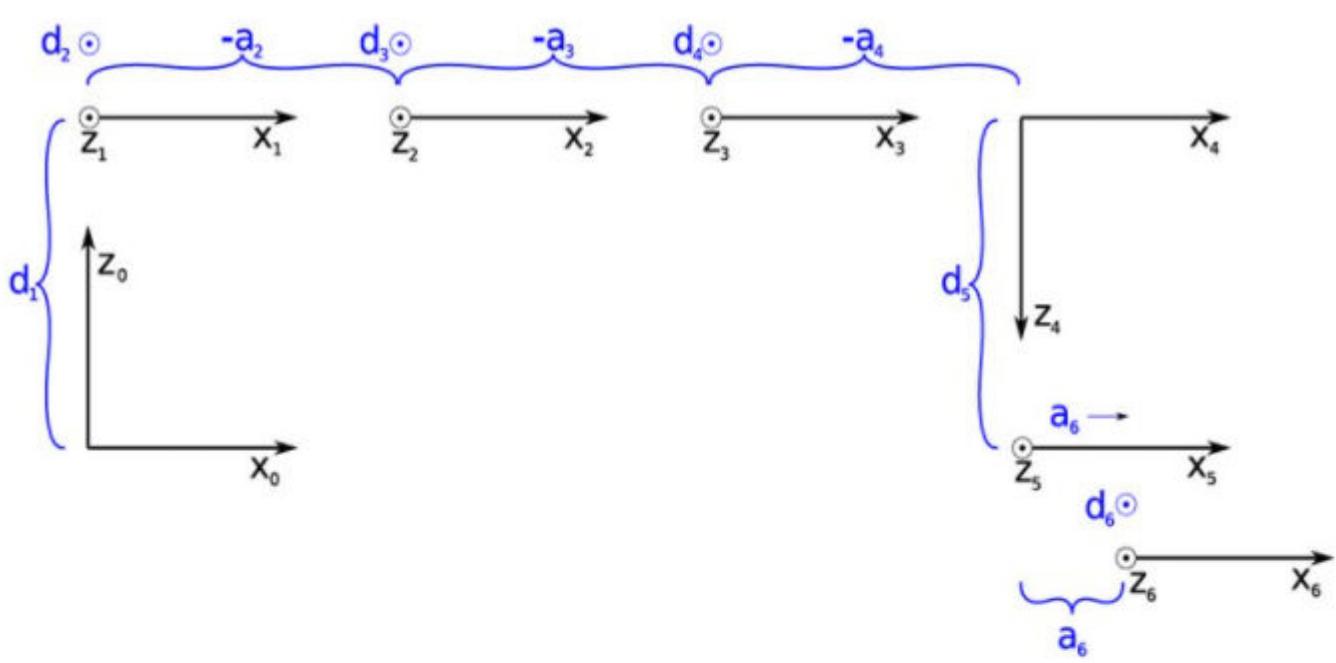
Tilted

Espacio Alcanzable por el UR3

f) Tabla de parámetros DH.

- La siguiente tabla corresponde a los parámetros DH estándar suministrados por Universal Robots

UR3							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	Dynamics	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.1519	$\pi/2$	Link 1	2	[0, -0.02, 0]
Joint 2	0	-0.24365	0	0	Link 2	3.42	[0.13, 0, 0.1157]
Joint 3	0	-0.21325	0	0	Link 3	1.26	[0.05, 0, 0.0238]
Joint 4	0	0	0.11235	$\pi/2$	Link 4	0.8	[0, 0, 0.01]
Joint 5	0	0	0.08535	$-\pi/2$	Link 5	0.8	[0, 0, 0.01]
Joint 6	0	0	0.0819	0	Link 6	0.35	[0, 0, -0.02]



g) Qué software utiliza el fabricante para diseño de celdas o programación.

Polyscope - <https://www.universal-robots.com/download/?query=>

h) ¿Qué otras características tiene? (Grado de protección IP, colaborativo, normas de seguridad etc.)

Grado de Protección IP64 IP54 (E67ON_Global). Es colaborativo y tiene 15 funciones avanzadas ajustables.

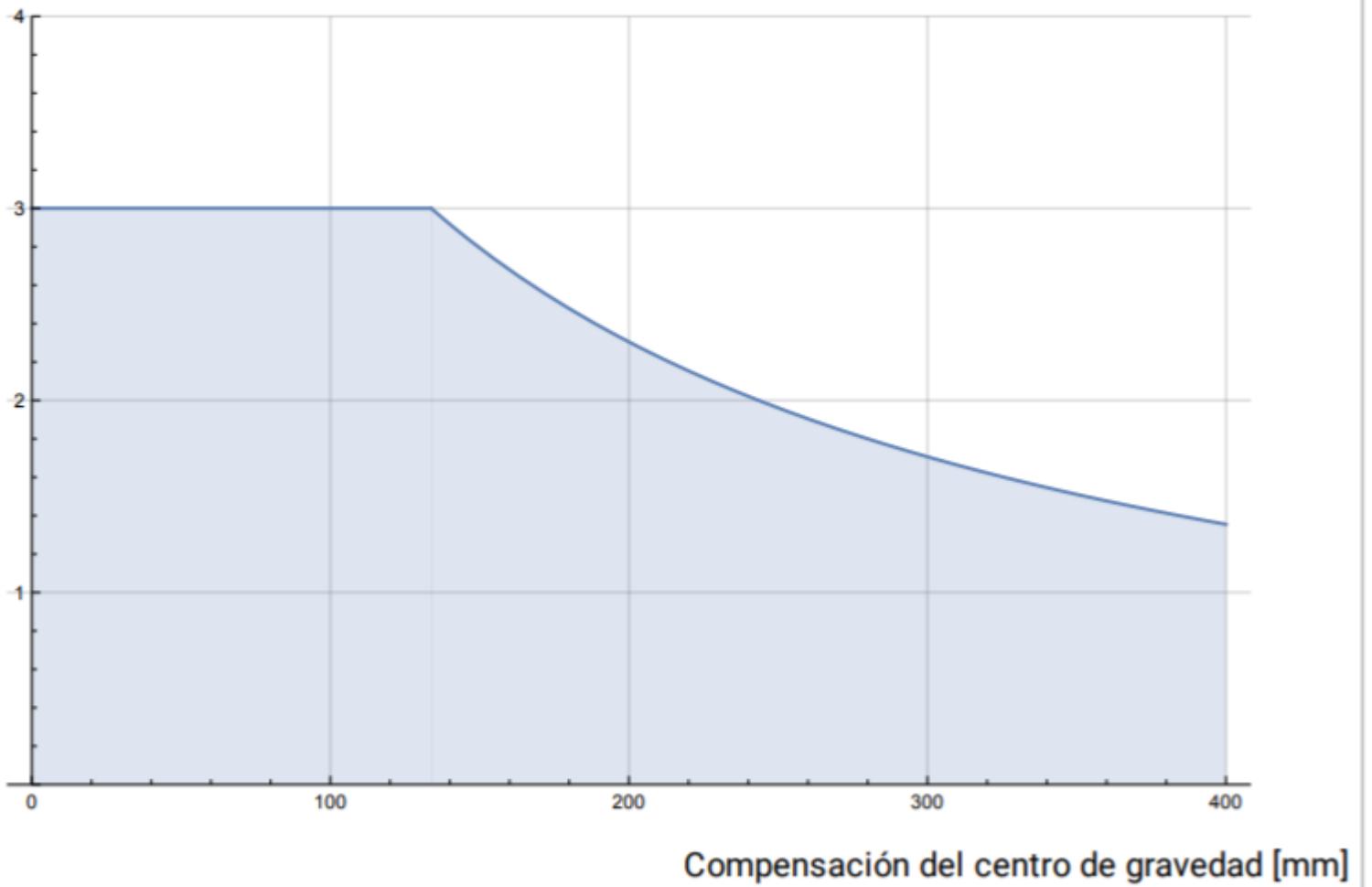
De acuerdo con: ISO13849-1:2008, Pld y en ISO 10218-1:2011, cláusula 5.10.5. Las normas de seguridad

asociadas al robot son: EC 60204-1, IEC60947-5-1 (direct acting contacts), ISO 13850 and ISO 13849-1.

Certificado por TÜV NORD SS1 (Safe Stop 1)according to IEC 615800-5-2, IEC61784-3, 61800-5-2

i) Haga un análisis del diagrama de la capacidad de carga.

Payload [kg]



La capacidad de carga del robot está ligada a la compensación del centro de gravedad, por tanto a mayor distancia este separada la carga de la base del robot, menor será el peso que puede soportar este.

2. Según las características anteriores explique para qué aplicaciones se usa este robot.

Debido a su pequeño tamaño, está dirigido a aplicaciones que no requieren tanta carga mecánica, aplicaciones como: soldadura, en aplicaciones de pegado, aplicaciones de atornillamiento, de pintura, de pick and place, de operación de herramientas de mano, trabajo en laboratorios, y aplicaciones donde existen gases tóxicos.

Parte II | Modelos

Iniciando

Haciendo uso de MATLAB® y los toolboxes construya un modelo del robot o utilice uno disponible en los toolboxes y que corresponda al robot asignado.

1. Realice el análisis geométrico del robot asignado a través de DH modificado y utilice las funciones de RVC para construir un modelo en alambres.

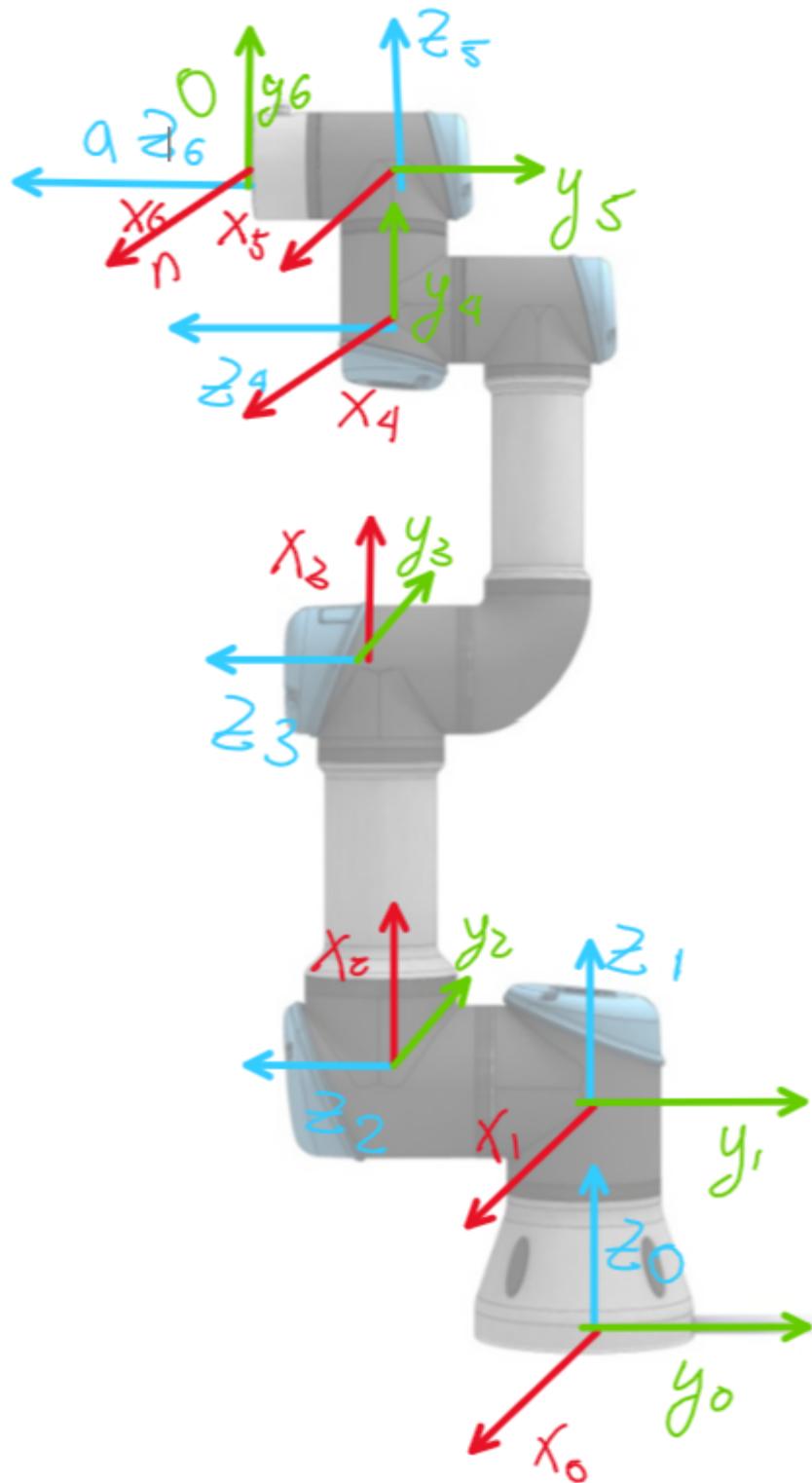


Tabla de parametros Denavit-Hartenberg

i	α_{i-1}	a_{i-1}	d_i	θ_i	offset
1	0	0	152	q1	0
2	$\frac{\pi}{2}$	0	120	q2	$\frac{\pi}{2}$
3	0	244	0	q3	0
4	0	213	-10	q4	$-\frac{\pi}{2}$
5	$-\frac{\pi}{2}$	0	83	q5	0
6	$\frac{\pi}{2}$	0	82	q6	0

```
clear all;clc;

%Creación de eslabones
```

```
L(1) = Link([0 0.152 0 0], 'modified');
L(2) = Link([0 0.120 0 pi/2 0 pi/2], 'modified');
L(3) = Link([0 0 0.244 0], 'modified');
L(4) = Link([0 -0.010 0.213 0 0 -pi/2], 'modified');
L(5) = Link([0 0.083 0 -pi/2], 'modified');
L(6) = Link([0 0.082 0 pi/2], 'modified');
R = SerialLink(L, 'name', 'Universal Robot UR3')
```

R =

Universal Robot UR3 (6 axis, RRRRRR, modDH, fastRNE)

j	theta	d	a	alpha	offset
1	q1	0.152	0	0	0
2	q2	0.12	0	1.571	1.571
3	q3	0	0.244	0	0
4	q4	-0.01	0.213	0	-1.571
5	q5	0.083	0	-1.571	0
6	q6	0.082	0	1.571	0

```
grav = [0 0 0 0 0 0];
base = [1 0 0 0 0 0;
        0 1 0 0 0 0;
        9.81 0 0 1 0 0;
        0 0 0 1 0 0];
tool = [1 0 0 0 0 0;
        0 1 0 0 0 0;
        0 0 1 0 0 0;
        0 0 0 1 0 0];
```

Aquí presentamos un modelo de alambres.

```
figure(1)

hold on

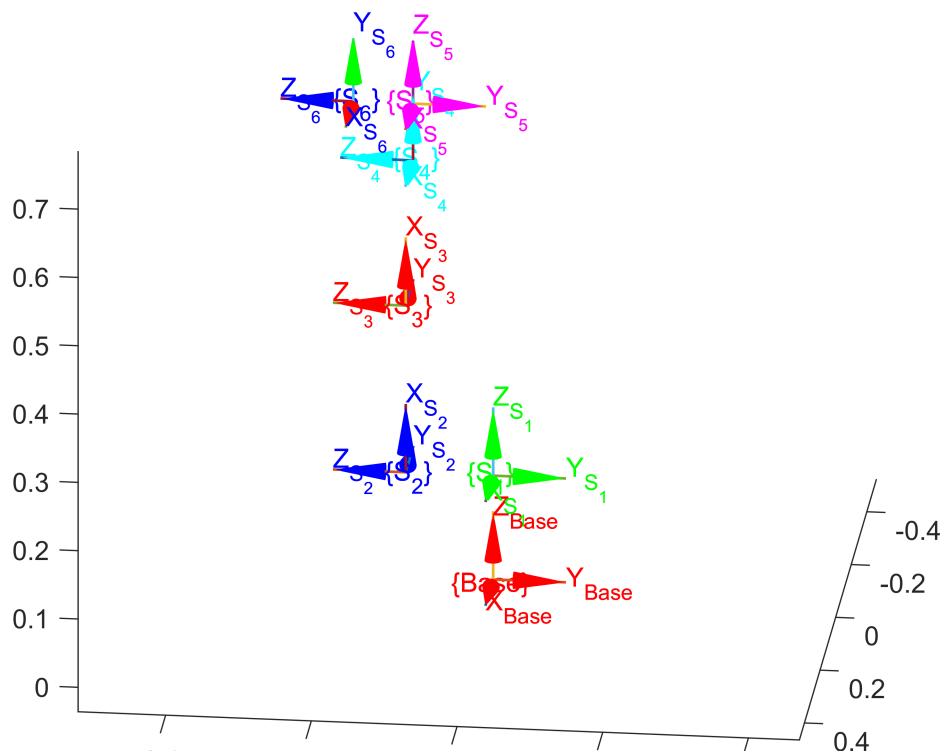
s=0.1;
trplot(R.base,'length', s,'arrow','frame','Base','color','r')
trplot(R.base*L(1).A(0), 'length', s, 'arrow', 'frame', 'S_{1}', 'color', 'g')
```

```

trplot(R.base*L(1).A(0)*L(2).A(0), 'length', s, 'arrow', 'frame', 'S_{2}', 'color', 'b')
trplot(R.base*L(1).A(0)*L(2).A(0)*L(3).A(0), 'length', s, 'arrow', 'frame', 'S_{3}', 'color', 'r')
trplot(R.base*L(1).A(0)*L(2).A(0)*L(3).A(0)*L(4).A(0), 'length', s, 'arrow', 'frame', 'S_{4}', 'color', 'c')
trplot(R.base*L(1).A(0)*L(2).A(0)*L(3).A(0)*L(4).A(0)*L(5).A(0), 'length', s, 'arrow', 'frame', 'S_{5}', 'color', 'm')
trplot(L(1).A(0)*L(2).A(0)*L(3).A(0)*L(4).A(0)*L(5).A(0)*L(6).A(0), 'length', s, 'arrow', 'frame', 'S_{6}', 'color', 'g')

xlim([-0.521 0.464])
ylim([-0.52 0.47])
zlim([-0.036 0.784])
view([96.105 21.239])
pause(1)

```

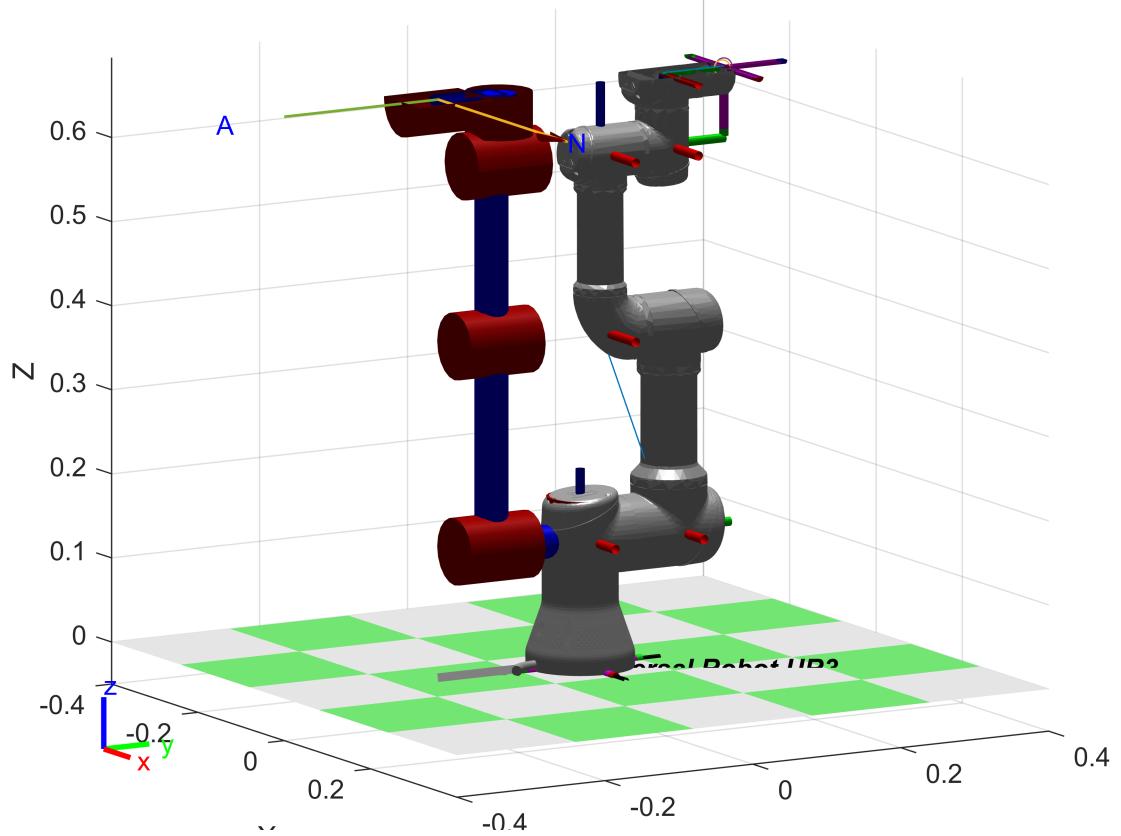
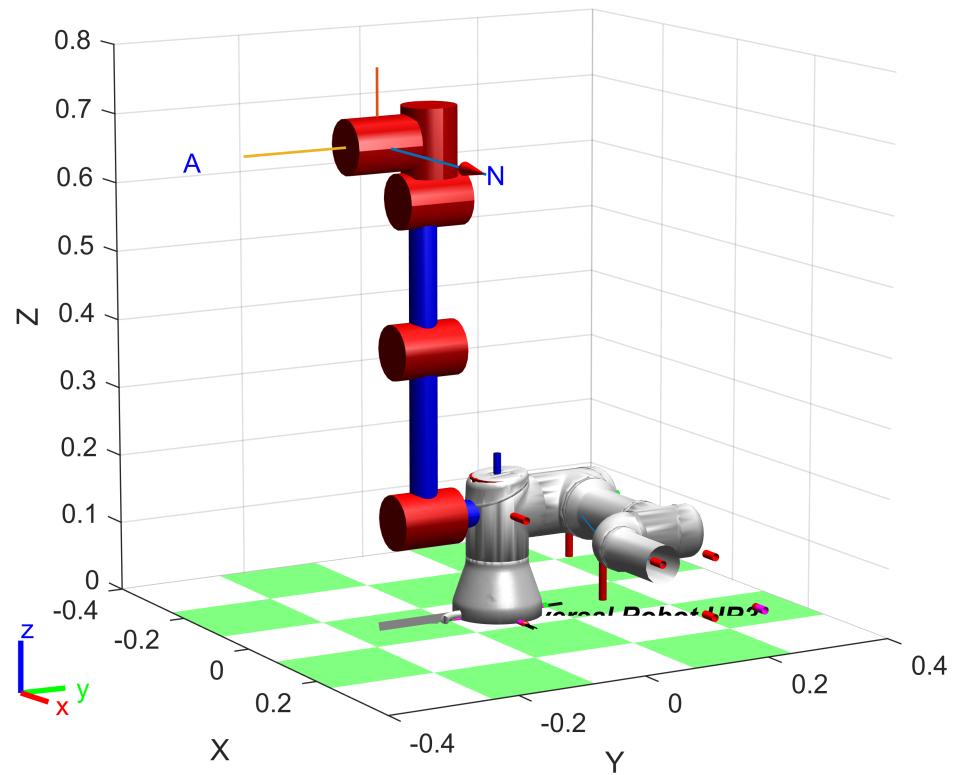


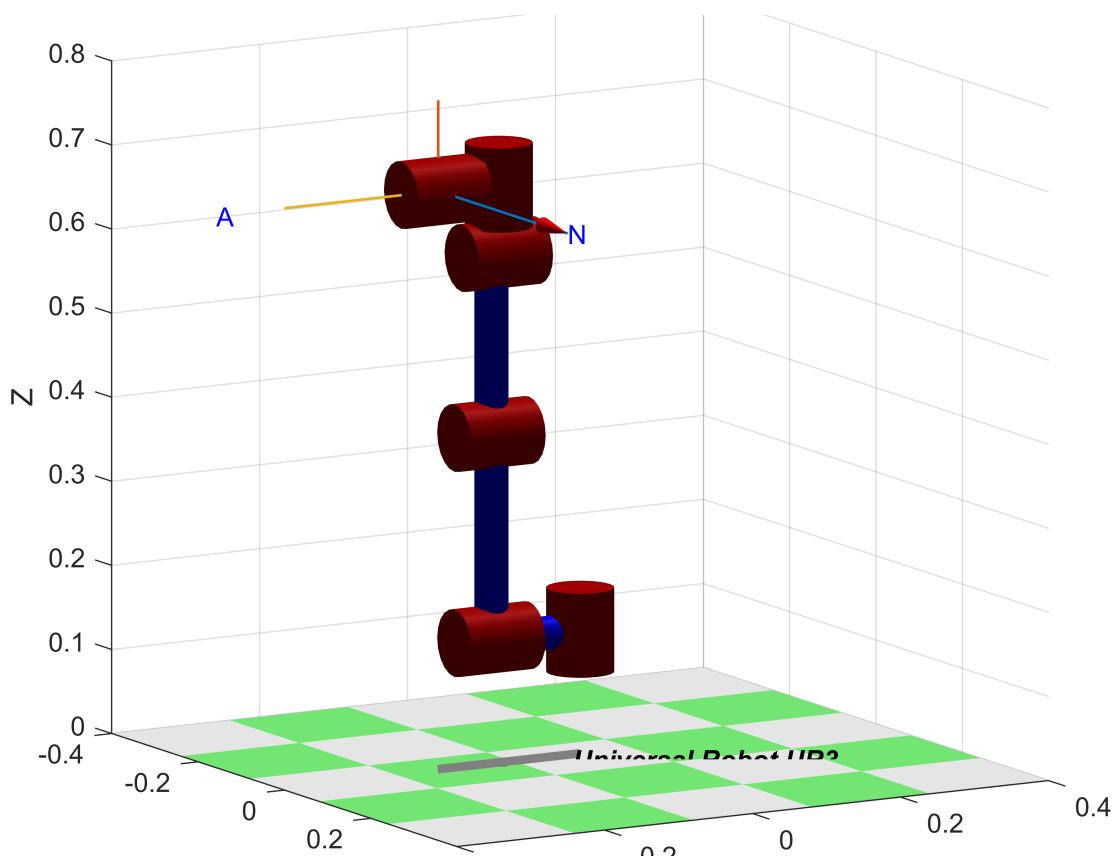
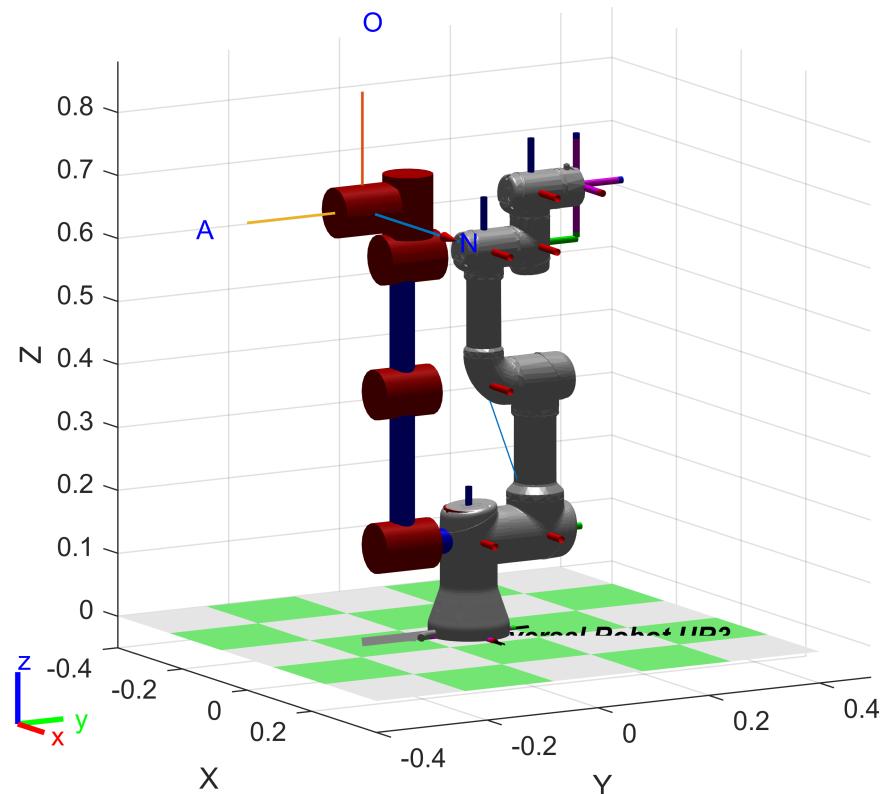
Y se observa el modelo del robot.

```

figure(2)
R.plot([0 0 0 0 0 0], 'workspace', [-0.4 0.4 -0.4 0.4 0 0.8], 'noa', 'view',[59.733 11.020])

```





```
pause(1)
```

2. Considerando el robot asignado, construya el modelo del robot utilizando RST.

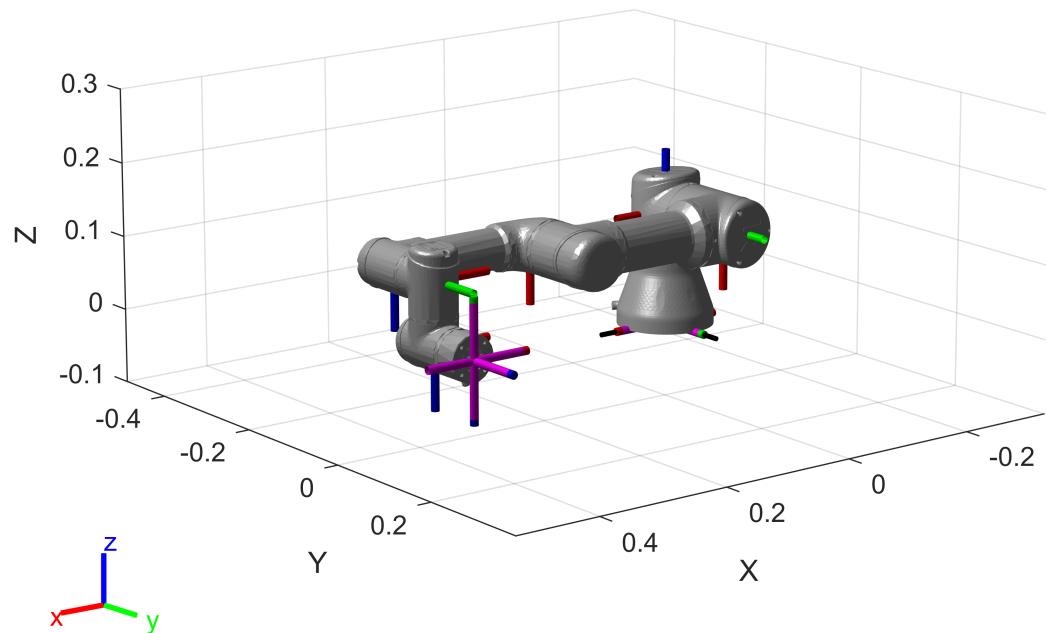
```

UR3=loadrobot("universalUR3");

show(UR3);

xlim([-0.336 0.528])
ylim([-0.491 0.373])
zlim([-0.1 0.3])
view([142.295 14.215])

```



```

pause(1)
showdetails(UR3)

```

Robot: (10 bodies)

Idx	Body Name	Joint Name	Joint Type
1	base_link	world_joint	fixed
2	base	base_link-base_fixed_joint	fixed
3	shoulder_link	shoulder_pan_joint	revolute
4	upper_arm_link	shoulder_lift_joint	revolute
5	forearm_link	elbow_joint	revolute
6	wrist_1_link	wrist_1_joint	revolute
7	wrist_2_link	wrist_2_joint	revolute
8	wrist_3_link	wrist_3_joint	revolute
9	ee_link	ee_fixed_joint	fixed
10	tool0	wrist_3_link-tool0_fixed_joint	fixed

3. Compare los dos métodos.

La metodología por el RVC (Robotics, Vision & Control) permite tener un análisis matemático más detallado, puesto que esta construido a partir de parámetros de Denavit-Hartenberg, sin embargo solo corresponde a una representación matemática, de manera que se limita en condiciones físicas como colisiones. El método por el RST (Robotics System Toolbox) , permite configurar el robot de manera más fiel a la real, con modelos 3D, y eslabones como está construido el robot en realidad. A pesar de considerar cada eslabón como un cuerpo rígido, si habilita opciones como masa, inercia, centro de masa y cálculo de torques en cada motor.

Una diferencia importante es que en el RST, se configura una posición de Home desde el inicio, a pesar de que ambos se construyen ubicando puntos de referencia en base al eslabón anterior, y luego se agrupan esos eslabones para tener un cuerpo rígido, un "RigidBody".

Modelo Geométrico Directo

1. Halle el modelo geométrico directo de su robot asignado usando DH modificado.

```
R
```

```
R =
```

```
Universal Robot UR3 (6 axis, RRRRRR, modDH, fastRNE)
```

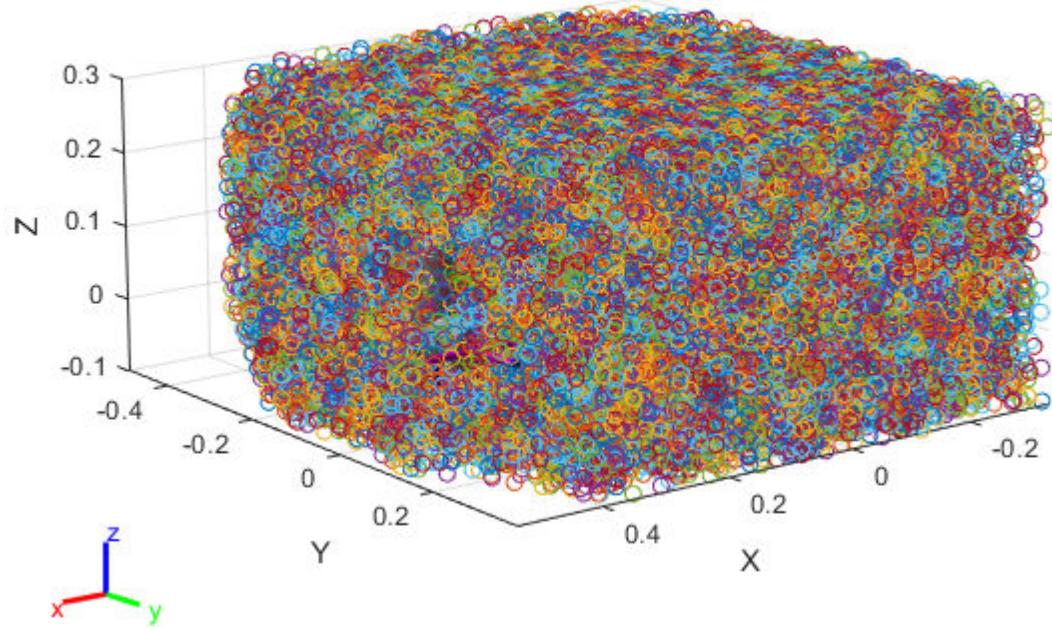
j	theta	d	a	alpha	offset
1	q1	0.152	0	0	0
2	q2	0.12	0	1.571	1.571
3	q3	0	0.244	0	0
4	q4	-0.01	0.213	0	-1.571
5	q5	0.083	0	-1.571	0
6	q6	0.082	0	1.571	0

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

2. Haciendo uso del modelo directo haga una representación del espacio de trabajo alcanzable del robot.

```
s = 2*pi;  
step=0.5;  
hold on  
for i = 0:step:s  
    for j = 0:step:s  
        for k = 0:step:s  
            for m = 0:step:s  
                for n = 0:step:s  
                    endeffect=R.base*L(1).A(i)*L(2).A(j)*L(3).A(k)*L(4).A(m)*L(5).A(n)*L(6).A(6);  
                    position=endeffect(1:3,4);
```

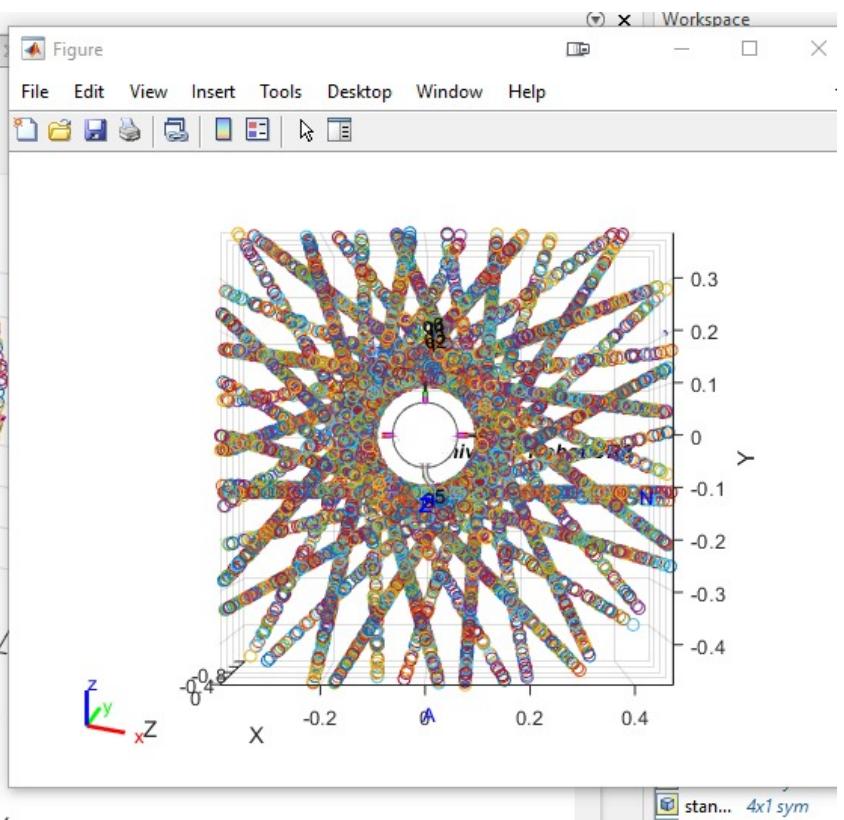
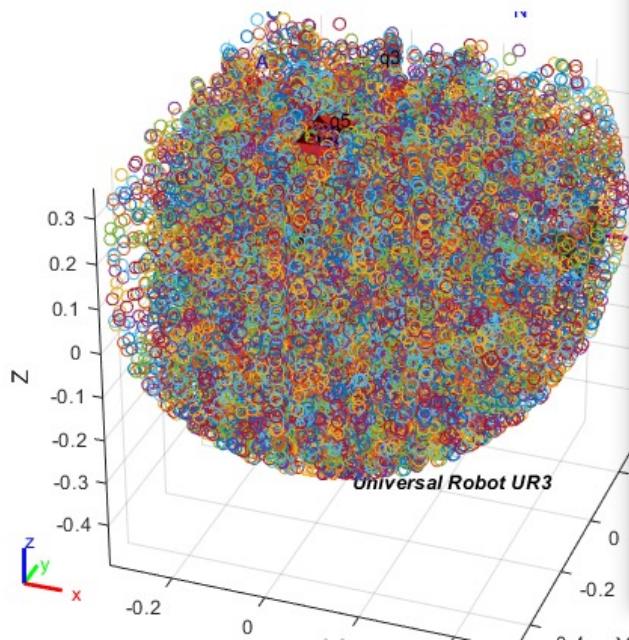
```
    plot3(position(1),position(2),position(3), 'o');
end
end
end
end
```



```

OjedaOlarte.mlx  ej_cinematica_directa_04_nov_2021.mlx  Taller2.mlx
end
end
end

```



3. Con la hoja técnica del robot, el fabricante provee puntos de calibración. Con la ayuda de la cinemática directa verifique dichos puntos.

Dado que Universal Robots indica son posiciones para calibrar el UR3, acorde con el manual se debe colocar un una posición vertical acorde con las siguientes imágenes el manual de servicio.



Base zero position is aligned so that the output flange is offset 180 degrees from the cable at the back of the robot base.

Shoulder, Elbow and Wrist 1 zero output flange vertically aligned (assuming Base is horizontal).
Make sure that base of robot is horizontal, use spirit level to align joints.

Wrist 2:

Wrist 3:

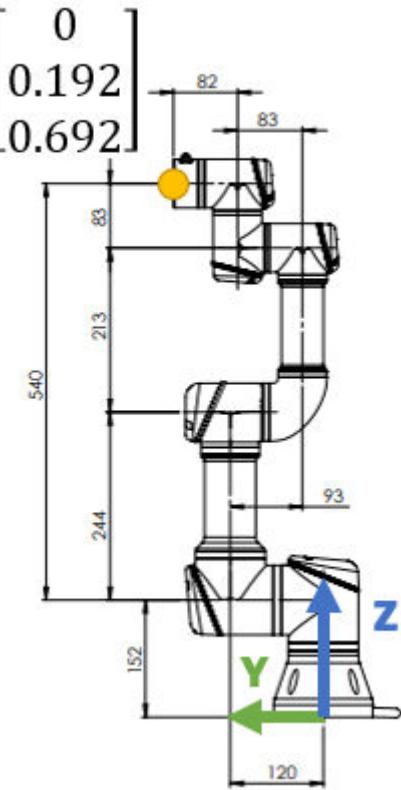


Wrist 2 zero position is aligned similar to Base joint, with tool flange parallel with wrist 1 output flange.

Wrist 3 zero position is aligned so tool connector is pointing upward.
Mount two bolts in tool holes and use spirit level to align joint.

10. Select Calibration tab and press Zero current joint position to calibrate the joint.

Con la cuál, a partir de las medidas expuestas anteriormente definimos el punto de referencia del efecto final como se ve en el siguiente diagrama.



Que acorde con el modelo

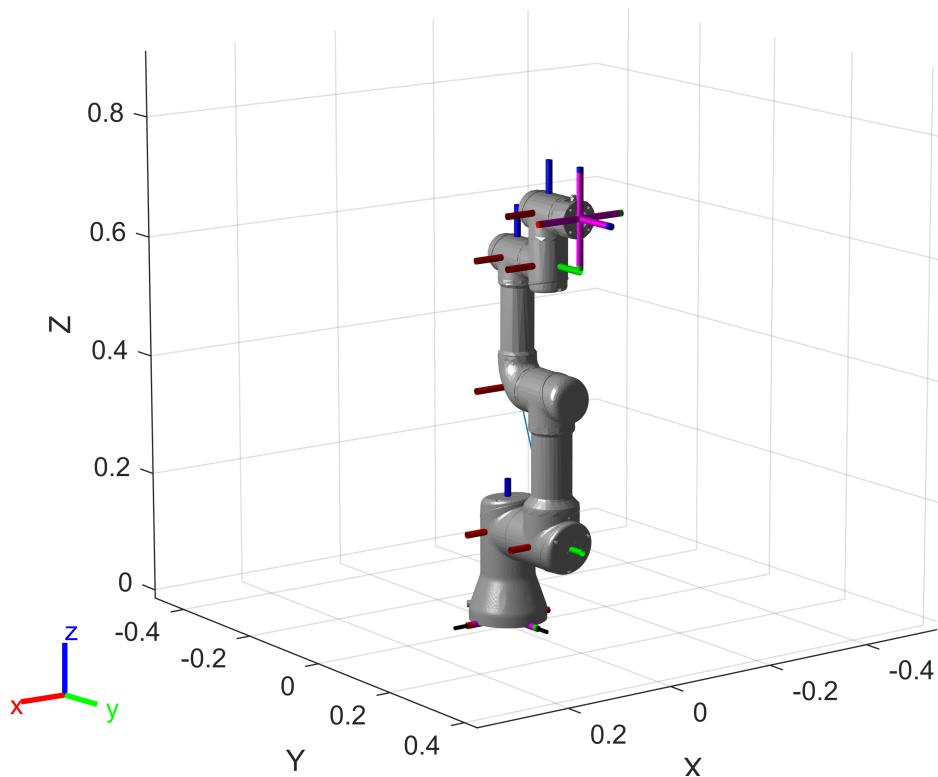
4. Haga uso de las funciones de cinemática directa de ambos toolboxes y compruebe los resultados anteriores.

Con RVC

```
figure(4)
R.plot([0 0 0 0 0 0], 'workspace', [-0.4 0.4 -0.4 0.4 0 0.8], 'noa', 'view',[59.733 11.020]);
pause(1)
```

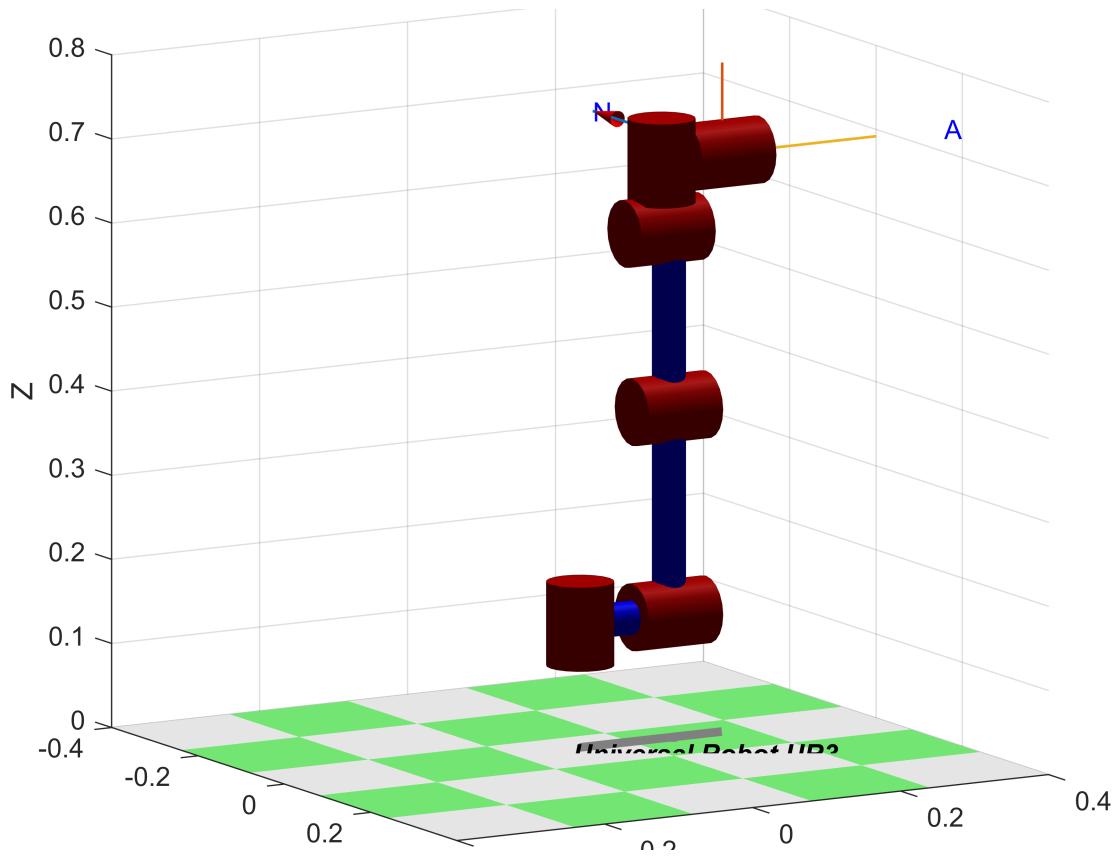
Con RST

```
load("configTest.mat");
figure(5)
show(UR3,configTest);
xlim([-0.549 0.373])
ylim([-0.489 0.433])
zlim([-0.014 0.908])
view([143.602 12.403])
```



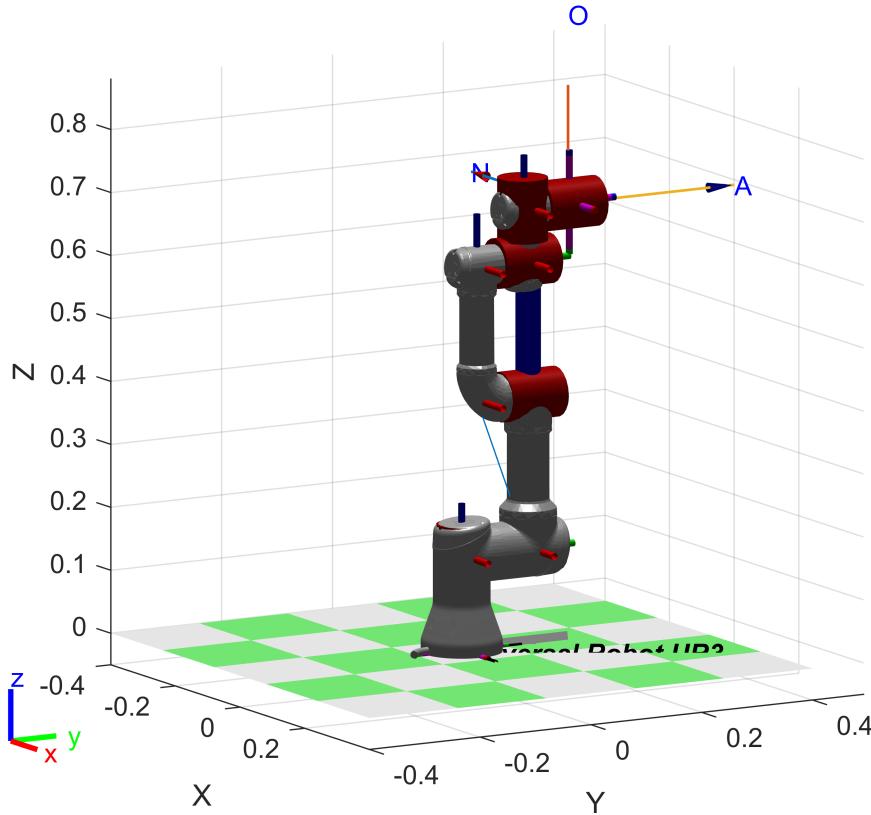
```
pause(1)
figure(6)

hold on
R.base=trotz(pi);
R.plot([0 0 0 0 0 0], 'workspace', [-0.4 0.4 -0.4 0.4 0 0.8], 'noa', 'view',[59.733 11.020]);
```



```
pause(1)
show(UR3,configTest);
hold off

xlim([-0.400 0.400])
ylim([-0.400 0.492])
zlim([-0.050 0.880])
```



```

drawnow
pause(1)
UR3HomeMTH = getTransform(UR3,configTest,'tool0','base_link');
RSTposition=UR3HomeMTH(1:3,4)

RSTposition = 3×1
-0.0000
0.1943
0.6942

RVCendeffect=R.base*L(1).A(0)*L(2).A(0)*L(3).A(0)*L(4).A(0)*L(5).A(0)*L(6).A(0);
RVCposition=RVCendeffect(1:3,4)

RVCposition = 3×1
-0.0000
0.1920
0.6920

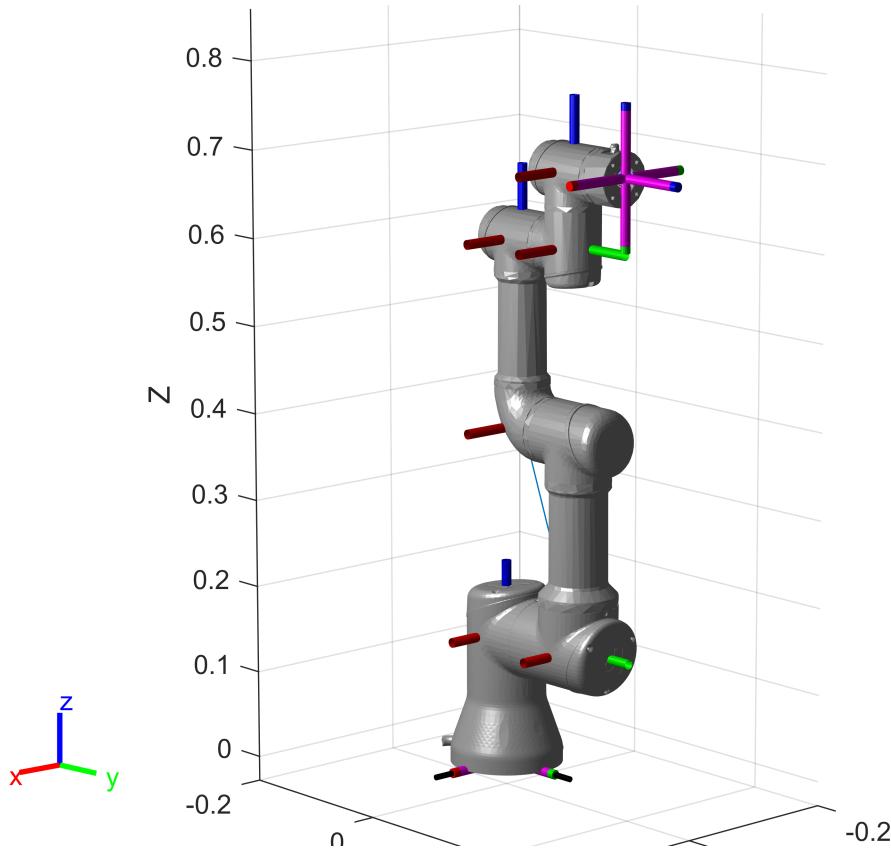
pause(1);
figure(8)
hold on
plot3(RVCposition(1),RVCposition(2),RVCposition(3),'o');
plot3(RSTposition(1),RSTposition(2),RSTposition(3),'o');
%sdR.plot([0 0 0 0 0 0],'workspace',[ -0.4 0.4 -0.4 0.4 0 1], 'noa','view',[59.733 11.020]);

show(UR3,configTest);
hold off

xlim([-0.2 0.2])
ylim([-0.2 0.3])

```

```
zlim([-0.028 0.857])
view([137.871 11.963])
```



```
pause(1)
```

5. Compare los métodos.

Se puede ver que existe una pequeña diferencia entre los dos modelos, cuya distancia total es:

```
PosError=sqrt((RVCposition(1)-RSTposition(1))^2+(RVCposition(2)-RSTposition(2))^2+(RVCposition(3)-RSTposition(3))^2)
```

```
PosError = 0.0031
```

Que corresponde al error de calibración según los modelos. La posición que se encuentra en x se puede aproximar a 0.

6. Exprese la pose del efecto en matriz de cosenos directores.

A continuación se calcula la MTH de la base a la ultima articulación.

```
T01 = L(1).A(0);
```

Warning: Camera up vector must be independent from camera direction.

```
T12 = L(2).A(0);
T23 = L(3).A(0);
T34 = L(4).A(0);
T45 = L(5).A(0);
T56 = L(6).A(0);
```

```
T06 = T01*T12*T23*T34*T45*T56
```

```
T06 = 4x4
1.0000      0      0      0.0000
0      0.0000   -1.0000   -0.1920
0      1.0000   0.0000    0.6920
0          0      0      1.0000
```

```
Cosenos = T06(1:3,1:3)
```

```
Cosenos = 3x3
1.0000      0      0
0      0.0000   -1.0000
0      1.0000   0.0000
```

7. Exprese la pose del efecto final en coordenadas generalizadas del efecto final.

Calculamos los angulos roll, pitch y yaw:

```
%xyz, rpy
rpy=tr2rpy(T06)
```

```
rpy = 1x3
1.5708      0      0
```

Finalmente sacamos el vector de traslación:

```
t= T06(1:3,4)
```

```
t = 3x1
0.0000
-0.1920
0.6920
```

8. Elija uno de los métodos anteriormente usados y desarrolle una GUI que permita mover cada articulación mediante controles tipo Slider, visualizar el robot y la posición del efecto final.

Se encuentra adjunto el link al repositorio donde se encuentra la aplicación desarrollada usando el metodo modificado y el toolbox de Peter Corke.

Consultando las distintas referencias las articulaciones tienen un rango de -180 a 180 en su rango de movimiento articular, sin embargo, el fabricante no expresa de manera explicita los puntos donde el robot tiene colisiones.

9. En el informe incluya capturas de pantalla verificando las posiciones EF y las articulaciones.

Referencias

- Corke, P. (2017). Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised (Vol. 118). Springer.
- Merat, F. (1987).
- Introduction to robotics: Mechanics and control. IEEE Journal on Robotics and Automation, 3(2), 166-166.
- Corke, P. (2011). Robotics toolbox for MATLAB, release 9 [Software]. Robotics toolbox for MATLAB.
- Universal Robots. UR3 Service Manual.
- Universal Robots. UR3 Technical specifications.
- Universal Robots. UR3/CB3