



Cinemática Inversa Parte III:

ABB IRB-140

Dr. Alejandro Gutiérrez–Giles
ivan.gutierrez@ingenieria.unam.edu

Robótica Industrial
Departamento de Control
Facultad de Ingeniería - UNAM

Google Classroom code: **gzt5pya**



ABB IRB-140

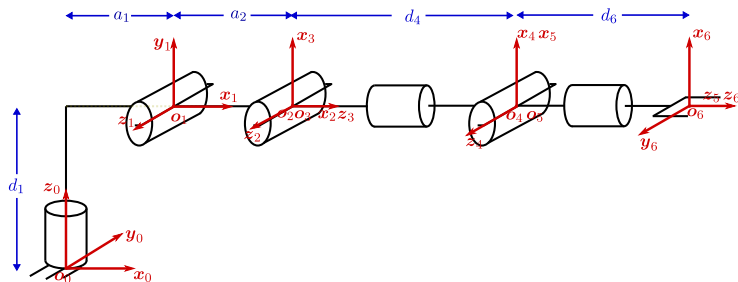
- Ahora calcularemos la cinemática inversa de un robot industrial muy conocido, el ABB IRB 140, que se muestra en la figura





ABB IRB-140

- En la clase de cinemática directa ya hicimos la asignación de Denavit Hartenberg para este robot, que se repite en la figura



- Este robot es muy parecido al robot antropomórfico con offsets de la clase anterior, sólo que en este caso $d_2 = 0$.



- También en la clase de cinemática directa se obtuvieron, a partir del manual de especificaciones del robot, los parámetros de Denavit-Hartenberg que se muestran en la tabla

i	a_i	d_i	α_i	θ_i	C.I.
1	0.070	0.3521	$\pi/2$	θ_1^*	0
2	0.360	0	0	θ_2^*	0
3	0	0	$\pi/2$	θ_3^*	$\pi/2$
4	0	0.380	$-\pi/2$	θ_4^*	0
5	0	0	$\pi/2$	θ_5^*	0
6	0	0.065	0	θ_6^*	0



Cinemática Inversa

- Una vez realizado el desacople cinemático ${}^0\mathbf{o}_c = {}^0\mathbf{o}_d - d_6 {}^0\mathbf{z}_d$ se procede a obtener la cinemática inversa de posición.
- Esta cinemática inversa es muy parecida a la del robot antropomórfico con offset de la clase anterior. Siguiendo el mismo procedimiento se puede obtener

$$\theta_1 = \text{atan2}(o_{cy}, o_{cx}) \quad (1)$$

$$\theta_3 = \alpha + \pi/2 \quad (2)$$

$$\theta_2 = \text{atan2}(o_{cz} - d_1, r - a_1) - \text{atan2}(d_4 \sin(\alpha), a_2 + d_4 \cos(\alpha)) , \quad (3)$$



Cinemática Inversa

- donde

$$\alpha = \text{atan2} \left(\pm \sqrt{1 - D^2}, D \right) \quad (4)$$

$$D = \frac{(r - a_1)^2 + (o_{cz} - d_1)^2 - a_2^2 - d_4^2}{2a_2d_4} \quad (5)$$

$$r = \sqrt{o_{cx}^2 + o_{cy}^2}. \quad (6)$$

- Recuerden que para obtener las tres últimas variables articulares le extraemos los ángulos de Euler ZYZ a la matriz ${}^3\mathbf{R}_6$ y los igualamos como $\theta_4 = \phi$, $\theta_5 = \theta$ y $\theta_6 = \psi$.



- A continuación se muestra el código en Matlab para calcular la cinemática directa (de una clase anterior)

```
function [H06] = directa_irb140(q)

% Parámetros de D-H del robot ABB IRB-140
a = [0.07, 0.36, 0, 0, 0, 0];
d = [0.3521, 0, 0, 0.38, 0, 0.065];
al = [pi/2, 0, pi/2, -pi/2, pi/2, 0];
th = [q(1), q(2), q(3), q(4), q(5), q(6)];

% Cálculo de la cinemática directa
H = eye(4,4);

for i = 1:6
    H = H*[cos(th(i)), -sin(th(i))*cos(al(i)), sin(th(i))*sin(al(i)), a(i)*cos(th(i));...
        sin(th(i)), cos(th(i))*cos(al(i)), -cos(th(i))*sin(al(i)), a(i)*sin(th(i));...
        0, sin(al(i)), cos(al(i)), d(i);...
        0, 0, 0, 1];
end

% Salida del programa y ángulos en grados para V-REP
H06 = H;
vrep = [q(1), q(2)-pi/2, q(3), q(4), q(5), q(6)]*180/pi
```



- Ahora se muestra la programación de la cinemática inversa

```
function [q] = inversa_irb140(Hd)
```

```
a = [0.070, 0.360, 0, 0, 0, 0];
```

```
d = [0.3521, 0, 0, 0.380, 0, 0.065];
```

```
al = [pi/2, 0, pi/2, -pi/2, pi/2, 0];
```

```
% Rotación y posición deseadas
```

```
Rd = Hd(1:3,1:3);
```

```
od = Hd(1:3,4);
```

```
% Desacople cinemático
```

```
oc = od - Rd(1:3,3)*d(6);
```

```
ocx = oc(1); ocy = oc(2); ocz = oc(3);
```

```
% Cinemática inversa de posición
```

```
th(1) = atan2(ocy,ocx);
```

```
r = sqrt(ocx*ocx + ocy*ocy);
```

```
D = ((r - a(1))*(r - a(1)) + (ocz - d(1))*(ocz - d(1)) - a(2)*a(2) - d(4)*d(4))/(2*a(2)*d(4));
```

```
alpha = atan2(-sqrt(1-D*D),D);
```

```
th(3) = pi/2 + alpha;
```

```
th(2) = atan2(ocz-d(1),r-a(1)) - atan2(d(4)*sin(alpha),a(2)+d(4)*cos(alpha));
```



■ continuación...

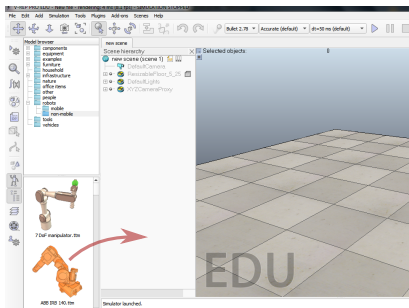
```
% Cálculo de las matrices R03 y R36
H = eye(4,4);
for i = 1:3
    H = H*[cos(th(i)), -sin(th(i))*cos(al(i)), sin(th(i))*sin(al(i)), a(i)*cos(th(i));...
          sin(th(i)), cos(th(i))*cos(al(i)), -cos(th(i))*sin(al(i)), a(i)*sin(th(i));...
          0, sin(al(i)), cos(al(i)), d(i);...
          0, 0, 0, 1];
end
R03 = H(1:3,1:3);
R36 = R03'*Rd;

% Extracción de los ángulos de Euler
anEu = ang_Euler(R36);
th(4) = anEu(1); th(5) = anEu(2); th(6) = anEu(3);

% Salida del programa y ángulos en grados para V-REP
q = [th(1), th(2), th(3), th(4), th(5), th(6)];
q_vrep = [th(1), th(2)-pi/2, th(3), th(4), th(5), th(6)]*180/pi
```

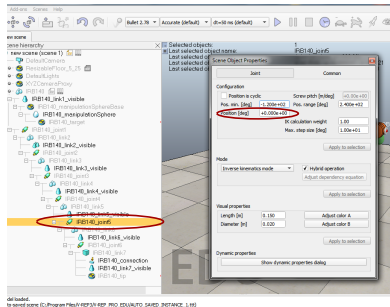


- Por último, con fines únicamente ilustrativos, se muestra un ejemplo de visualización de la cinemática inversa en el entorno virtual V-REP¹
- El primer paso es añadir el robot a la escena (arrastrar como se muestra en la figura)

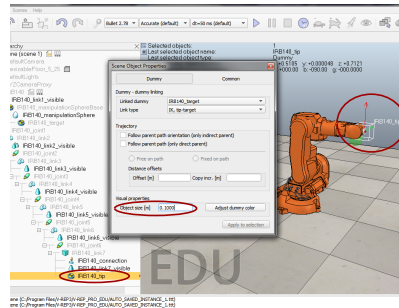


¹Pueden descargar la versión estudiantil gratuita V-REP PRO EDU en www.coppeliarobotics.com

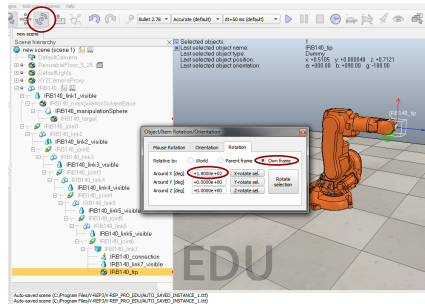
- Giremos la articulación 5 para que $\theta_5 = 0^\circ$.



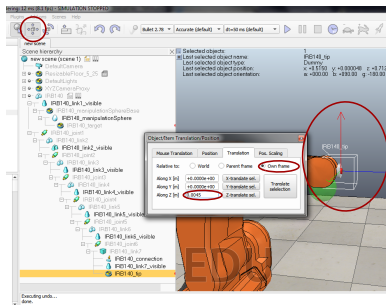
- A continuación seleccionamos el objeto que representará el sistema coordinado del efector final, el *IRB140_tip* y le cambiamos el tamaño del objeto a 0.1[m] para mejor visualización.



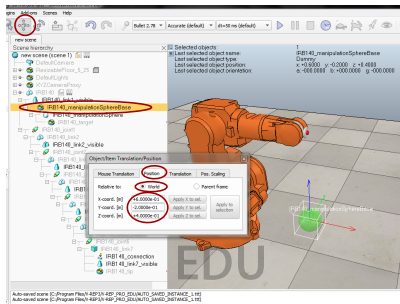
- Ahora giremos este mismo objeto 180° con respecto a su propio eje x para que coincida con nuestra asignación de DH.



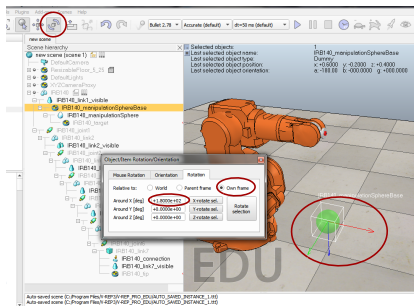
- También trasladamos este objeto 0.0045[m] sobre su propio eje z para que coincida con la punta del robot.



- Ahora vamos a definir una posición y orientación deseadas para nuestro efector final, para lo que seleccionamos el objeto *IRB140_manipulationSphereBase* (también podemos cambiar el tamaño del objeto a 0.1[m] para mejor visualización).
- Le damos la posición deseada con respecto al *World frame*, por ejemplo (0.6, -0.2, 0.4).



- Si deseamos, por ejemplo, que el robot apunte hacia abajo, entonces tenemos que hacer que la flecha azul (eje z) de este objeto apunte hacia abajo (recuerde que los ejes x, y, z corresponden a los colores rojo, verde y azul respectivamente).
- Podemos hacer esto si rotamos este objeto 180° con respecto a su propio eje x .



- Nuestro objetivo es llevar al efector final a la posición y orientación deseadas, descritas por el objeto *IRB140_manipulationSphereBase*. La transformación homogénea correspondiente es

$${}^0\mathbf{H}_d = \begin{bmatrix} 1 & 0 & 0 & 0.6 \\ 0 & -1 & 0 & -0.2 \\ 0 & 0 & -1 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$



- Esta matriz la podemos meter a la función de Matlab como se muestra en la figura (recuerde que V-REP maneja grados y que la articulación 2 tiene un offset de -90°)

```
Command Window
>> Hd = [1,0,0,0.6; 0,-1,0,-0.2; 0,0,-1,0.4; 0,0,0,1]

Hd =

    1.0000    0    0    0.6000
    0   -1.0000    0   -0.2000
    0    0   -1.0000    0.4000
    0    0    0    1.0000

>> inversa_irb140(Hd)

q_vrep =

-18.4349 -38.1971 11.6188 -180.0000 63.4217 161.5651

ans =

-0.3218  0.9041  0.2028 -3.1416  1.1069  2.8198

fx >> |
```



- [illegible]