Física II - Grado en Robótica (EPSE-USC Lugo)

Mecánica de Robots: Cinemática Directa

Ángel Piñeiro

E-mail: Angel.Pineiro@usc.es
Depto de Física Aplicada
Universidade de Santiago de Compostela

25 de marzo de 2025

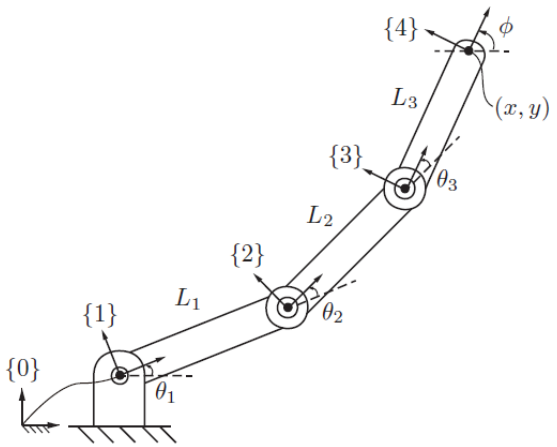
Resumen

Introducción

La resolución de la **Cinemática Directa** de un Robot $\Rightarrow$ cálculo de la **posición y orientación del elemento terminal** a partir de las coordenadas de las articulaciones (θ).

Para un Robot plano 3R cuyos eslabones tiene longitudes L_1, L_2 y L_3, escogiendo el SR fijo en la base, las posiciones y orientaciones del elemento terminal en función de θ vienen dadas por las siguientes ecuaciones:

$$\begin{aligned} x &= L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) + L_3 \cos(\theta_1+\theta_2+\theta_3) \\ y &= L_1 \operatorname{sen}\theta_1 + L_2 \operatorname{sen}(\theta_1+\theta_2) + L_3 \operatorname{sen}(\theta_1+\theta_2+\theta_3) \\ \phi &= \theta_1+\theta_2+\theta_3 \end{aligned}$$



Este sistema es fácil de resolver pero en general el análisis puede ser mucho más complicado.

Introducción

Un método más sistemático de resolver el problema consiste en definir un SR en cada articulación y de esa manera la cinemática directa se resuelve simplemente multiplicando las matrices: $T_{04} = T_{01}\,T_{12}\,T_{23}\,T_{34}$.

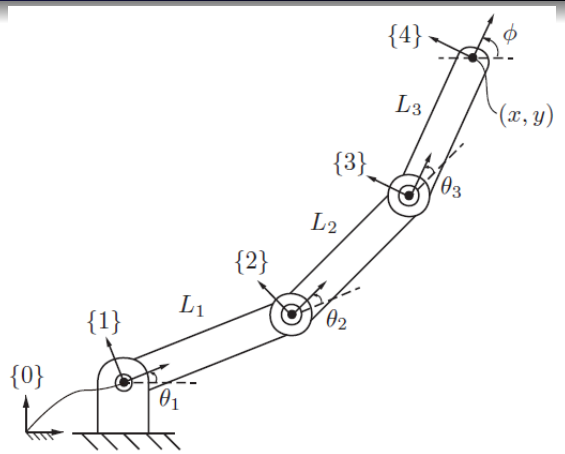
$$T_{01} = \begin{bmatrix} \cos\theta_1 & -\operatorname{sen}\theta_1 & 0 & 0 \\ \operatorname{sen}\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_{12} = \begin{bmatrix} \cos\theta_2 & -\operatorname{sen}\theta_2 & 0 & L_1 \\ \operatorname{sen}\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_{23} = \begin{bmatrix} \cos\theta_3 & -\operatorname{sen}\theta_3 & 0 & L_2 \\ \operatorname{sen}\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_{34} = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T_{34} es una matriz constante y el resto de las matrices sólo depende de una variable (θ_i)

Fórmula del producto de Matrices Exponenciales

Otra alternativa: podemos definir la matriz de transformación correspondiente a la posición y orientación del elemento terminal cuando las coordenadas de todas las articulaciones son nulas (posición "cero" del Robot)$\Rightarrow$:



$$M = \begin{bmatrix} 1 & 0 & 0 & L_1 + L_2 + L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ahora consideramos cada articulación como un eje helicoidal con paso (pitch) nulo. Si θ_1 y θ_2 son nulas, el eje helicoidal correspondiente a la rotación entorno a la tercera articulación expresada en el SR fijo $\{0\}$ es:

$$\mathcal{S}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -(L_1 + L_2) \\ 0 \end{bmatrix} \Rightarrow [\mathcal{S}_3] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -(L_1 + L_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow T_{04} = e^{[\mathcal{S}_3]\theta_3} M$$

Para calcular esto imaginamos el brazo en su configuración cero con el SR $\{0\}$ rotando entorno a la tercera articulación con $\omega_3 = 1\, rad/s$ ($\Rightarrow \omega_3$ sólo tiene componente $z = 1$ en las mismas unidades) y v_3 es la velocidad lineal de esta rotación en el punto $\{0\}$, $\Rightarrow v_3 = -\omega_3 \times q_3$, donde q_3 es un punto del eje de la tercera articulación expresado en $\{0\}$, p. ej. $q_3 = (L_1 + L_2, 0, 0)$.

Fórmula del producto de Matrices Exponenciales

La anterior expresión sólo es válida cuando $\theta_1 = \theta_2 = 0$. Si repetimos el proceso para la segunda articulación, asumiendo que sólo $\theta_1 = 0 \Rightarrow$

$$\mathcal{S}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -L_1 \\ 0 \end{bmatrix} \Rightarrow [\mathcal{S}_2] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -L_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow T_{04} = e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} M$$

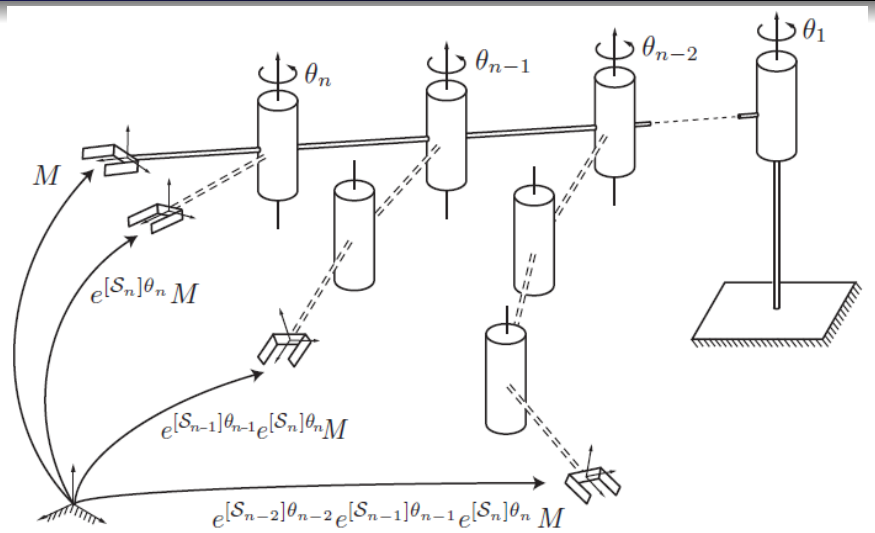
y si repetimos el proceso sin ninguna restricción:

$$\mathcal{S}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow [\mathcal{S}_1] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow T_{04} = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} M$$

$\Rightarrow$ la **cinemática directa** puede expresarse como un **producto de matrices exponenciales**, cada una de ellas correspondiente a un movimiento helicoidal (sólo se necesita un SR fijo $\{0\}$ en la base y la matriz $M \rightarrow$ configuración cero del Robot).
Denavit-Hartenberg: Metodología alternativa que utiliza un número mínimo de parámetros para describir el movimiento del robot (dof) pero necesita definir un SR en cada articulación (está basada en la ecuación $T_{04} = T_{01} T_{12} T_{23} T_{34}$).
La fórmula del producto de matrices exponenciales, utiliza $6n$ números para describir un robot de n articulaciones.

Fórmula del producto de Matrices Exponenciales

La fórmula del producto de matrices exponenciales para la resolución de la cinemática directa de un robot está basada en la aplicación de un movimiento helicoidal en cada articulación.
En la figura se ilustra la aplicación de la fórmula:



- Se escoge un SR fijo en la base $\{s\}$ y otro SR en el elemento terminal.
- Se sitúa el robot en su posición cero ($\theta = 0$), especificando la dirección del desplazamiento positivo.
- Generamos la matriz de transformación $M \in SE(3)$ de la posición cero con respecto al SR fijo.
- Se supone que la articulación n se mueve a $\theta_n \Rightarrow$ el SR del elemento terminal experimenta una transformación $T = e^{[\mathcal{S}_n]\theta_n} M$ donde $T \in SE(3)$ es la nueva configuración del elemento terminal y $\mathcal{S}_n = (\omega_n, v_n)$ es el eje helicoidal de la articulación n expresada en el SR fijo $\{s\}$.
- Asumimos que la articulación $n-1$ también varía $\Rightarrow$ aplicamos un nuevo movimiento helicoidal. Repitiendo este proceso sobre todas las articulaciones

$$\Rightarrow \boxed{T(\theta) = e^{[\mathcal{S}_1]\theta_1} \dots e^{[\mathcal{S}_{n-1}]\theta_{n-1}} e^{[\mathcal{S}_n]\theta_n} M}$$

Fórmula del producto de Matrices Exponenciales

Para el cálculo de la matriz de transformación homogénea del elemento terminal $T(\theta) \in SO(3)$ utilizando la fórmula del producto de matrices exponenciales, $\mathcal{S}_n = (\omega_n, v_n)$ es el eje helicoidal de la articulación n en coordenadas del SR fijo $\{s\}$.

- Si n es una articulación de revolución ($\Rightarrow$ movimiento helicoidal con *paso* nulo) $\Rightarrow \omega_n \in \mathbb{R}^3$ es un vector unitario en la dirección positiva de la articulación n; $v_n = -\omega_n \times q_n$; q_n es cualquier punto arbitrario sobre el eje de la articulación n escrito en coordenadas del SR fijo y θ_n es es el ángulo de la articulación.
- Si la articulación n es prismática $\Rightarrow \omega_n = 0$; $v_n \in \mathbb{R}^3$ es un vector unitario en la dirección de traslación positiva; y θ_n representa el valor de la extensión de la articulación.

Lo único que necesitamos para resolver la cinemática directa utilizando la fórmula del producto de Matrices Exponenciales es la configuración del elemento terminal cuando el robot está en la posición cero, los ejes helicoidales $\mathcal{S}_i$ de cada articulación desde la posición cero expresados en el SR fijo, y los valores de las coordenadas de las articulaciones θ_i.

Resolución de un Robot 2R

```
import numpy as np
from optparse import OptionParser
import matplotlib.pyplot as plt

def robot_structure():
    # Estructura del robot
    axis1 = np.array([0, 0, 1]) # Eje de rotación de la primera articulación
    axis2 = np.array([0, 0, 1]) # Eje de rotación de la segunda articulación
    q1 = np.array([0, 0, 0]) # Posición de la primera articulación
    q2 = np.array([0, 1, 0]) # Posición de la segunda articulación
    # Ejes helicoidales
    S1 = np.hstack((axis1, np.cross(q1, axis1))) # Eje helicoidal de la primera articulación
    S2 = np.hstack((axis2, np.cross(q2, axis2))) # Eje helicoidal de la segunda articulación
    # Matriz de transformación homogénea en la posición cero del robot
    p = np.array([0, 0, 2])
    M0 = T_matrix(np.eye(3), p)
    return S1, S2, M0

def antisymR3(v): # Calcula la matriz antisimétrica de un vector R3
    return np.array([[0, -v[2], v[1]], [v[2], 0, -v[0]], [-v[1], v[0], 0]])

def R_matrix(axis, angle):
    # Devuelve la matriz de rotación para un eje y un ángulo dados usando la fórmula de Rodrigues
    # El ángulo ha de estar en radianes
    axis = axis/np.linalg.norm(axis)
    amatrix = antisymR3(axis)
    return np.eye(3) + np.sin(angle)*amatrix + (1-np.cos(angle))*np.dot(amatrix, amatrix)

def T_matrix(R, p):
    # Devuelve la matriz de transformación homogénea a partir de una matriz de rotación y un vector de traslación
    return np.vstack((np.hstack((R, p.reshape(3,1))), np.array([0, 0, 0, 1])))
```

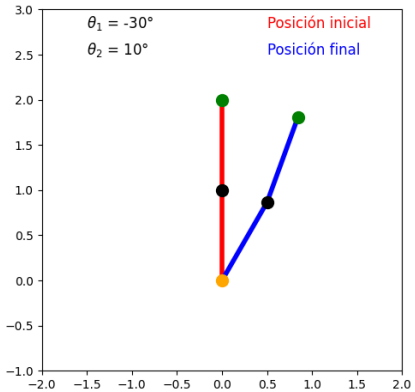
Resolución de un Robot 2R

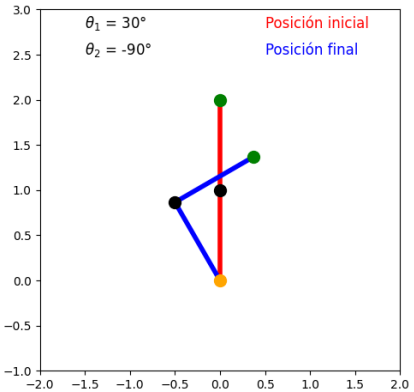
```
# Calcula la matriz exponencial a partir de un eje helicoidal y de un ángulo de rotación
def exp_matrix(S, theta):
    w = S[0:3] # Eje de rotación del eje helicoidal
    v = S[3:6] # Velocidad lineal del eje helicoidal
    R = R_matrix(w, theta) # Matriz de rotación
    wmatrix = antisymR3(w)
    # Vector de traslación
    p = (np.eye(3)*theta + (1-np.cos(theta))*wmatrix + (theta-np.sin(theta))*np.dot(wmatrix, wmatrix)) @ v
    return T_matrix(R, p)

parser = OptionParser()
parser.add_option("-a", "--a1", type="float", default=-30, help="Ángulo de rotación para la primera articulación (grados)")
parser.add_option("-b", "--a2", type="float", default=10, help="Ángulo de rotación para la segunda articulación (grados)")
(options, args) = parser.parse_args()

a1 = np.deg2rad(options.a1) # Ángulo de rotación de la primera articulación
a2 = np.deg2rad(options.a2) # Ángulo de rotación de la segunda articulación

S1, S2, M0 = robot_structure()
T1 = exp_matrix(S1, a1) # Matriz de transformación homogénea para la primera articulación
T2 = exp_matrix(S2, a2) # Matriz de transformación homogénea para la segunda articulación
M = T1 @ T2 @ M0 # Matriz de transformación homogénea para la posición final del robot

# Dado que es un movimiento en 2D, representamos la posición final del robot en el plano XY
fig = plt.figure()
ax = fig.add_subplot(111)
# Definimos el origen y el final de cada eslabón en la posición cero y en la posición final del robot
p10ini = np.array([0, 0, 0, 1])
p11ini = np.array([0, 1, 0, 1])
p20ini = p11ini
p21ini = np.array([0, 2, 0, 1])
p10fin = p10ini
p11fin = T1 @ p11ini
p20fin = p11fin
p21fin = T1 @ T2 @ p21ini
```

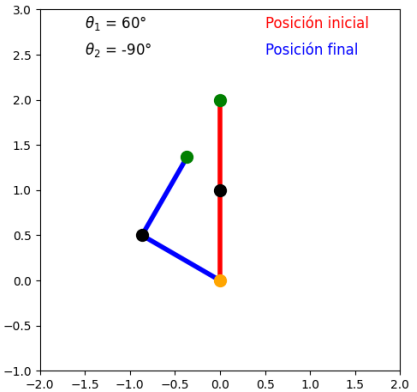
Resolución de un Robot 2R

```
# Dibuja el primer eslabón
ax.plot([p10ini[0], p11ini[0]], [p10ini[1], p11ini[1]], 'r', linewidth=4)
ax.plot([p10fin[0], p11fin[0]], [p10fin[1], p11fin[1]], 'b', linewidth=4)
# Dibuja el segundo eslabón
ax.plot([p20ini[0], p21ini[0]], [p20ini[1], p21ini[1]], 'r', linewidth=4)
ax.plot([p20fin[0], p21fin[0]], [p20fin[1], p21fin[1]], 'b', linewidth=4)
# Dibuja los puntos de interés
ax.plot(p10ini[0], p10ini[1], 'o', color='orange', markersize=10) # punto inicial de L1
ax.plot(p11ini[0], p11ini[1], 'ko', markersize=10) # punto final de L1 en la posición inicial
ax.plot(p11fin[0], p11fin[1], 'ko', markersize=10) # punto final de L1 en la posición final
ax.plot(p21ini[0], p21ini[1], 'go', markersize=10) # punto final de L2 en la posición inicial
ax.plot(p21fin[0], p21fin[1], 'go', markersize=10) # punto final de L2 en la posición final

# Ajusta el tamaño del gráfico y lo guarda
ax.set_xlim(-2, 2)
ax.set_ylim(-1, 3)
ax.set_aspect('equal', 'box')
plt.tight_layout()
plt.savefig('2R2D.png')
```

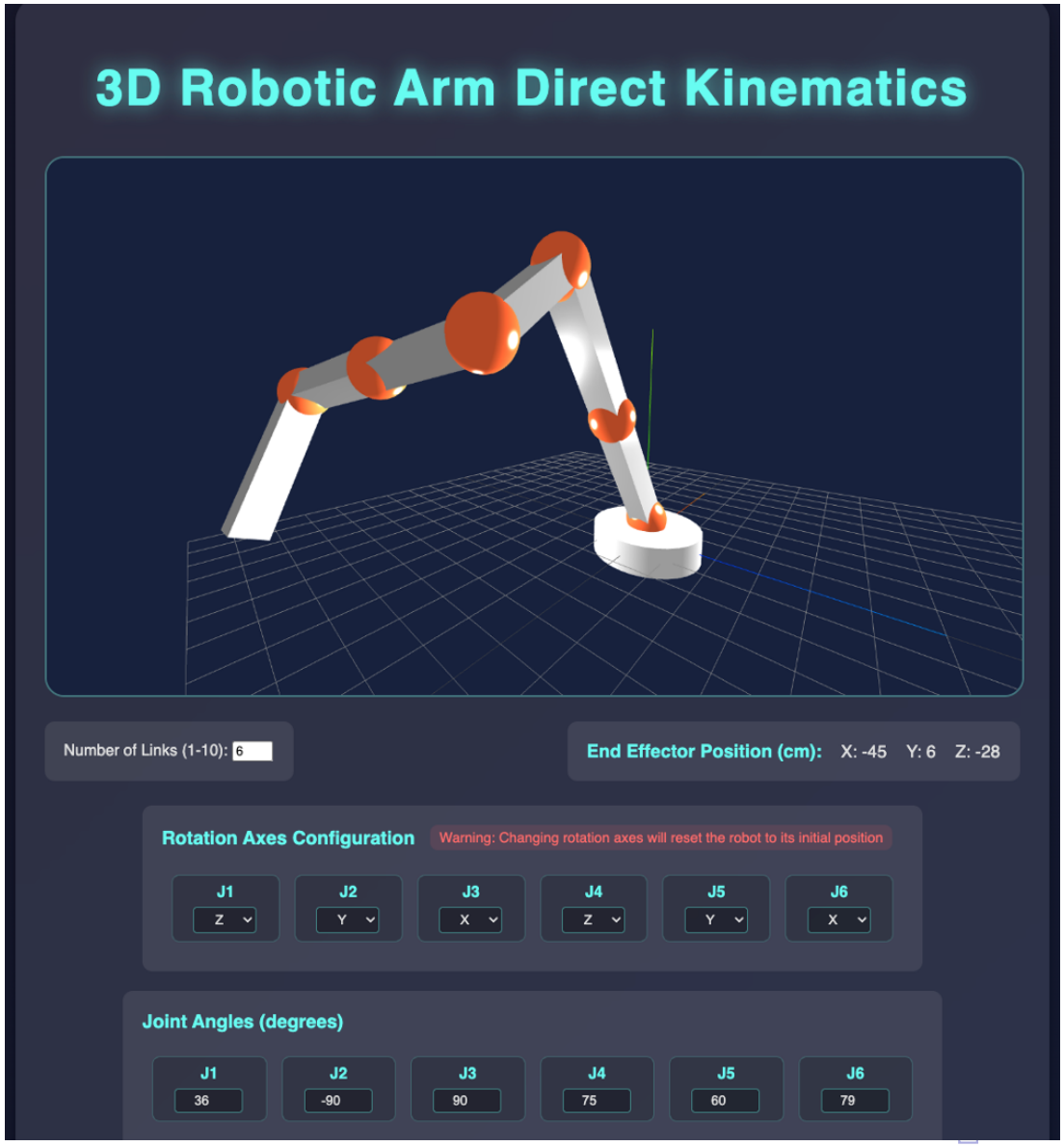
Ejemplo de ejecución: python3 2R2D.py –a1 10 –a2 10 (para ángulos de 10 grados en ambas articulaciones)



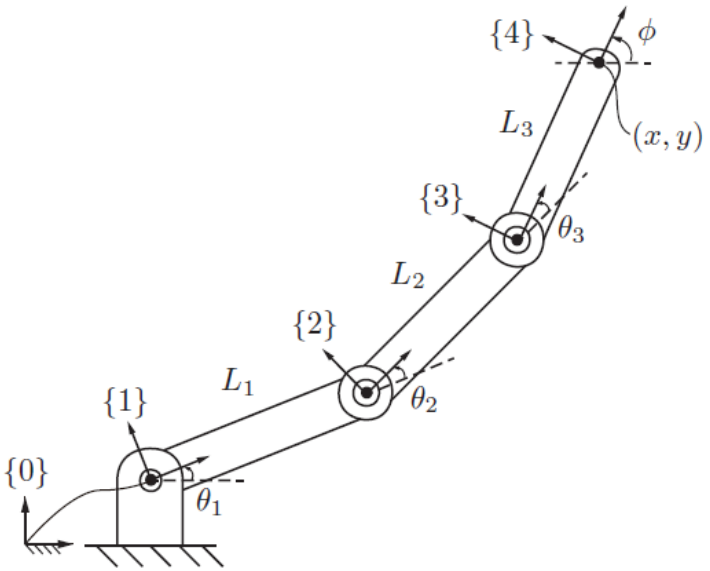




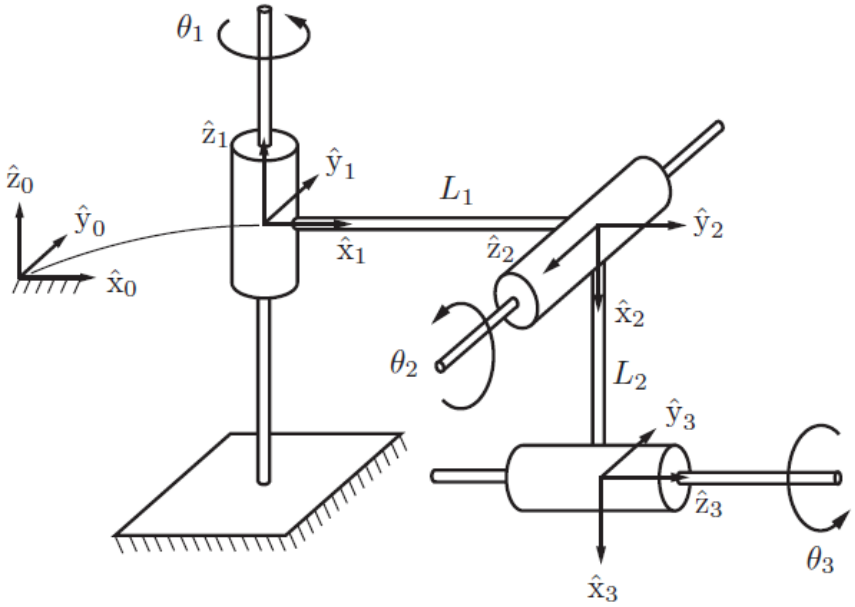
3D Robotic Arm https://angelpineiro.github.io/3D_DirectKinematics/3D_DirectKinematics.html



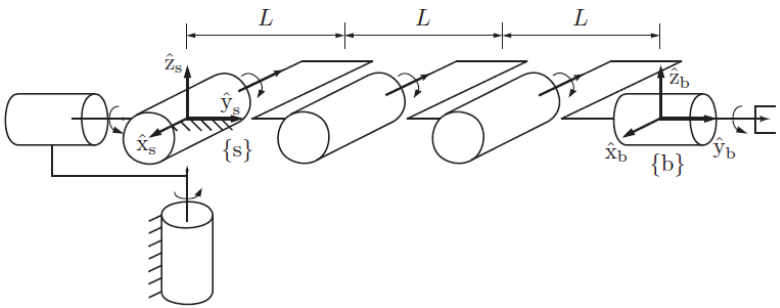
Resuelve $T(\theta)$ para los siguientes Robots



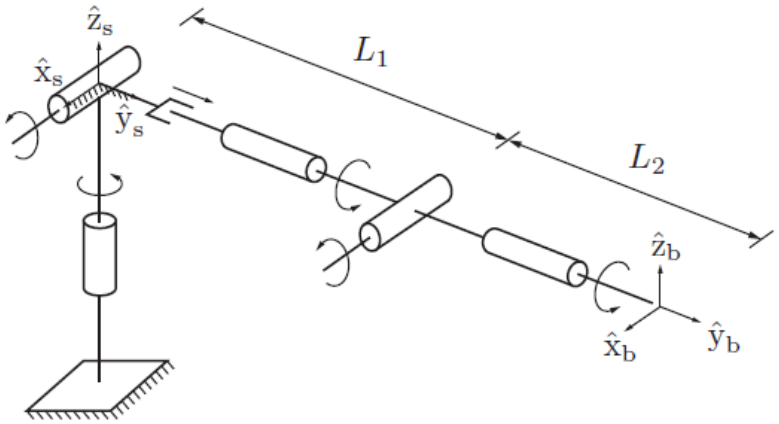
3R plano



3R 3D



6R



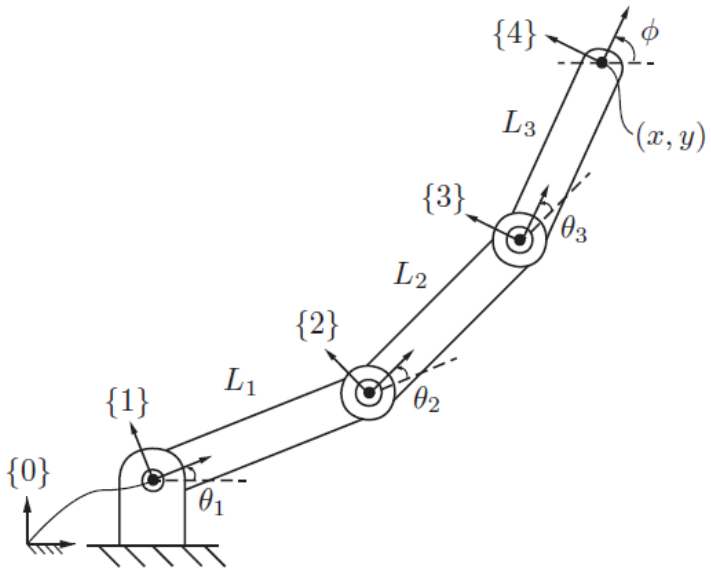
2RP3R

Robot 3R Plano

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} M$$

$$M = \begin{bmatrix} 1 & 0 & 0 & L_1 + L_2 + L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(0,0,1)$	$(L_1,0,0)$	$(0,-L_1,0)$
3	$(0,0,1)$	$(L_1+L_2,0,0)$	$(0,-(L_1+L_2),0)$



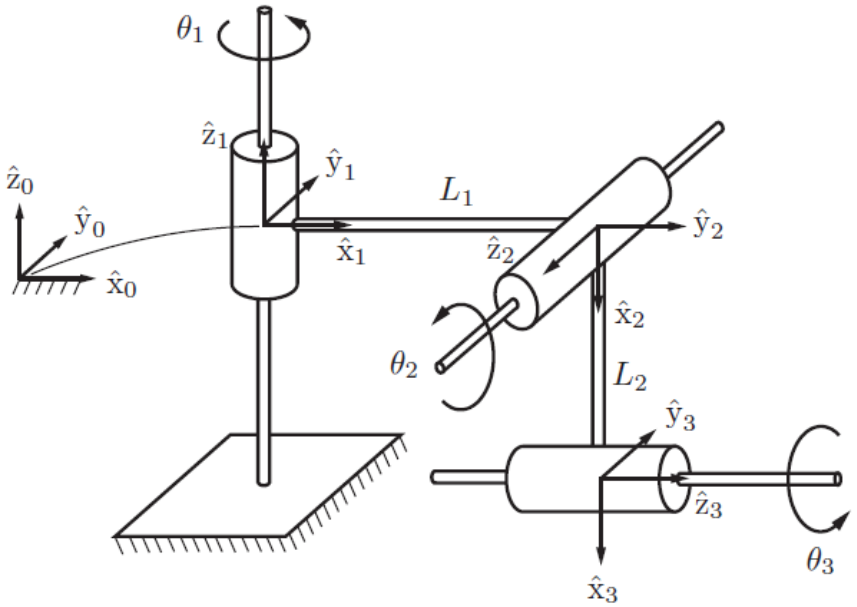
$$[\mathcal{S}_1] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; [\mathcal{S}_2] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -L_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; [\mathcal{S}_3] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -(L_1+L_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Robot 3R 3D

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} M$$

$$M = \begin{bmatrix} 0 & 0 & 1 & L_1 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(0,-1,0)$	$(L_1,0,0)$	$(0,0,-L_1)$
3	$(1,0,0)$	$(0,0,-L_2)$	$(0,-L_2,0)$



$$[\mathcal{S}_1] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; [\mathcal{S}_2] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -L_1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; [\mathcal{S}_3] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

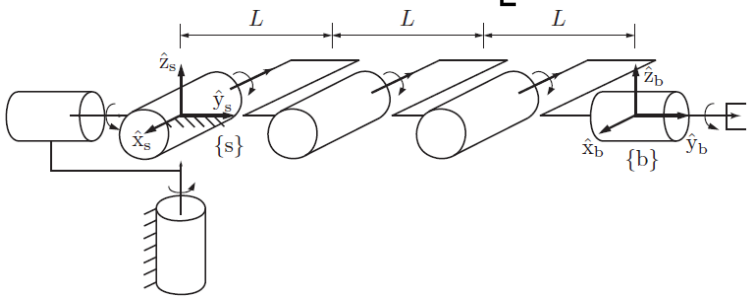
Robots de 6 dof (manipuladores de propósito general): 6R y 2RP3R

Estos Robots tienen el dof mínimo para mover el elemento terminal con todos los grados de libertad de un sólido rígido.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3L \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} e^{[\mathcal{S}_4]\theta_4} e^{[\mathcal{S}_5]\theta_5} e^{[\mathcal{S}_6]\theta_6} M$$

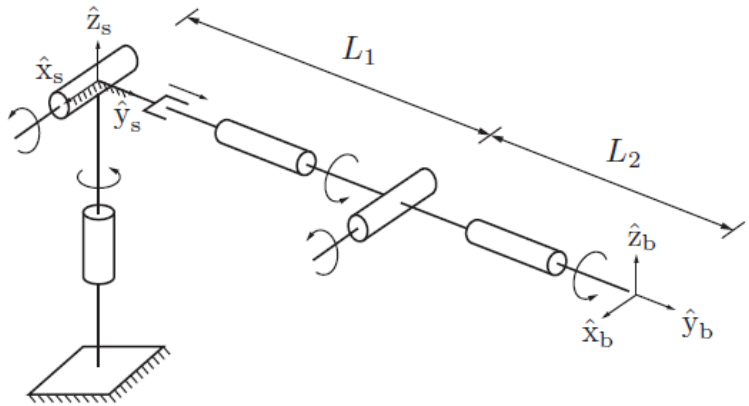
i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$
3	$(-1,0,0)$	$(0,0,0)$	$(0,0,0)$
4	$(-1,0,0)$	$(0,L,0)$	$(0,0,L)$
5	$(-1,0,0)$	$(0,2L,0)$	$(0,0,2L)$
6	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$



$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 + L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

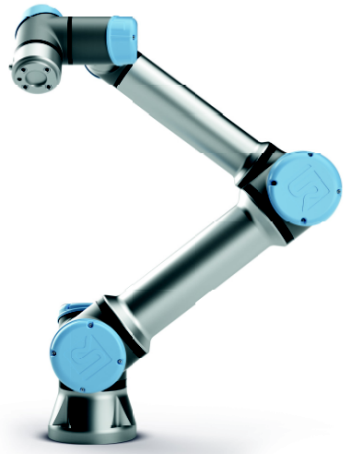
$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} e^{[\mathcal{S}_3]\theta_3} e^{[\mathcal{S}_4]\theta_4} e^{[\mathcal{S}_5]\theta_5} e^{[\mathcal{S}_6]\theta_6} M$$

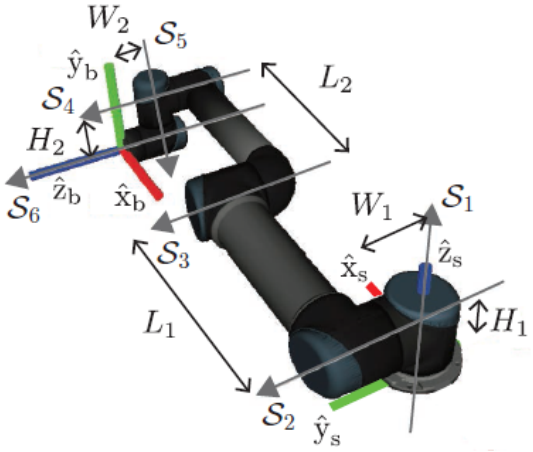
i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(1,0,0)$	$(0,0,0)$	$(0,0,0)$
3	$(0,0,0)$	$--$	$(0,1,0)$
4	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$
5	$(1,0,0)$	$(0,L_1,0)$	$(0,0,-L_1)$
6	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$

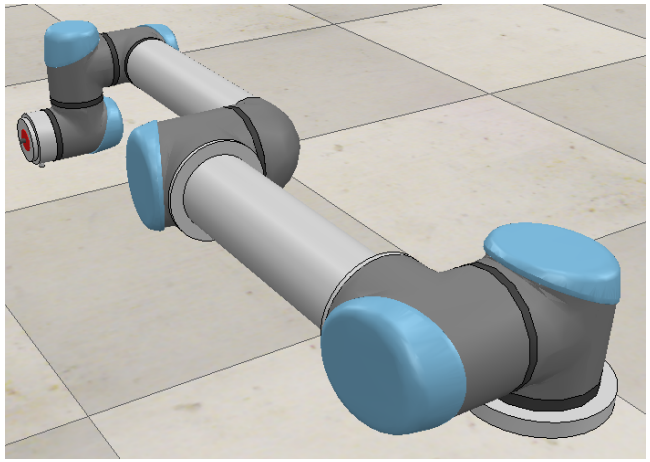


Robot UR5 6R

https://www.universal-robots.com/products/ur5-robot/







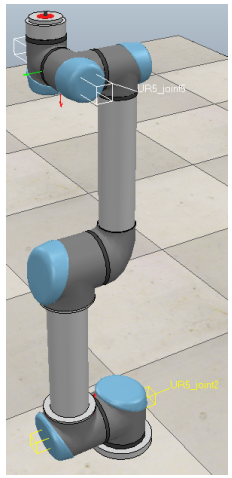
i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(0,1,0)$	$(0,0,H_1)$	$(-H_1,0,0)$
3	$(0,1,0)$	$(L_1,0,H_1)$	$(-H_1,0,L_1)$
4	$(0,1,0)$	$(L_1+L_2,0,H_1)$	$(-H_1,0,L_1+L_2)$
5	$(0,0,-1)$	$(L_1+L_2,W_1,0)$	$(-W_1,L_1+L_2,0)$
6	$(0,1,0)$	$(L_1+L_2,0,H_1-H_2)$	$(H_2-H_1,0,L_1+L_2)$

$$M=\begin{bmatrix}-1 & 0 & 0 & L_1+L_2\\ 0 & 0 & 1 & W_1+W_2\\ 0 & 1 & 0 & H_1-H_2\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Calcula la posición del elemento terminal para $\theta_2=-\pi/2$ y $\theta_5=\pi/2$ con $\theta_i=0$ para $i=1,3,4,6 \Rightarrow$ resolver el problema cinemático directo. Como $e^0=\mathbb{I} \Rightarrow T(\theta)=\mathbb{I}\, e^{[\mathcal{S}_2]\theta_2}\,\mathbb{I}\,\mathbb{I}\, e^{[\mathcal{S}_5]\theta_5}\,\mathbb{I}\, M$

Robot UR5 6R

Las dimensiones del Robot son: $W_1 = 109\,mm$; $W_2 = 82\,mm$; $L_1 = 425\,mm$; $L_2 = 392\,mm$; $H_1 = 89\,mm$; $H_2 = 95\,mm$;



$$e^{-[\mathcal{S}_2]\pi/2} = \begin{bmatrix} 0 & 0 & -1 & 0.089 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0.089 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \; e^{[\mathcal{S}_5]\pi/2} = \begin{bmatrix} 0 & 1 & 0 & 0.708 \\ -1 & 0 & 0 & 0.926 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

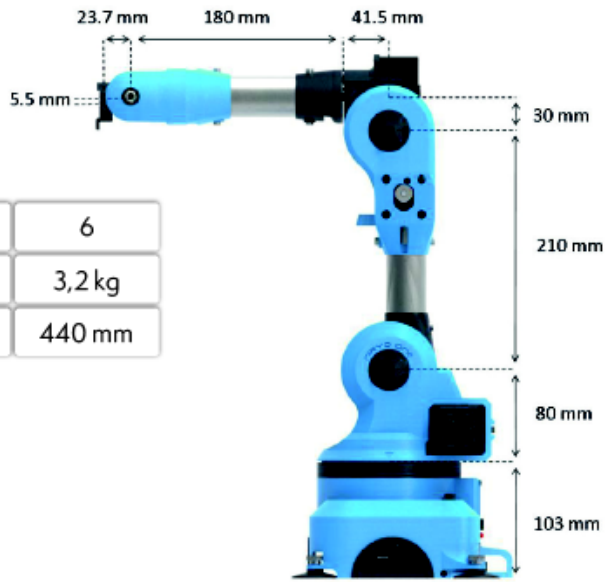
$$M = \begin{bmatrix} -1 & 0 & 0 & 0.817 \\ 0 & 0 & 1 & 0.191 \\ 0 & 1 & 0 & -0.006 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad T(\theta) = e^{-[\mathcal{S}_2]\pi/2} e^{[\mathcal{S}_5]\pi/2} M = \begin{bmatrix} 0 & -1 & 0 & 0.095 \\ 1 & 0 & 0 & 0.109 \\ 0 & 0 & 1 & 0.988 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se puede comprobar que la solución coincide con la configuración del robot.

Robot Niryo One: https://niryo.com/niryo-one/

Este es el brazo robótico con el que trabajarás en el laboratorio.

Calcula la posición del elemento terminal para las coordenadas intermedias, máximas y mínimas de todas las articulaciones.

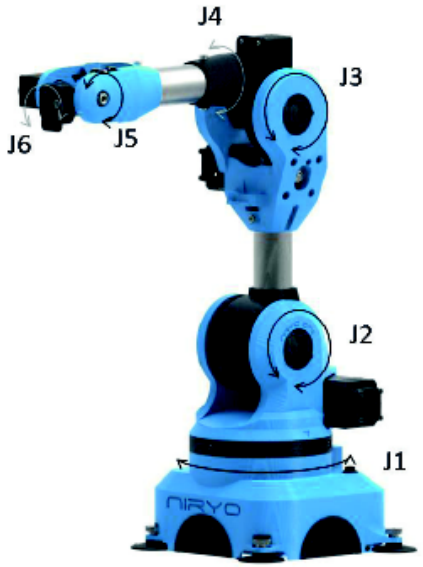


DEGREES OF FREEDOM	6
WEIGHT	3,2 kg
MAXIMUM REACH	440 mm

MAXIMUM ROTATIONS

Each axis has a movement called J:

	MIN	MAX
J1	-175°	175°
J2	-90°	36,7°
J3	-80°	90°
J4	-175°	175°
J5	-100°	100°
J6	-147,5°	147,5°



Cinemática Directa utilizando el elemento terminal como SR

Las matrices exponenciales verifican la siguiente identidad: $e^{(PDP^{-1})t} = Pe^{Dt}P^{-1}$
Si aplicamos esta identidad a la última matriz exponencial de la fórmula de producto de matrices exponenciales:

$$e^{[\mathcal{S}_n]\theta_n} M = M\, M^{-1} e^{[\mathcal{S}_n]\theta_n} M = M\, e^{M^{-1}[\mathcal{S}_n]M\theta_n}$$

Si utilizamos la definición de matriz adjunta:

$$M^{-1}[\mathcal{S}_n]M = [Ad_{M^{-1}}]\mathcal{S}_n = \mathcal{S}'_n = \beta_n \leftarrow \text{eje helicoidal en el SR del elemento terminal}$$

$$\Rightarrow e^{[\mathcal{S}_n]\theta_n} M = M\, e^{[\beta_n]\theta_n}$$

y repitiendo el proceso iterativamente para todas las exponenciales de la fórmula del producto:

$$\Rightarrow \boxed{T(\theta) = M\, e^{[\beta_1]\theta_1} \ldots e^{[\beta_{n-1}]\theta_{n-1}} e^{[\beta_n]\theta_n}}$$

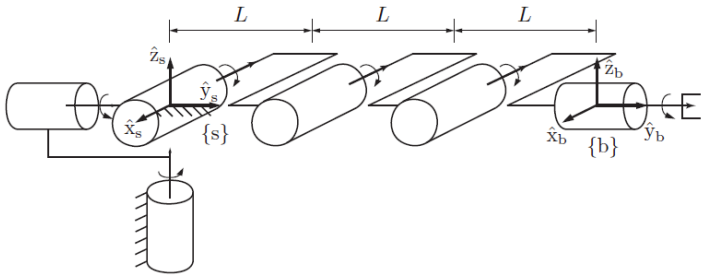
donde β_i son los ejes helicoidales en el SR del elemento terminal.

- Cuando utilizamos el SR fijo en la base del robot $\{s\}$, la matriz que define la posición del elemento terminal con respecto a $\{s\}$ se transforma iterativamente desde el último elemento hasta el primero multiplicándose por las matrices exponenciales de los ejes helicoidales de las correspondientes articulaciones.
- Cuando utilizamos el SR localizado en el elemento terminal $\{b\}$, la matriz que define su posición con respecto a $\{s\}$ se transforma iterativamente desde la articulación más cercana a la base (la más distante al SR utilizado) hasta el elemento terminal.

Robot 6R

$$T(\theta) = M\, e^{[\beta_1]\theta_1}\, e^{[\beta_2]\theta_2}\, e^{[\beta_3]\theta_3}\, e^{[\beta_4]\theta_4}\, e^{[\beta_5]\theta_5}\, e^{[\beta_6]\theta_6}$$

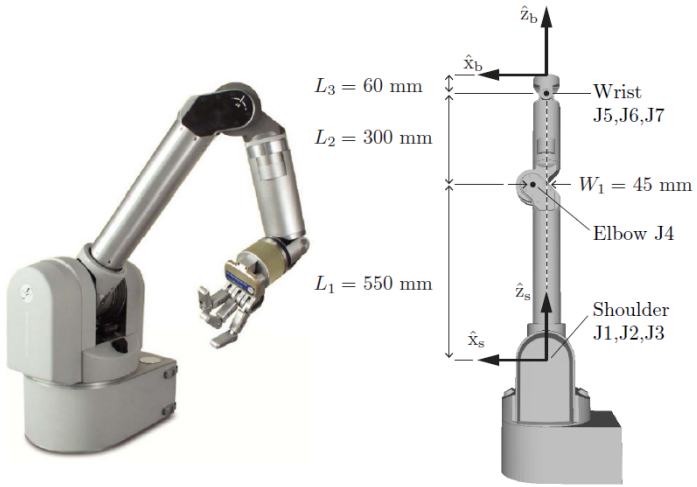
$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3L \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,-3L,0)$	$(-3L,0,0)$
2	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$
3	$(-1,0,0)$	$(0,-3L,0)$	$(0,0,-3L)$
4	$(-1,0,0)$	$(0,-2L,0)$	$(0,0,-2L)$
5	$(-1,0,0)$	$(0,-L,0)$	$(0,0,-L)$
6	$(0,1,0)$	$(0,0,0)$	$(0,0,0)$

Robot WAM 7R (Barrett Technology)

Se trata de un robot de 7 dof $\Rightarrow$ es un robot redundante. El dof extra puede utilizarse para evitar obstáculos o para minimizar la potencia del motor que se necesita para que el elemento terminal alcance ciertas configuraciones.



$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 + L_2 + L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Utilizando el elemento terminal como SR:

i	ω_i	q_i	v_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(0,1,0)$	$(0,0,-L_1-L_2-L_3)$	$(L_1+L_2+L_3,0,0)$
3	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
4	$(0,1,0)$	$(0,0,-L_2-L_3)$	$(L_2+L_3,0,0)$
5	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
6	$(0,1,0)$	$(0,0,-L_3)$	$(L_3,0,0)$
7	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$

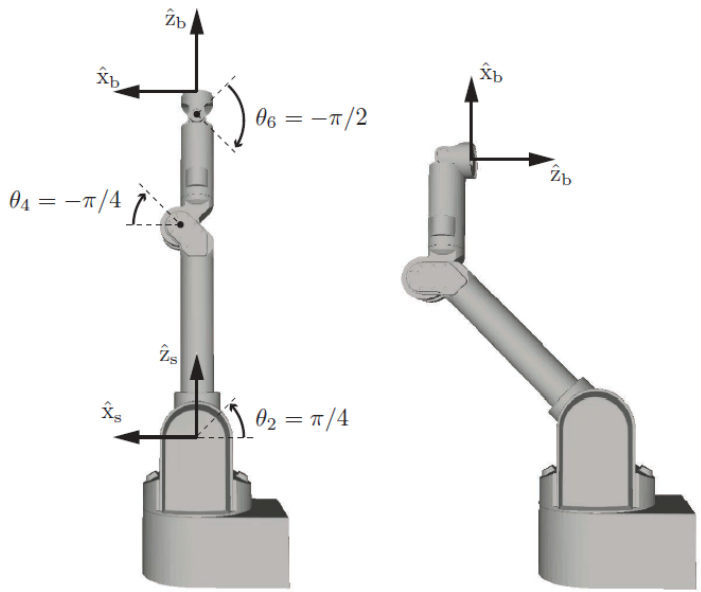
Robot WAM 7R (Barrett Technology)

Calcula la configuración del elemento terminal para $\theta_2 = \pi/4$, $\theta_4 = -\pi/4$, $\theta_6 = -\pi/2$, y $\theta_1 = \theta_3 = \theta_5 = \theta_7 = 0$

$$\Rightarrow T(\theta) = M\, e^{[\beta_2]\frac{\pi}{4}}\, e^{-[\beta_4]\frac{\pi}{4}}\, e^{-[\beta_6]\frac{\pi}{2}}$$

Resolviendo el producto de exponenciales:

$$T(\theta) = \begin{bmatrix} 0 & 0 & -1 & 0.328 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0.689 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Robot WAM 7R (Barrett Technology)

Resolución Cinemática Directa

- Crea la función *CinematicaDirectaS(M, Smatrix, theta)* en Python que calcule la configuración del elemento terminal como una Matriz de Transformación Homogénea a partir de la configuración del elemento terminal en la posición cero del Robot (M), de una matriz (Smatrix) en la que las columnas son los ejes helicoidales de las articulaciones cuando el robot está en la posición cero, y de la lista de coordenadas de las articulaciones (theta).
- Crea la función *CinematicaDirectaB(M, Bmatrix, theta)* análoga a la anterior pero expresando los ejes helicoidales de cada articulación en el SR del elemento terminal.