ROBOTICA INDUSTRIAL

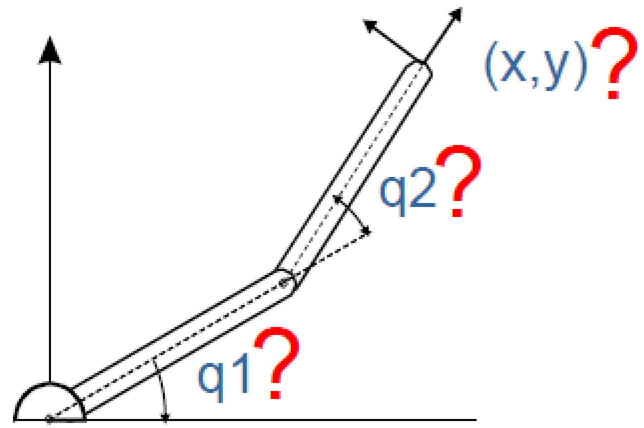
TEMA 2

MODELADO CINEMÁTICO

INTRODUCCIÓN

- **Cinemática del robot** : Estudio de su movimiento con respecto a un sistema de referencia **sin considerar las fuerzas** que intervienen
  - Relación entre la localización del extremo del robot y los valores de sus articulaciones
  - Descripción analítica del movimiento espacial en función del tiempo
- **Problema cinemático directo**: Determinar la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas de referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot
- **Problema cinemático inverso**: Determinar la configuración que debe adoptar el robot para alcanzar una posición y orientación del extremo conocidas
- **Modelo diferencial (matriz Jacobiana):** Relaciones entre las velocidades de movimiento de las articulaciones y las del extremo del robot

Cinemática directa e inversa



Cinemática Directa

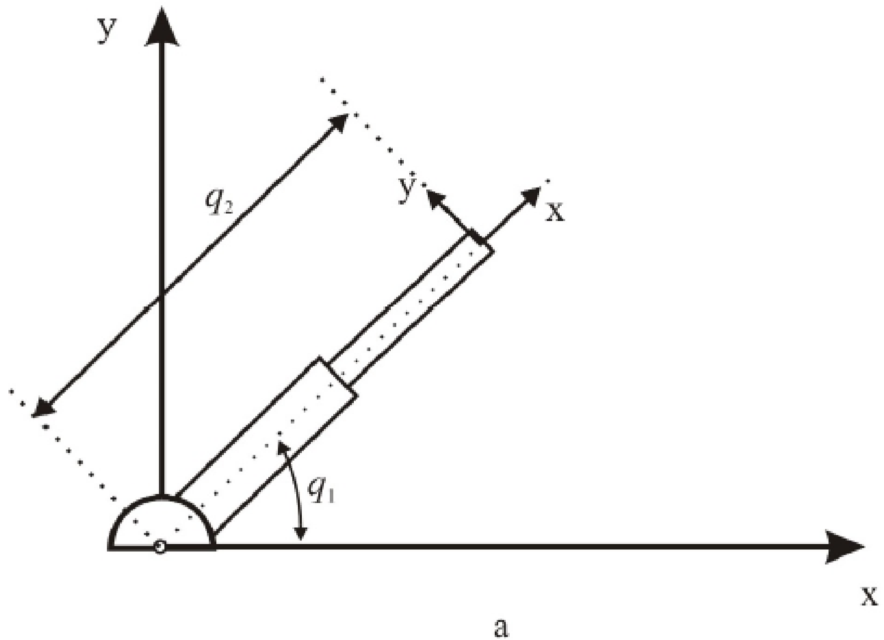
Coordenadas articulares $(q_1,q_2,\ldots,q_n)$ → Posición y orientación del extremo del robot $(x, y, z, \varphi,\theta,\psi)$

← Cinemática Inversa

Modelo cinemático directo. Cálculo

- **Métodos geométricos**
  - Método no sistemático (depende de la habilidad)
  - Consideraciones geométricas (trigonometría, etc.)
  - Válido para robots de pocos grados de libertad
- **Métodos basados en cambios de base (MTH o Cuaterníos)**
  - Puede ser sistematizado (Denavit-Hartenberg)
  - Aplicable a cualquier cadena cinemática (n gdl)
  - Ayuda de herramientas computacionales.

MCD por métodos geométricos

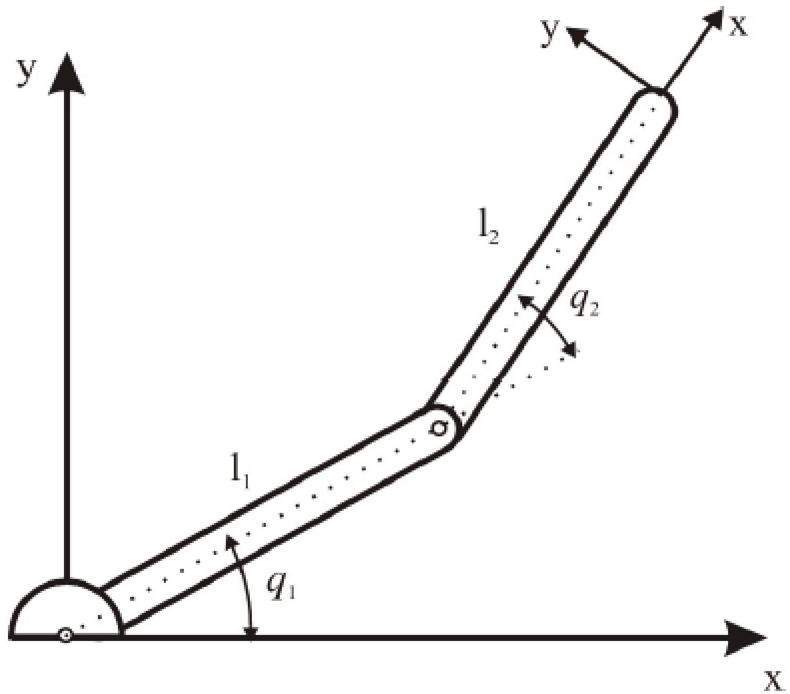


$$x = q_2 \cos q_1$$

$$y = q_2 \sin q_1$$

$$z = 0$$

MCD por métodos geométricos

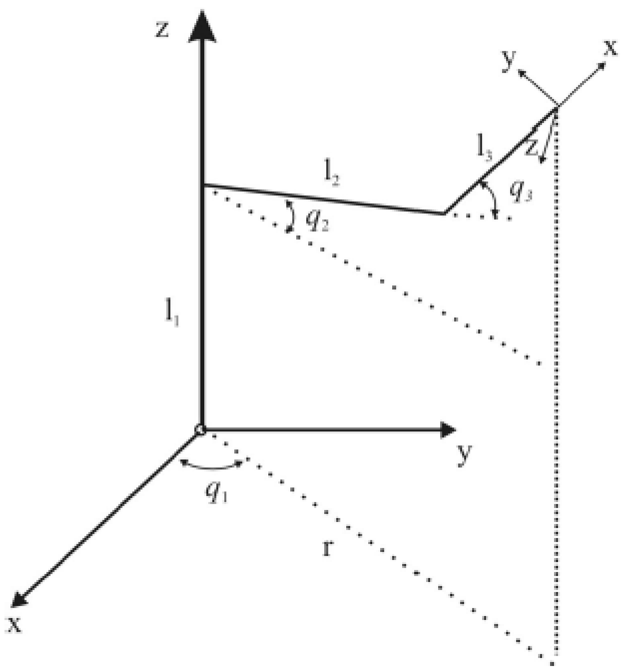


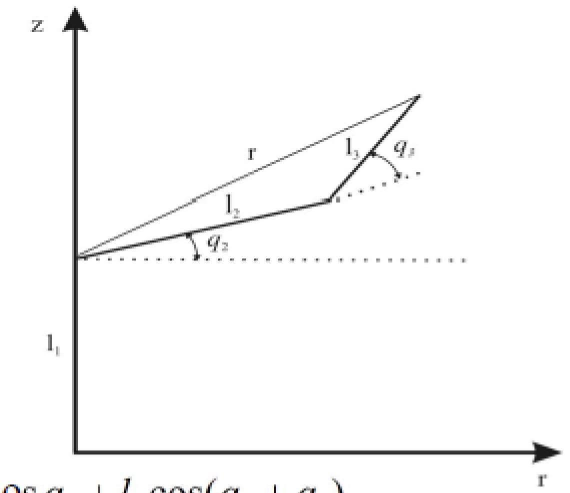
$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2)$$

$$y = l_1 sen q_1 + l_2 sen(q_1 + q_2)$$

$$z = 0$$

MCD por métodos geométricos

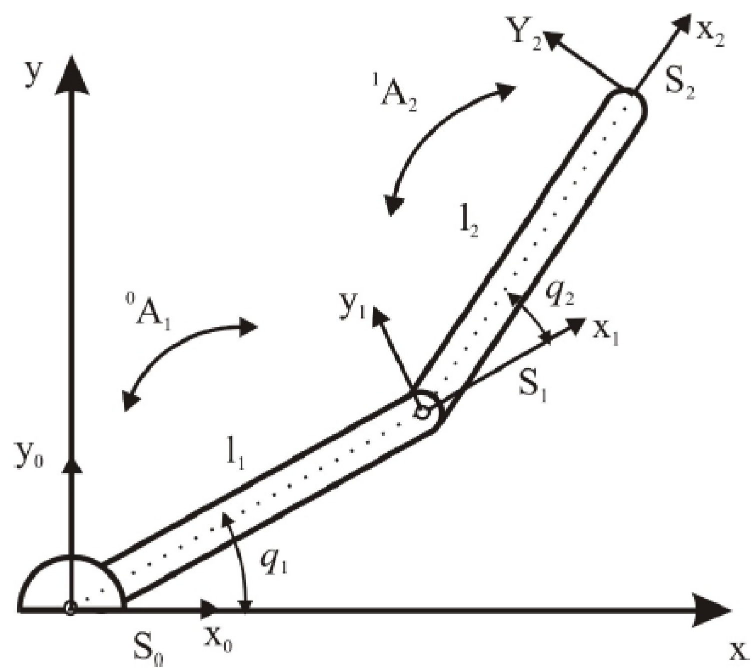




$$r = l_2 \cos q_2 + l_3 \cos(q_2 + q_3)$$
$$z = l_1 + l_2 \operatorname{sen} q_2 + l_3 \operatorname{sen}(q_2 + q_3)$$
$$x = r \cos q_1$$
$$y = r \operatorname{sen} q_1$$

MCD mediante cambios de base

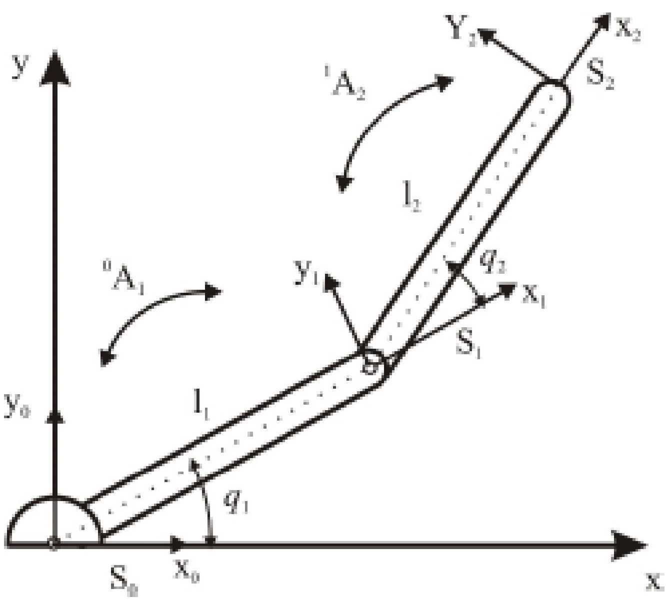
- Asociar a cada eslabón del robot un sistema de referencia solidario a él.
- De un sistema al siguiente se pasa mediante un cambio de base definido por rotaciones y traslaciones.
- Estos cambios de base dependerán de las dimensiones del robot y de los valores de las variables articulares q_n
- Encontrar la MTH (o cuaternio-vector) que define el cambio de base de un sistema al siguiente.
- Multiplicar los sucesivos cambios de base en el orden adecuado, para obtener el cambio de base entre el origen y el extremo del robot. Este dependerá de los n grados de libertad q_n
- Y representará la relación entre la posición y orientación del extremo en el sistema de la base, en función de las coordenadas articulares (MCD)



$$ {}^0\mathbf{A}_2 = {}^0\mathbf{A}_1 \, {}^1\mathbf{A}_2 $$

Sistema móvil

MCD mediante MTH



$$^{0}\mathbf{A}_1 = \mathbf{Rotz}(q_1)\cdot \mathbf{T}(l_1,0,0) =$$

$$= \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_1 & -S_1 & 0 & l_1C_1 \\ S_1 & C_1 & 0 & l_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}\mathbf{A}_2 = \mathbf{Rotz}(q_2)\cdot \mathbf{T}(l_2,0,0) =$$

$$= \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_2 & -S_2 & 0 & l_2C_2 \\ S_2 & C_2 & 0 & l_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Posición(x_2, y_2, z_2)

$$T = {}^{0}A_1\ {}^{1}A_2 = \begin{bmatrix} C_1S_2 - S_1S_2 & -C_1S_2 - S_1C_2 & 0 & l_1C_1 + l_2C_{12} \\ S_1C_2 + C_1S_2 & -S_1S_2 + C_1C_2 & 0 & l_1S_1 + l_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1C_1 + l_2C_{12} \\ S_{12} & C_{12} & 0 & l_1S_1 + l_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Orientación

MCD mediante Denavit-Hatenverg

- Un procedimiento de obtención del MCD basado en la realización de cambios de base mediante MTH
- **Sistematiza la selección de los sistemas de coordenadas**, garantizando que de uno a otro se pasa mediante una **secuencia concreta de 4 movimientos simples** (rotación o traslación entorno a ejes concretos).
- Cada movimiento simple depende de un parámetro
- De este modo la matriz de cambio de base de un sistema a otro $^{i-1}A_i$ responde a una expresión predefinida, función de los **4 parámetros**.
- Permite el utilizar un convenio estandarizado, que sirve de:
  - Lenguaje común
  - Desarrollo de herramientas de cálculo
- Hartenberg, R. S., & Denavit, J. (1955). A kinematic notation for lower pair mechanisms based on matrices.

MCD mediante Denavit-Hatenverg

- **Procedimiento general:**

1. Establecer para cada elemento del robot un sistema de coordenadas cartesiano ortonormal (x_i, y_i, z_i) donde $i = 0,1,2, \dots, n$ (n=número de gdl). Cada sistema de coordenadas corresponderá a la articulación $i + 1$ y estará fijo en el elemento i (Algoritmo de D-H)
2. Encontrar los parámetros D-H de cada una de las articulaciones
3. Calcular, a partir de los parámetros, las matrices ${}^{i-1}A_i$
4. Calcular la matriz $\boldsymbol{T_n = {}^0A_1 \, {}^1A_2 \ldots {}^{n-1}A_n}$
5. $\boldsymbol{T_n}$ expresa el modelo cinemático directo

MCD mediante Denavit-Hatenverg.
Transformaciones básicas

- Transformaciones básicas en cada articulación:

(definidas sobre sistema móvil =>**postmultiplicar**):

1 – Rotación alrededor del eje z_{i-1} un ángulo θ_i

2 – Traslación a lo largo de z_i una distancia d_i

3 – Traslación a lo largo de x_i una distancia a_i

4 – Rotación alrededor del eje x_i un ángulo α_i

MCD mediante Denavit-Hatenverg.

Transformaciones básicas

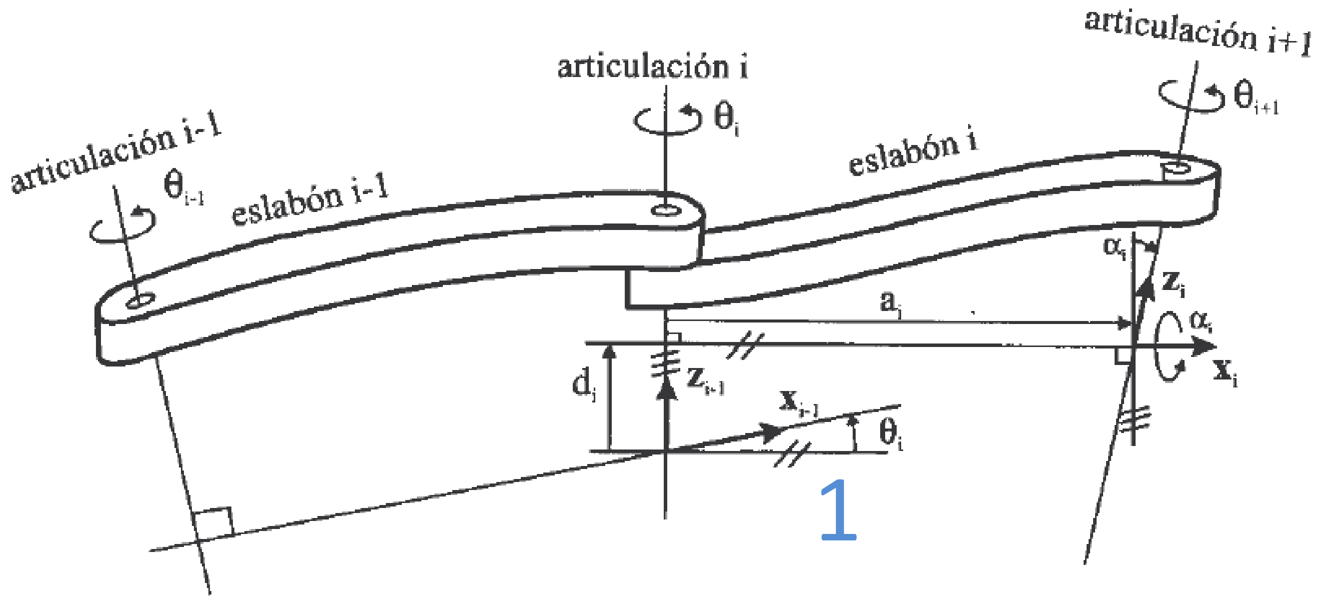
$$\overset{}{{}^{i-1}\mathbf{A}_i} = \overset{1}{\mathbf{Rotz}(\theta_i)}\,\overset{2}{\mathbf{T}(0,0,d_i)}\,\overset{3}{\mathbf{T}(a_i,0,0)}\,\overset{4}{\mathbf{Rotx}(\alpha_i)}$$

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\boxed{= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}} \Rightarrow \text{Matriz de D-H}$$

MCD mediante Denavit-Hatenverg.

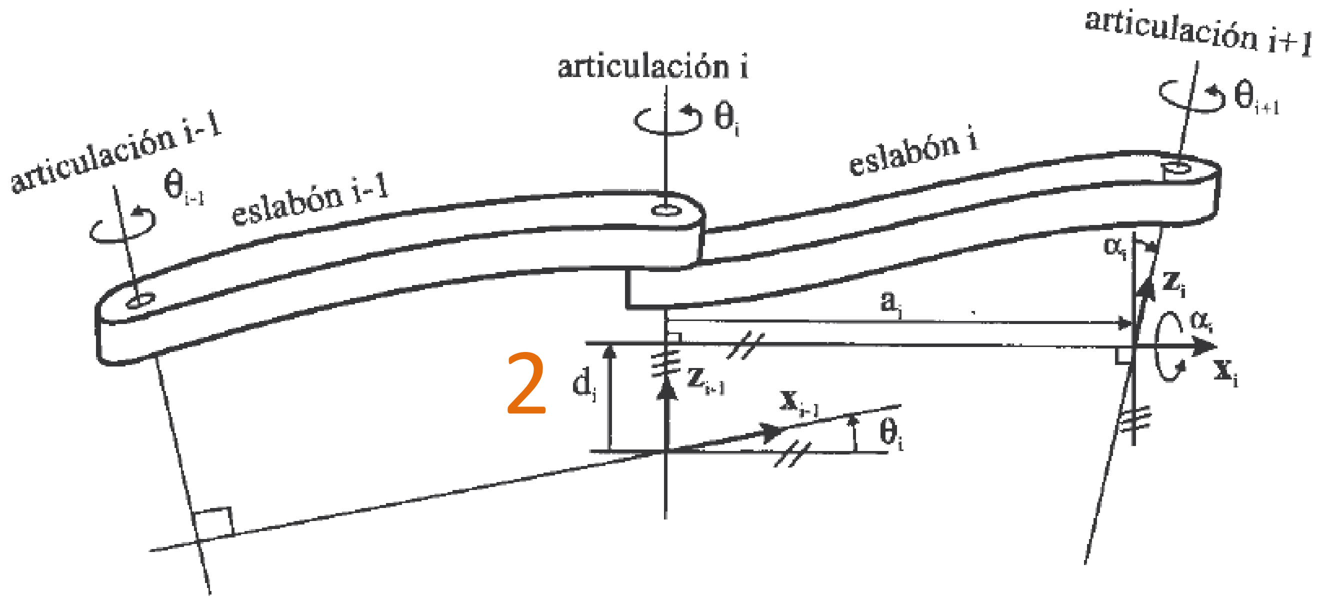
Parámetros



θ_i **Es el ángulo que forman los ejes xi-1 y xi medido en un plano perpendicular al eje zi-1, utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.**

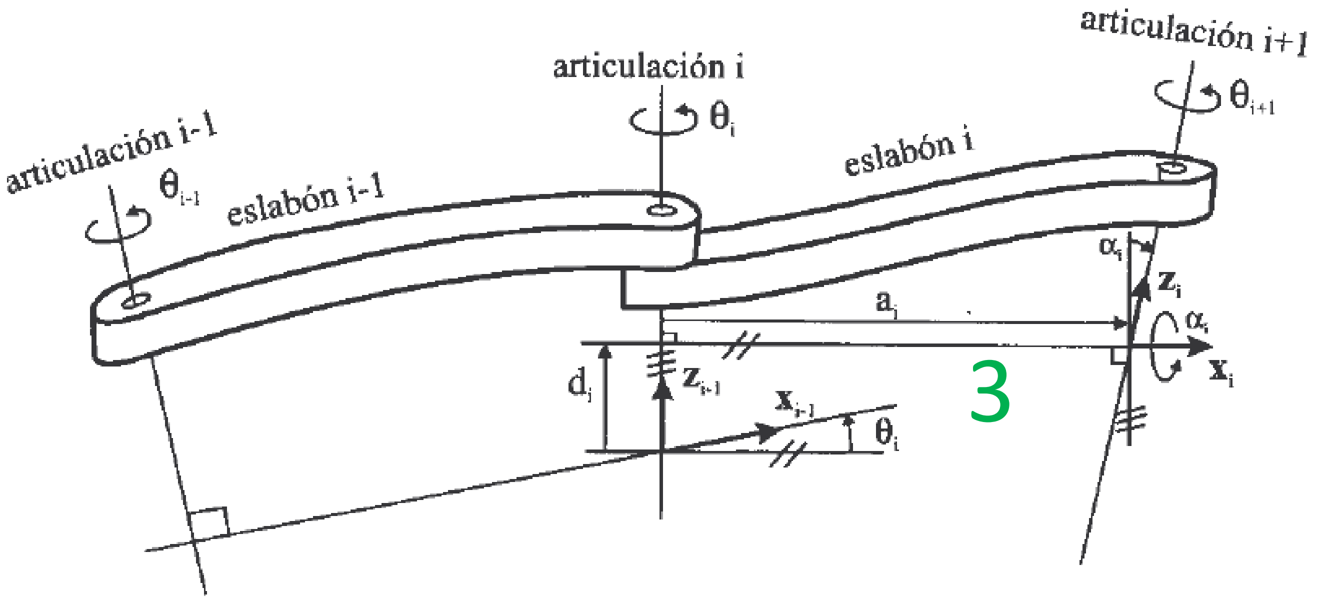
MCD mediante Denavit-Hatenverg.

Parámetros



d_i **Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas (i-1)-ésimo hasta la intersección del eje z_{i-1} con el eje x_i. Se trata de un parámetro variable en articulaciones prismáticas**

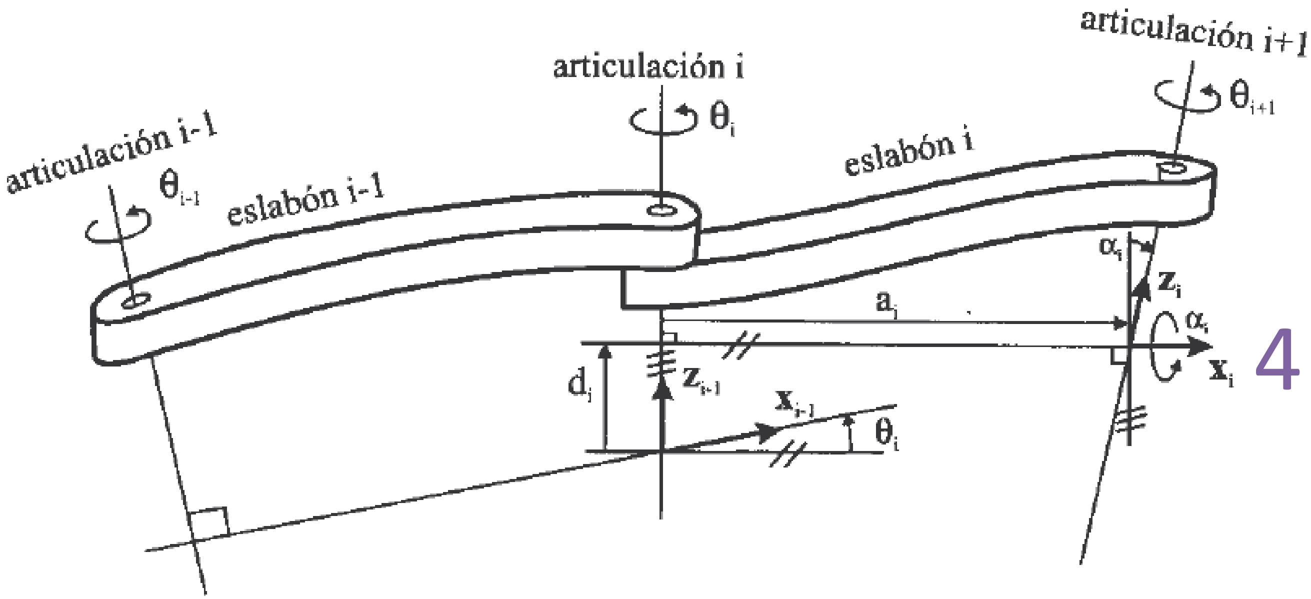
MCD mediante Denavit-Hatenverg.
Parámetros



a_i **Es la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i-ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i.**

MCD mediante Denavit-Hatenverg.

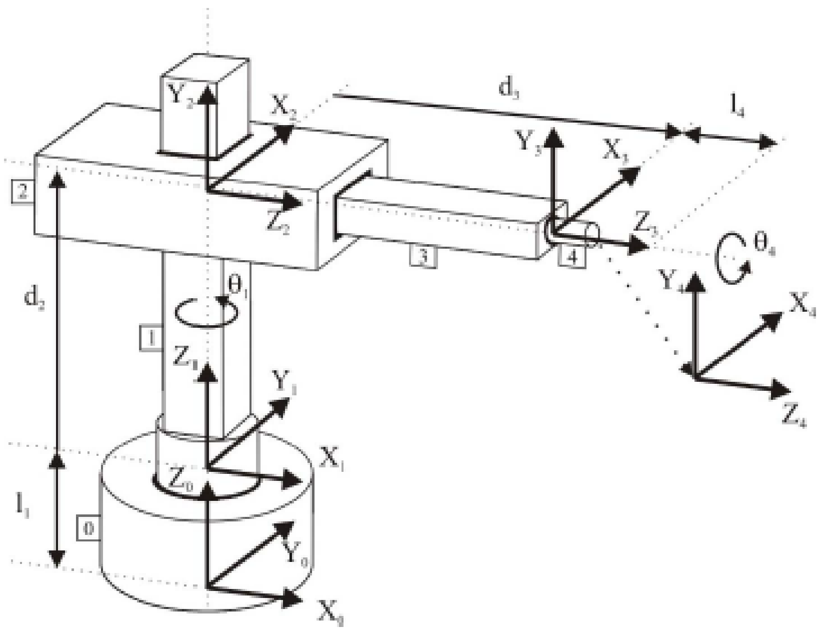
Parámetros



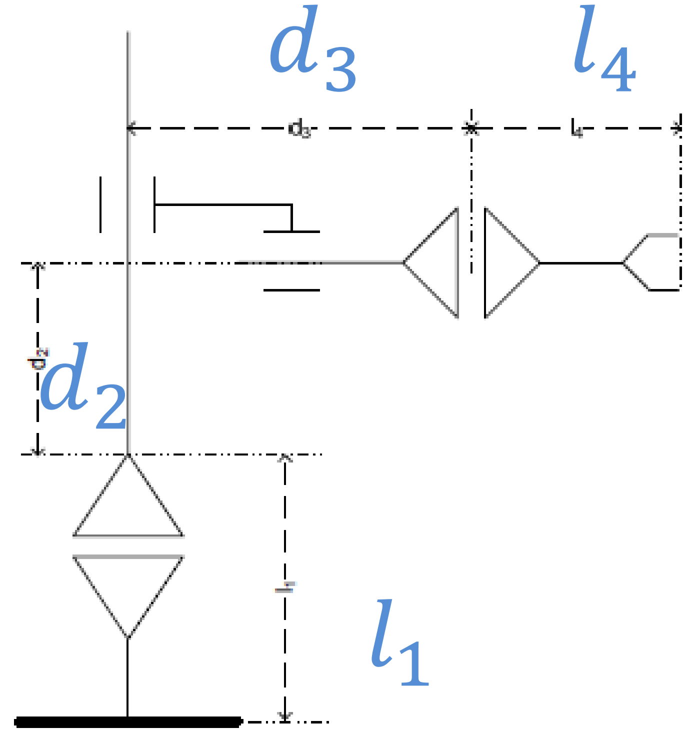
α_i **Es el ángulo de separación del** z_{i-1} **y el eje** z_i**, medido en un plano perpendicular al eje** x_i**, utilizando la regla de la mano derecha**

MCD mediante Denavit-Hatenverg.

Algoritmo con ejemplos



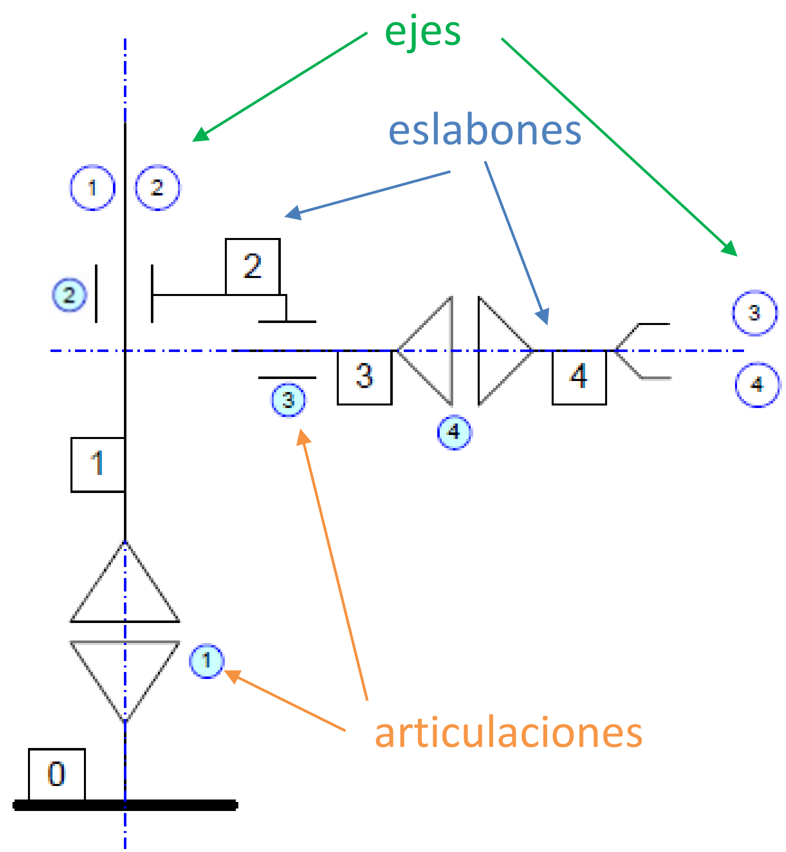
Robot cilíndrico 4 gdl



Representación esquemática

MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

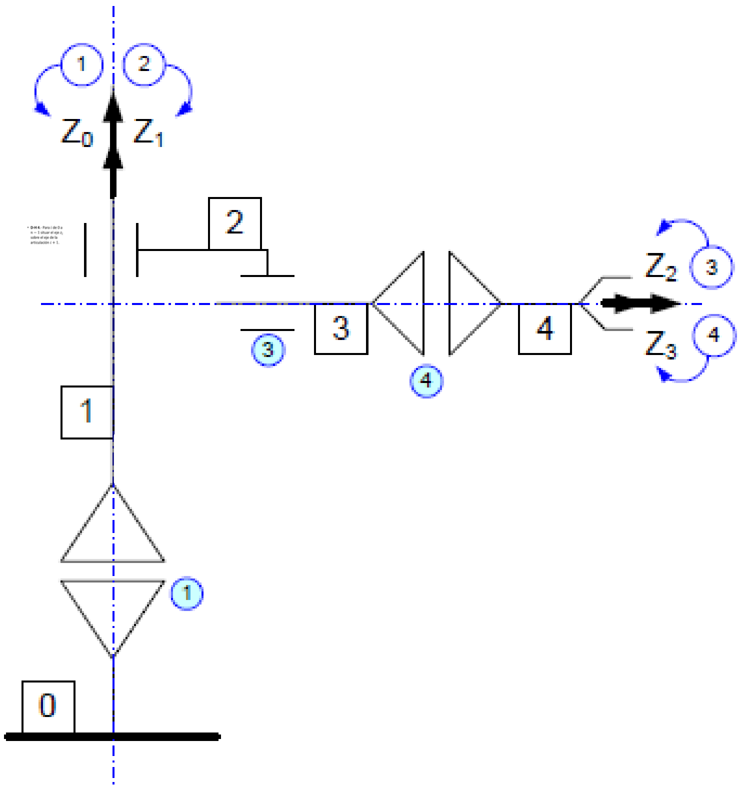
- **D-H 1.-** Numerar los **eslabones** comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- **D-H 2.-** Numerar cada **articulación** comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n
- **D-H 3**.- Localizar el **eje** de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento



MCD mediante Denavit-Hatenverg.

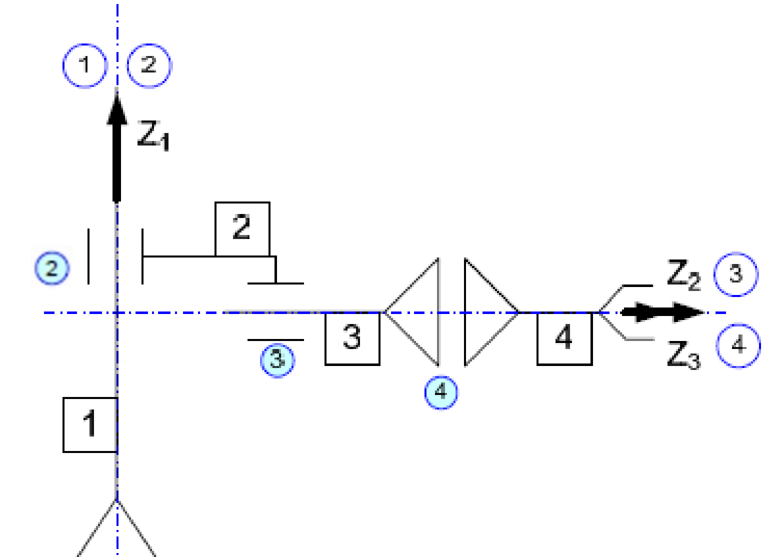
Algoritmo con ejemplos

- **D-H 4**.- Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i+1$.



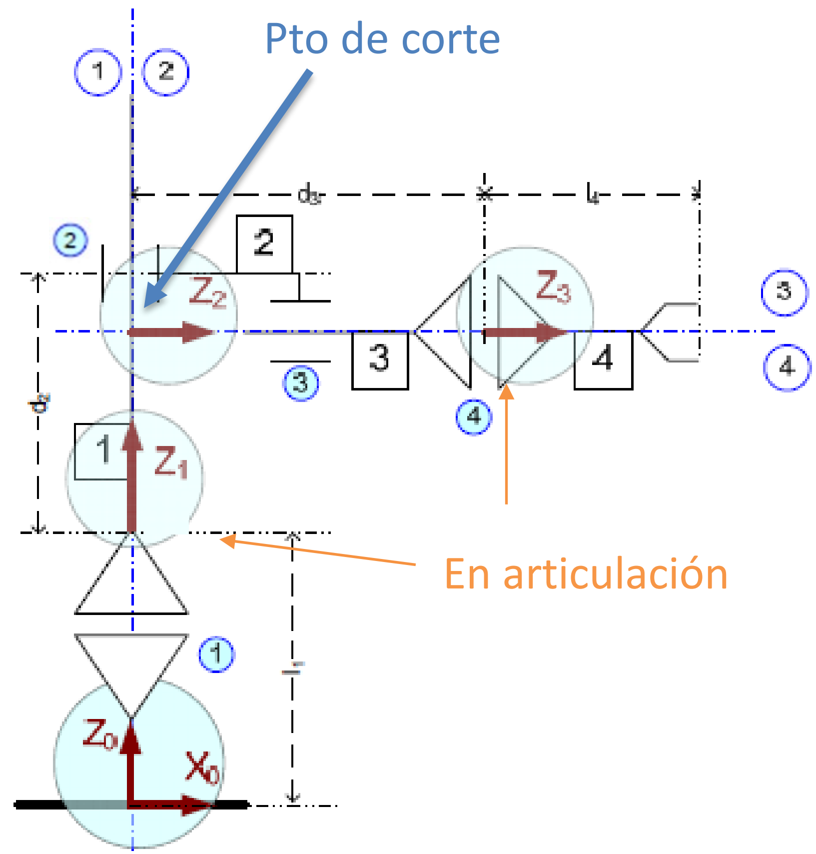
MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

- **D-H 5**.- Situar el origen del sistema de la base {S0} en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0



MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

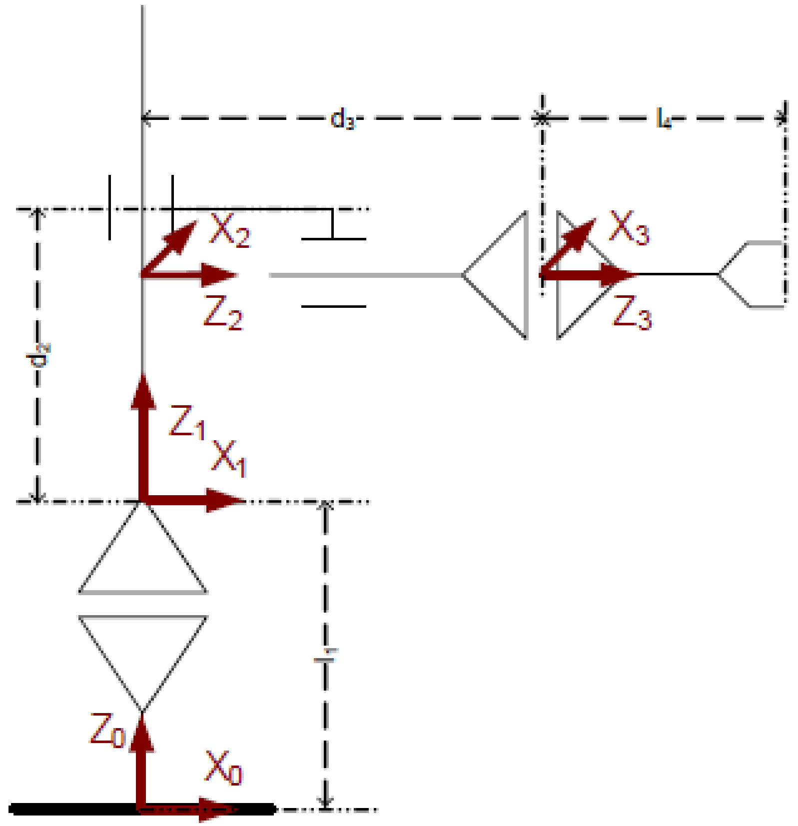
- **D-H 6**.- Para i de 1 a $n - 1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i. Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i + 1$



MCD mediante Denavit-Hatenverg.

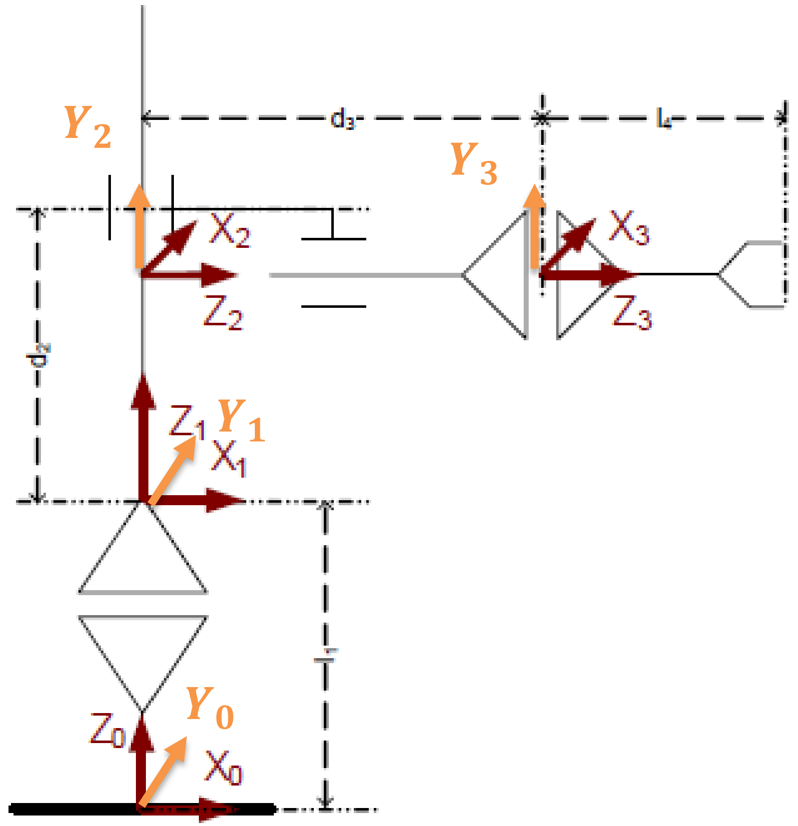
Algoritmo con ejemplos

- **D-H 7**.- Situar x_i en la línea normal común a z_{i-1} y z_i
- $(x_i \perp z_i) \wedge (x_i \perp z_{i-1})$



MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

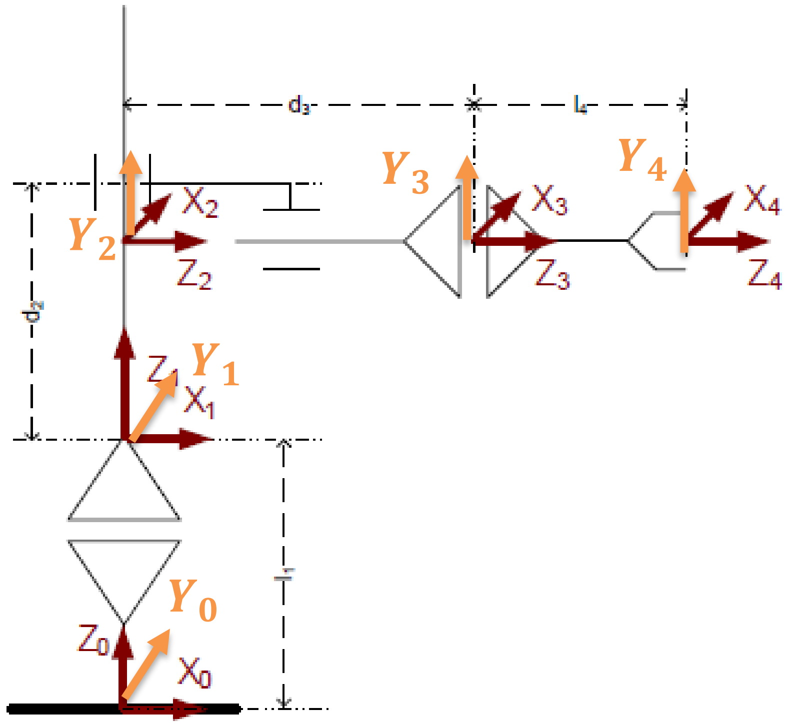
- **D-H 8**.- Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .



MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

- **D-H 9**.- Situar el sistema $\{S_n\}$ en el extremo del robot

de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n

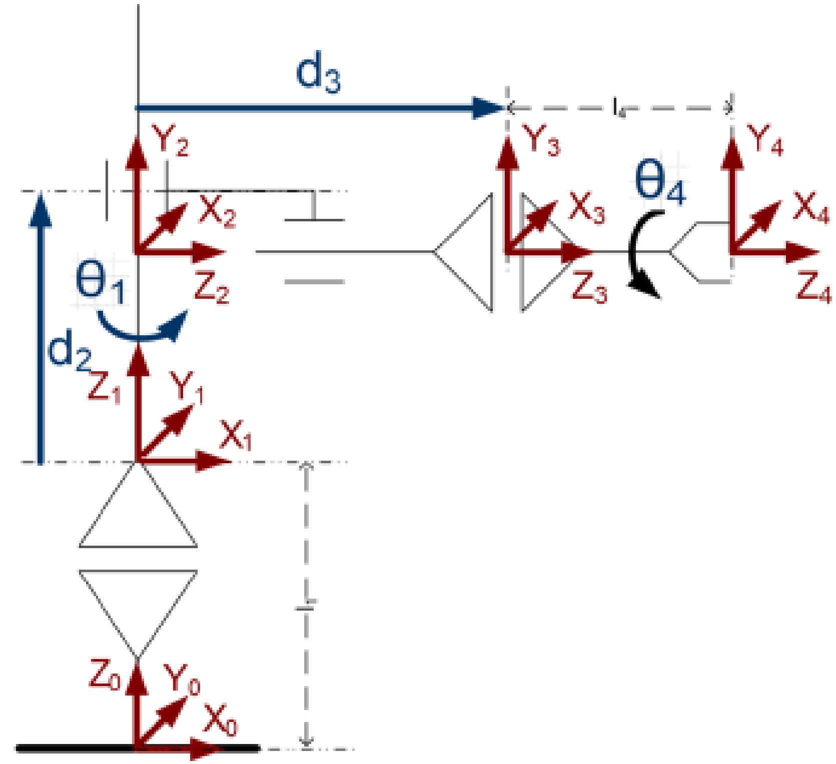


MCD mediante Denavit-Hatenverg.

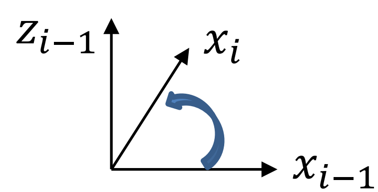
Algoritmo con ejemplos

- **D-H 10.-** Obtener $\boldsymbol{\theta_i}$ como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.

Articulación	θ
1	q_1
2	90º
3	0
4	q_4



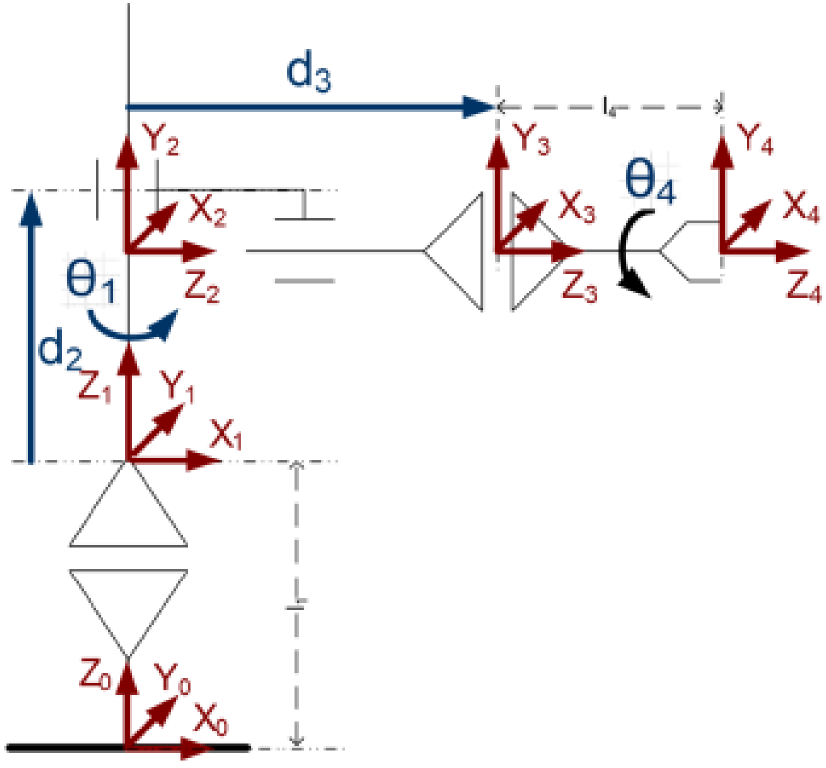
- **Ángulo + →** x_{i-1} a x_i regla mano derecha



MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

- **D-H 11.**- Obtener $\boldsymbol{d_i}$ como la distancia, medida a lo largo de z_{i-1}, que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.

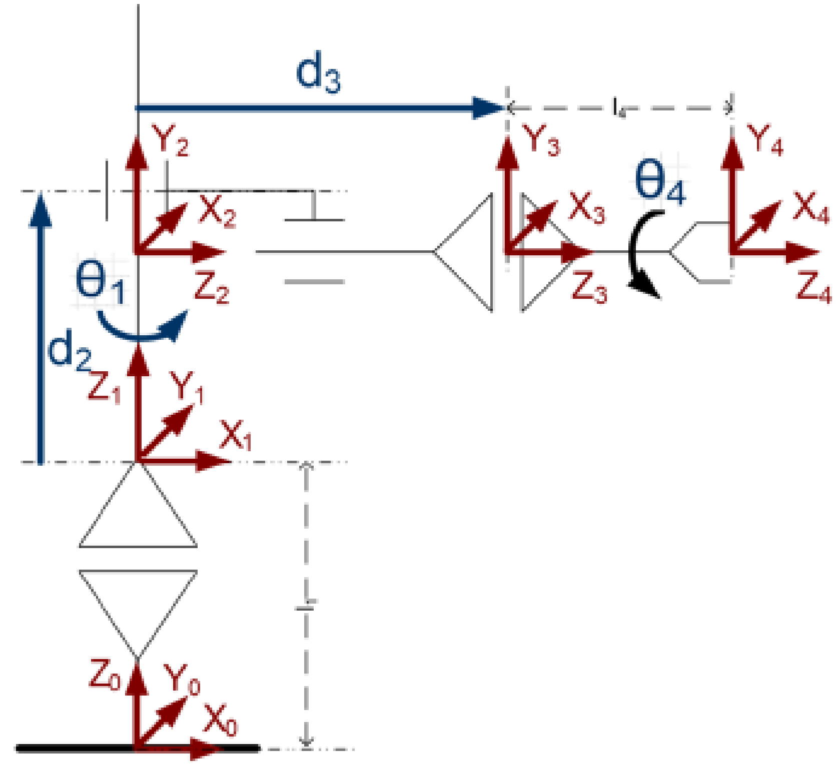
Articulación	d
1	l_1
2	d_2
3	d_3
4	l_4



MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

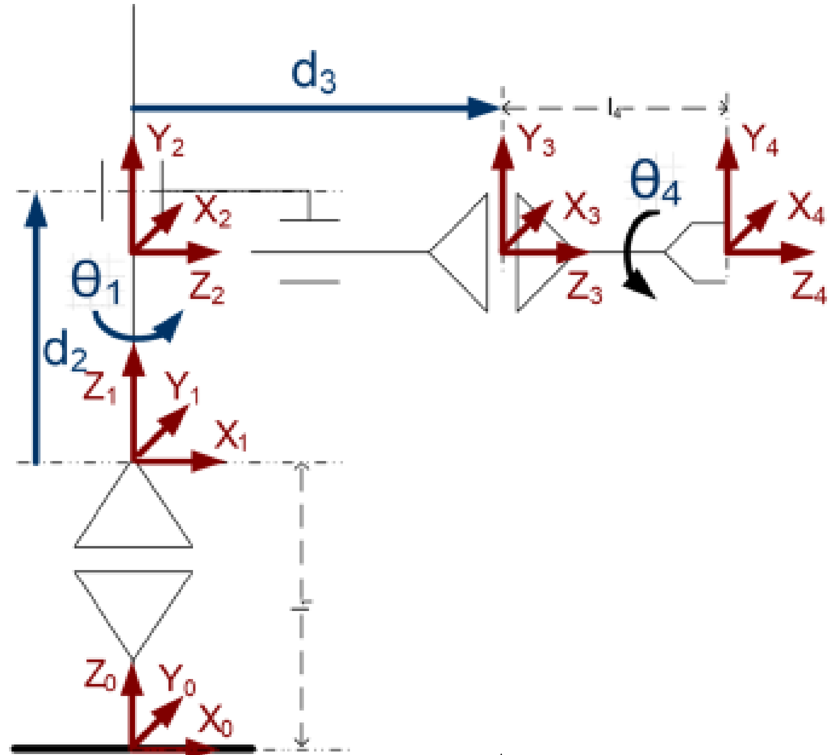
- **DH 12.-** Obtener $\boldsymbol{a_i}$ como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$

Articulación	a
1	0
2	0
3	0
4	0

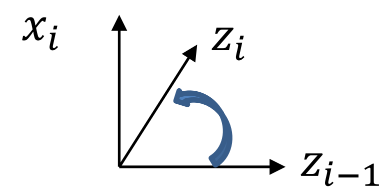


MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

DH 13.- Obtener $\boldsymbol{\alpha_i}$ como el ángulo que habría que girar entorno a x_i (que ahora coincidiría con x_{i-1}), para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$. $\boldsymbol{\alpha_i}$ es el ángulo de separación del z_{i-1} y el eje z_i, medido en un plano perpendicular al eje x_i, utilizando la regla de la mano derecha

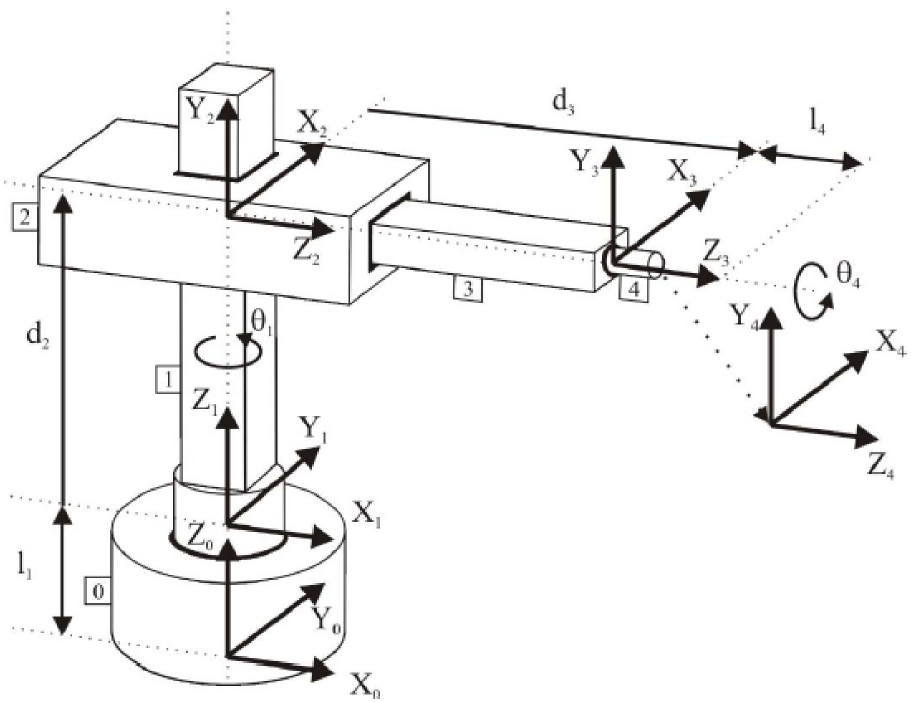


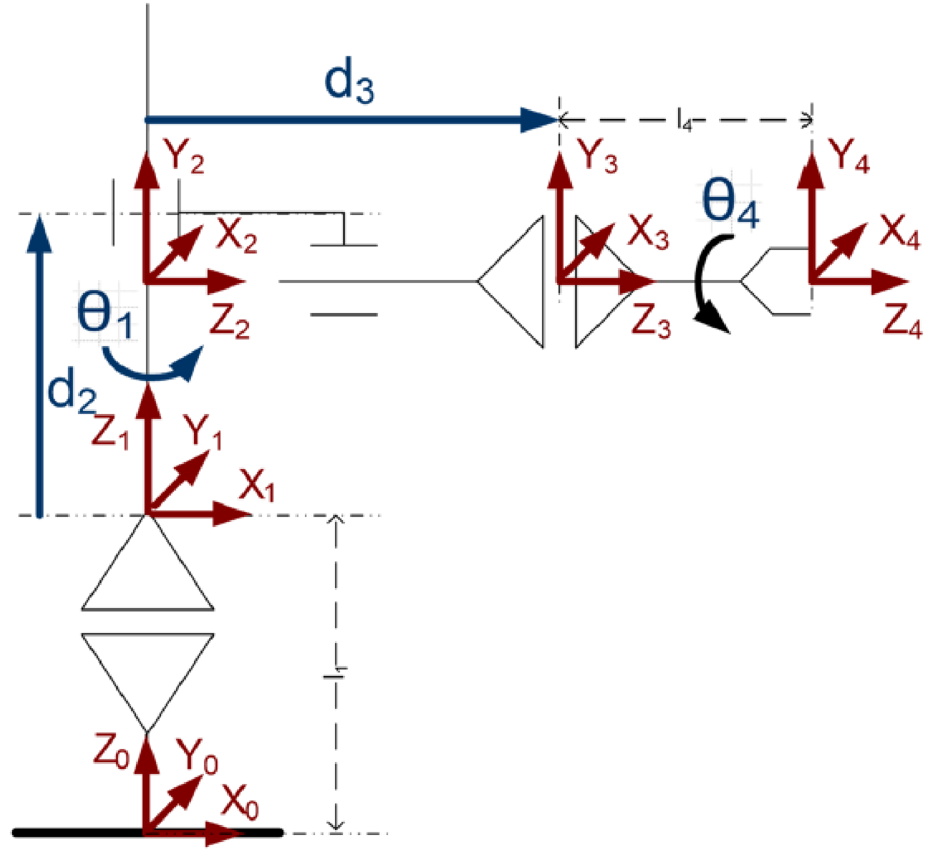
Articulación	α
1	0
2	90º
3	0
4	0



- **Ángulo + →** z_{i-1} a z_i regla mano derecha

MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos





MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

DH 14.- Obtener las matrices de transformación $^{n-1}\boldsymbol{A}_n$

$$\mathbf{A}_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0\mathbf{A}_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^1\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathrm{d}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^2\mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^3\mathbf{A}_4 = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

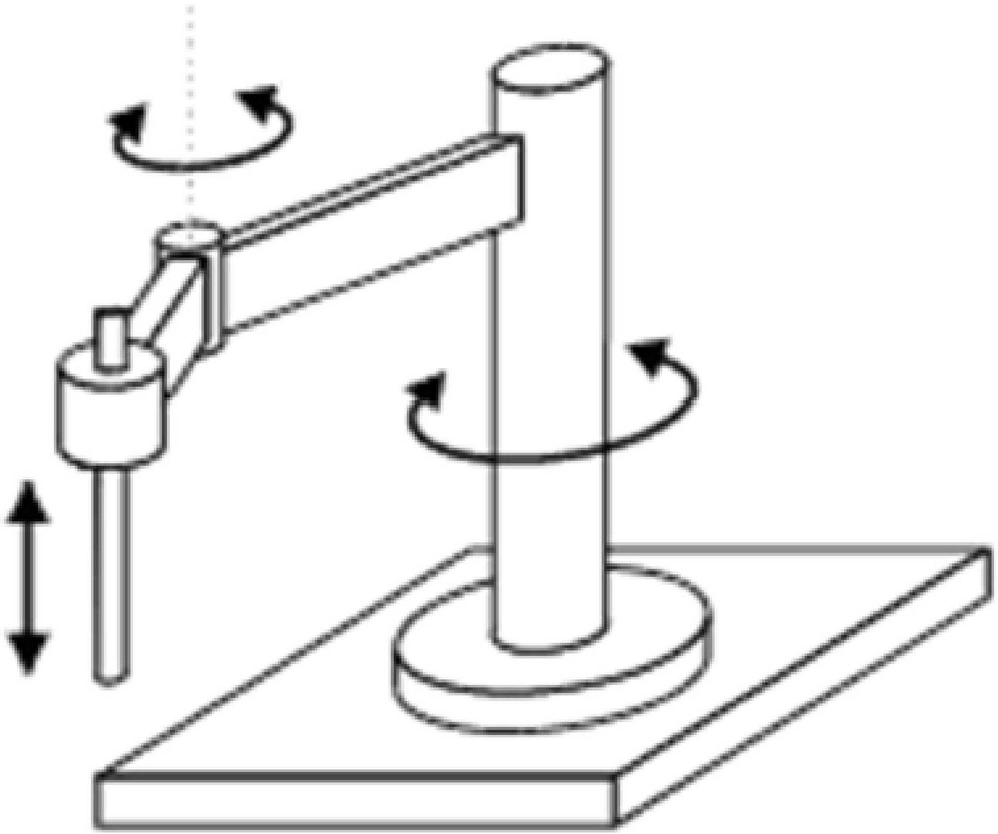
MCD mediante Denavit-Hatenverg.
Algoritmo con ejemplos

- **DH 15**.- Obtener la matriz de transformación entre la base y el extremo del robot $T_n = {}^{0}A_1\ {}^{1}A_{2\ldots}{}^{n-1}A_n$ POSMULTIPLICAR
- **DH 16**.- La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares

$$\mathbf{T} = {}^{0}\mathbf{A}_1\,{}^{1}\mathbf{A}_2\,{}^{2}\mathbf{A}_3\,{}^{3}\mathbf{A}_4 = \begin{bmatrix} -S_1C_4 & S_1S_4 & C_1 & C_1(d_3+l_4) \\ C_1C_4 & -C_1S_4 & S_1 & S_1(d_3+l_4) \\ S_4 & C_4 & 0 & d_2+l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

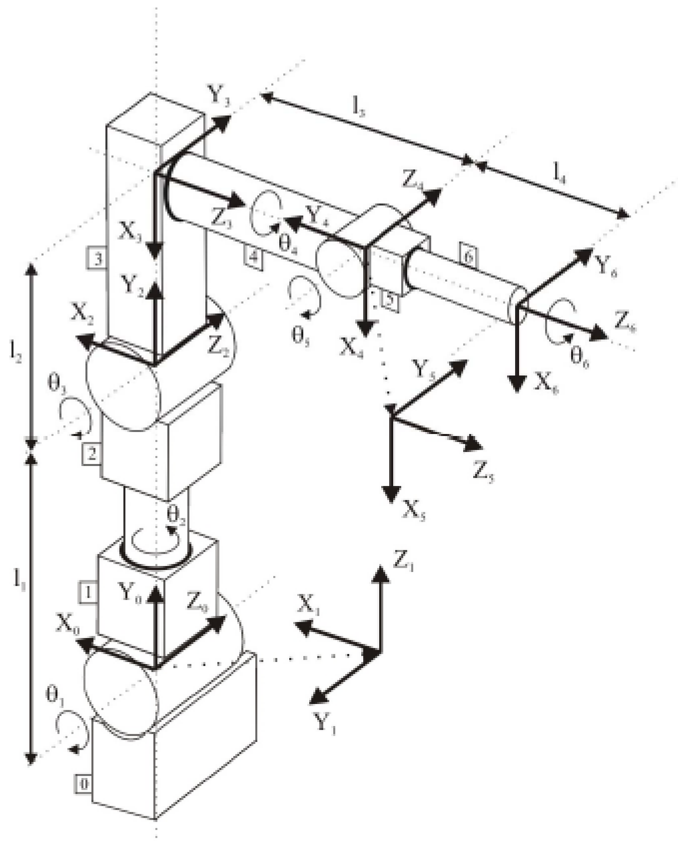
MCD mediante Denavit-Hatenverg.

Robot SCARA



MCD mediante Denavit-Hatenverg.

ABB IRB 6400C



Articulación	θ	d	a	α
1	θ_1	0	0	-90
2	θ_2	l_1	0	90
3	θ_3-90	0	$-l_2$	90
4	θ_4	l_3	0	-90
5	θ_5	0	0	90
6	θ_6	l_4	0	0

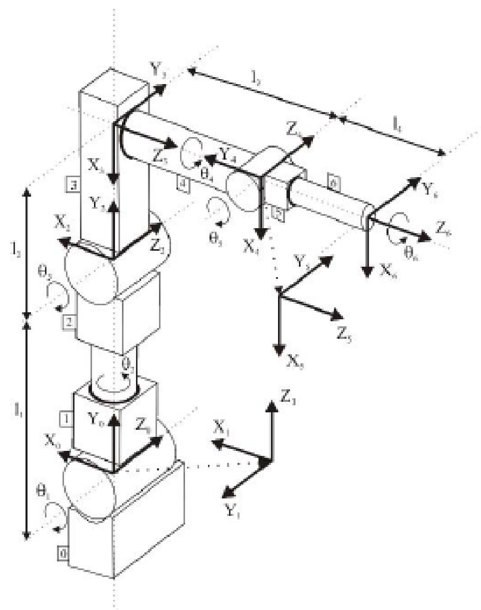
$$^0\mathbf{A}_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^1\mathbf{A}_2 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^2\mathbf{A}_3 = \begin{bmatrix} S_3 & 0 & -C_3 & -l_2S_3 \\ -C_3 & 0 & -S_3 & l_2C_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^3\mathbf{A}_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^4\mathbf{A}_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^5\mathbf{A}_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

MCD mediante Denavit-Hatenverg.
ABB IRB 6400C

$$\mathbf{T}={}^0\mathbf{A}_1\,{}^1\mathbf{A}_2\,{}^2\mathbf{A}_3\,{}^3\mathbf{A}_4\,{}^4\mathbf{A}_5\,{}^5\mathbf{A}_6=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$n_x = (C_1C_2S_3+S_1C_3)(C_4C_5C_6-S_4S_6)+C_1S_2(S_4C_5C_6+C_4S_6)+(-C_1C_2C_3+S_1S_3)S_5C_6$$

$$n_y = (-S_1C_2S_3+S_1C_3)(C_4C_5C_6-S_4S_6)+S_1S_2(S_4C_5C_6+C_4S_6)+(-S_1C_2C_3-C_1S_3)S_5C_6$$

$$n_z = (-S_2S_3)(C_4C_5C_6-S_4S_6)+C_2(S_4C_5C_6+C_4S_6)+S_2C_3S_5C_6$$

$$o_x = (C_1C_2S_3+S_1C_3)(-C_4C_5C_6-S_4S_6)+C_1S_2(-S_4C_5C_6+C_4S_6)+(-C_1C_2C_3+S_1S_3)(-S_5C_6)$$

$$o_y = (-S_1C_2S_3+S_1C_3)(-C_4C_5C_6-S_4S_6)+S_1S_2(-S_4C_5C_6+C_4S_6)+(-S_1C_2C_3-C_1S_3)(-S_5C_6)$$

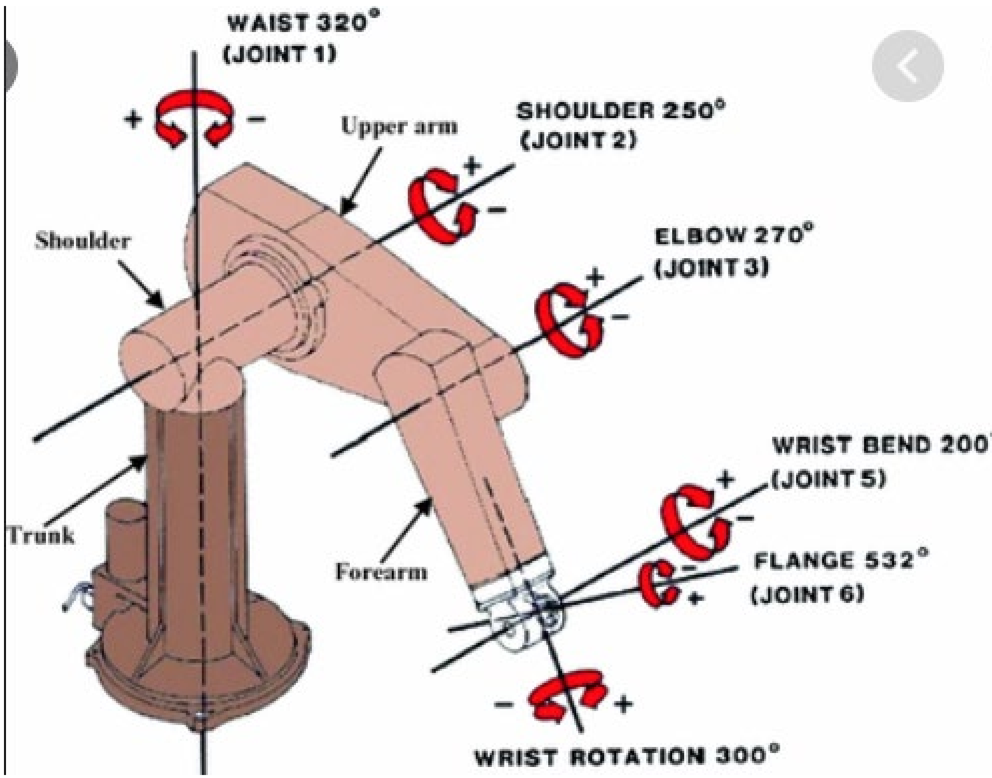
$$o_z = (-S_2S_3)(-C_4C_5C_6-S_4S_6)+C_2(-S_4C_5C_6+C_4S_6)+S_2C_3(-S_5C_6)$$

$$p_x = (C_1C_2S_3+S_1C_3)(l_4C_4S_5)+C_1S_2(l_4S_4S_5)+(C_1C_2C_3+S_1S_3)(-l_4C_5+l_3)+ \\ (-l_2C_1C_2S_3-l_2S_1C_3-l_1S_1)$$

$$p_y = (-S_1C_2S_3-C_1C_3)(l_4C_4S_5)+S_1S_2(l_4S_4S_5)+(-C_1C_2C_3-C_1S_3)(-l_4C_5+l_3)+ \\ (-l_2S_1C_2S_3-l_2C_1C_3+l_1C_1)$$

$$p_z = (-S_2S_3)(l_4C_4C_5)+C_2(l_4S_4S_5)+S_2C_3(-l_4C_5+l_3)+l_2S_2S_3$$

MCD mediante Denavit-Hatenverg.
Robot puma 560



MCD mediante cuaternios

- Idéntico planteamiento al de las MTH, pero utilizando cuaternios y vectores para expresar las transformaciones:

Resultado de trasladar según p_i y rotar según Q_i

$$(0,\mathbf{a}_{i-1}) = Q_i(0,\mathbf{a}_i)Q_i^{*} + (0,\mathbf{p}_i)$$

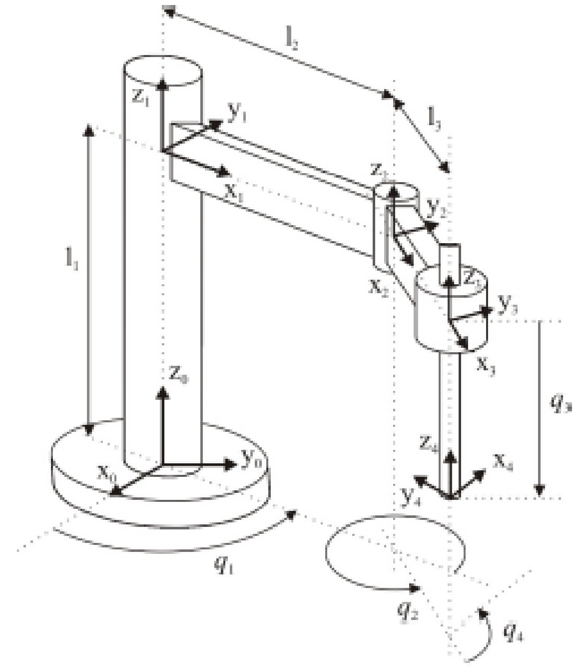
$$\mathbf{R}_{i-1} = Q_i\mathbf{R}_i$$

1. Seleccionar sistemas de coordenadas
2. Encontrar los cambios de base entre sistemas
3. Componer los cambios de base parciales
4. El cambio de base global recoge el MCD

MCD mediante cuaternios

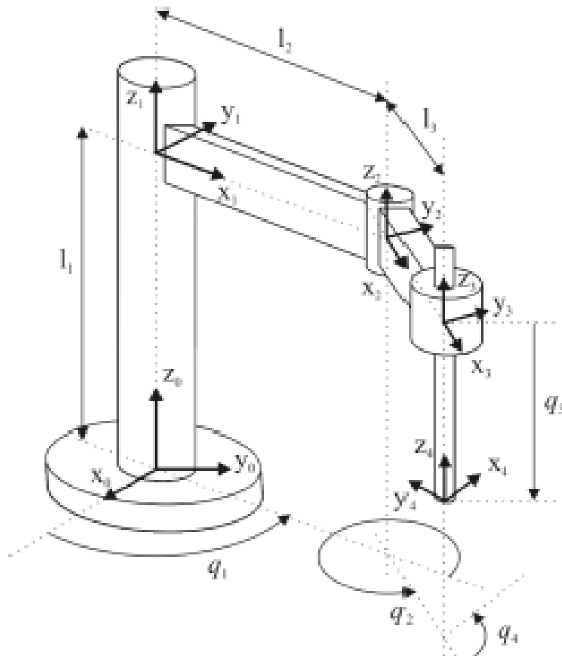
Para obtener la relación entre {S_0} y {S_4} se irá convirtiendo sucesivamente {S_0} en {S_1}, {S_2}, {S_3} y {S_4} según la siguiente serie de transformaciones

1. Desplazamiento de {S_0} una distancia l_1 a lo largo del eje $\mathbf{z}_0$ y giro un ángulo q_1 alrededor del eje $\mathbf{z}_0$, llegándose a {S_1}.
2. Desplazamiento de {S_1} una distancia l_2 a lo largo del eje $\mathbf{x}_1$ y giro un ángulo q_2 alrededor del nuevo eje $\mathbf{z}$, para llegar al sistema {S_2}.
3. Desplazamiento a lo largo del eje $\mathbf{x}_2$ una distancia l_3 para llegar al sistema {S_3}.
4. Desplazamiento de {S_3} una distancia q_3 a lo largo del eje $\mathbf{z}_3$ y giro en torno a $\mathbf{z}_4$ de un ángulo q_4, llegándose finalmente a {S_4}.



$$S_0 \rightarrow S_1: \quad \mathbf{T}(\mathbf{z}, l_1) \quad \mathbf{Rot}(\mathbf{z}, q_1)$$
$$S_1 \rightarrow S_2: \quad \mathbf{T}(\mathbf{x}, l_2) \quad \mathbf{Rot}(\mathbf{z}, q_2)$$
$$S_2 \rightarrow S_3: \quad \mathbf{T}(\mathbf{x}, l_3) \quad \mathbf{Rot}(\mathbf{z}, 0)$$
$$S_3 \rightarrow S_4: \quad \mathbf{T}(\mathbf{z}, -q_3) \quad \mathbf{Rot}(\mathbf{z}, q_4)$$

MCD mediante cuaternios



$S_0 \rightarrow S_1$: $\mathbf{T}(\mathbf{z}, l_1)$ $\mathbf{Rot}(\mathbf{z}, q_1)$ $\mathbf{p}_1 = (0,0,l_1)$ $Q_1 = (\hat{C}_1, 0, 0, \hat{S}_1)$

$S_1 \rightarrow S_2$: $\mathbf{T}(\mathbf{x}, l_2)$ $\mathbf{Rot}(\mathbf{z}, q_2)$ $\mathbf{p}_2 = (l_2, 0, 0)$ $Q_2 = (\hat{C}_2, 0, 0, \hat{S}_2)$ con $\hat{C}_i = \cos\left(\frac{q_i}{2}\right)$

$S_2 \rightarrow S_3$: $\mathbf{T}(\mathbf{x}, l_3)$ $\mathbf{Rot}(\mathbf{z}, 0)$ $\mathbf{p}_3 = (l_3, 0, 0)$ $Q_3 = (1,0,0,0)$ $\hat{S}_i = \operatorname{sen}\left(\frac{q_i}{2}\right)$

$S_3 \rightarrow S_4$: $\mathbf{T}(\mathbf{z}, -q_3)$ $\mathbf{Rot}(\mathbf{z}, q_4)$ $\mathbf{p}_4 = (0,0,-q_3)$ $Q_4 = (\hat{C}_4, 0, 0, \hat{S}_4)$

MCD mediante cuaternios

Operando con cuaternios se tiene

$$(0,\mathbf{a}_{i-1}) = Q_i(0,\mathbf{a}_i)Q_i^* + (0,\mathbf{p}_i)$$
$$\mathbf{R}_{i-1} = Q_i\mathbf{R}_i$$

$$(0,\mathbf{a}_0) = Q_1(0,\mathbf{a}_1)Q_1^* + (0,\mathbf{p}_1)$$
$$\mathbf{R}_0 = Q_1\mathbf{R}_1$$
$$(0,\mathbf{a}_1) = Q_2(0,\mathbf{a}_2)Q_2^* + (0,\mathbf{p}_2)$$
$$\mathbf{R}_1 = Q_2\mathbf{R}_2$$
$$(0,\mathbf{a}_2) = Q_3(0,\mathbf{a}_3)Q_3^* + (0,\mathbf{p}_3)$$
$$\mathbf{R}_3 = Q_3\mathbf{R}_3$$
$$(0,\mathbf{a}_3) = Q_4(0,\mathbf{a}_4)Q_4^* + (0,\mathbf{p}_4)$$
$$\mathbf{R}_3 = Q_4\mathbf{R}_4$$

$$\begin{aligned}(0,\mathbf{a}_0) &= Q_1\Big[Q_2\Big[Q_3\big[Q_4(0,\mathbf{a}_4)Q_4^* + (0,\mathbf{p}_4)\big]Q_3^* + (0,\mathbf{p}_3)\Big]Q_2^* + (0,\mathbf{p}_2)\Big]Q_1^* + (0,\mathbf{p}_1) = \\ &= Q_1Q_2Q_3Q_4(0,\mathbf{a}_4)Q_4^*Q_3^*Q_2^*Q_1^* + Q_1Q_2Q_3(0,\mathbf{p}_4)Q_3^*Q_2^*Q_1^* + \\ &\quad + Q_1Q_2(0,\mathbf{p}_3)Q_2^*Q_1^* + Q_1(0,\mathbf{p}_2)Q_1^* + (0,\mathbf{p}_1) = \\ &= Q_{1234}(0,\mathbf{a}_4)Q_{1234}^* + Q_{123}(0,\mathbf{p}_4)Q_{123}^* + Q_{12}(0,\mathbf{p}_3)Q_{12}^* + Q_1(0,\mathbf{p}_2)Q_1^* + (0,\mathbf{p}_1)\end{aligned}$$

$$\boxed{\begin{aligned}(0,\mathbf{a}_0) = (0,\, & a_{4x}\hat{C}_{112244} - a_{4y}\hat{S}_{112244} + l_3\hat{C}_{1122} + l_2\hat{C}_{11}, \\ & a_{4y}\hat{C}_{112244} - a_{4x}\hat{S}_{112244} + l_3\hat{S}_{1122} + l_2\hat{S}_{11}, a_{4z} - q_3 + l_1)\end{aligned}}$$

$$\boxed{\mathbf{R}_0 = Q_1Q_2Q_3Q_4\mathbf{R}_4 = Q_{1234}\mathbf{R}_4 = (\hat{C}_{124},0,0,\hat{S}_{124})\,\mathbf{R}_4}$$

MCD mediante cuaternios

El extremo del robot $a_4=(0,0,0)$, se encuentra en:

$$(0,\mathbf{a}_0) = (0, a_{4x}\hat{C}_{112244} - a_{4y}\hat{S}_{112244} + l_3\hat{C}_{1122} + l_2\hat{C}_{11},$$
$$a_{4y}\hat{C}_{112244} - a_{4x}\hat{S}_{112244} + l_3\hat{S}_{1122} + l_2\hat{S}_{11}, a_{4z} - q_3 + l_1)$$
$$= (0, l_3\hat{C}_{1122} + l_2\hat{C}_{11}, l_3\hat{S}_{1122} + l_2\hat{S}_{11}, l_1 - q_3)$$

$$\left\{ \boxed{\begin{array}{l} x = a_{0x} = l_3\cos(q_1+q_2) + l_2\cos q_1 \\ y = a_{0y} = l_3\operatorname{sen}(q_1+q_2) + l_2\operatorname{sen} q_1 \\ z = a_{0z} = l_1 - q_3 \end{array}} \right.$$

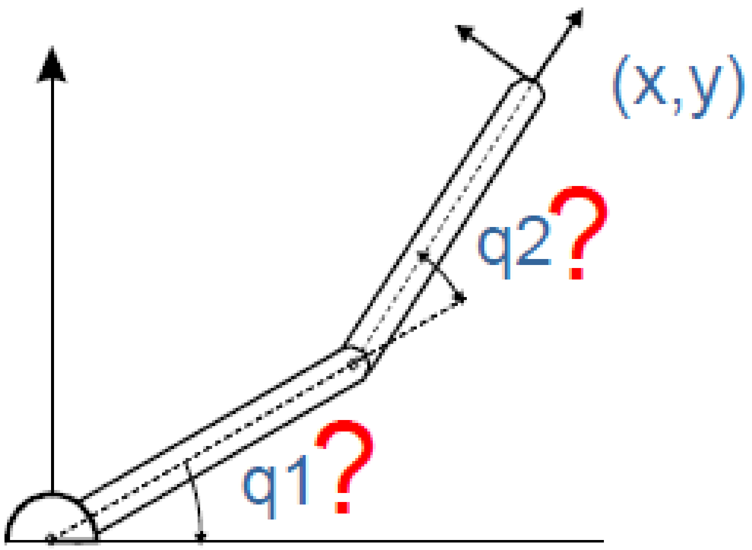
El sistema del extremo $(R_4=(1,0,0,0))$se encuentra girado respecto del de la base, según:

$$\mathbf{R}_0 = Q_1Q_2Q_3Q_4\mathbf{R}_4 = Q_{1234}\mathbf{R}_4 = (\hat{C}_{124}, 0, 0, \hat{S}_{124})\,\mathbf{R}_4$$

$$\boxed{\mathbf{Rotz}(q_1+q_2+q_4)}$$

Modelo cinemático inverso

- **Objetivo**: encontrar los valores que deben adoptar las coordenadas articulares del robot (q1, …, qn) para que su extremo se posicione y oriente según una determinada localización espacial



Modelo cinemático inverso
Métodos

- **Soluciones cerradas** $\boldsymbol{q = f(x, y, z, \phi, \theta, \psi)}$
  - Garantiza una solución en un tiempo definido
  - En caso de solución múltiple, permite seleccionar la más adecuada
  - No siempre existe
  - Métodos:
    - Basados en consideraciones geométricas
    - Manipulación de las ecuaciones del MCD
- **Métodos iterativos**
  - Algoritmo generalizable para cualquier robot
  - Problemas de convergencia

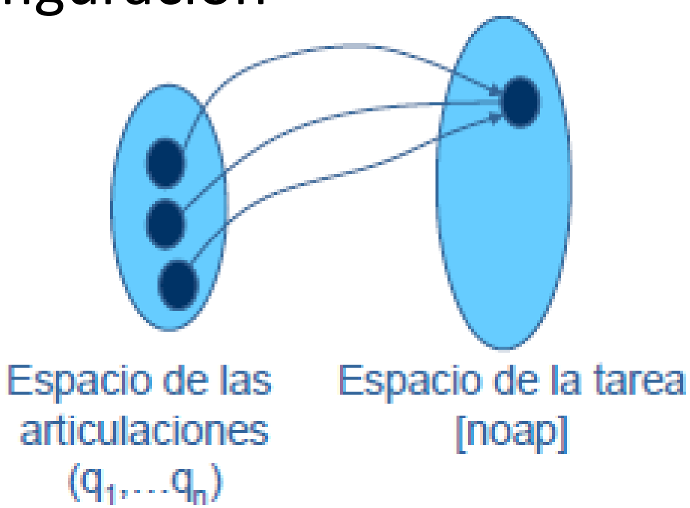
Modelo cinemático inverso
Problemas

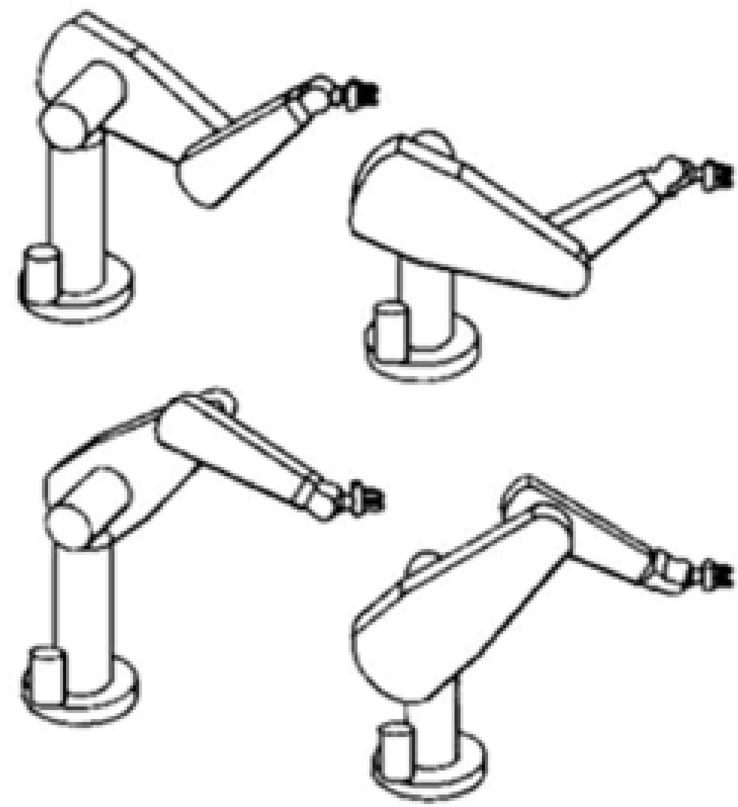
- No siempre existe una solución cerrada
- Posible solución múltiple
- Dificultad analítica si se busca una solución cerrada
- Problemas de convergencia y mínimos locales si se usa un método iterativo

Modelo cinemático inverso

Problemas

- Soluciones múltiples:
- Unas coordenadas en el espacio de la tarea, pueden corresponder a varias coordenadas articulares
- Se deben evitar cambios bruscos de configuración



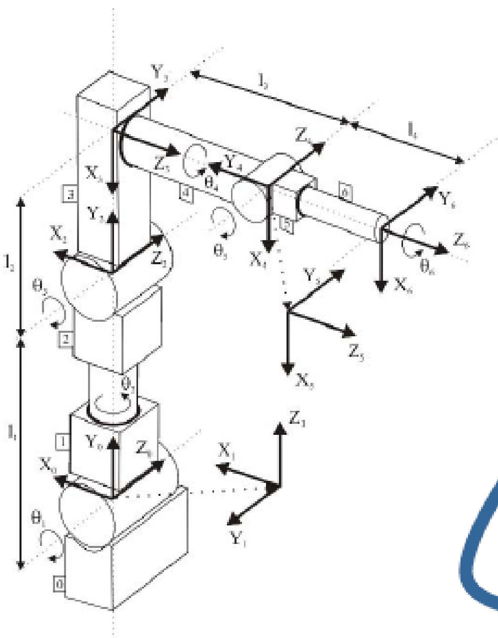


Modelo cinemático inverso

Problemas

Dificultad analítica:

Directo:
q1,..q6 datos
$n_x, \ldots p_x$ incógnitas



$$
\begin{aligned}
n_x &= (C_1C_2S_3 + S_1C_3)(C_4C_5C_6 - S_4S_6) + C_1S_2(S_4C_5C_6 + C_4S_6) + (-C_1C_2C_3 + S_1S_3)S_5C_6 \\
n_y &= (-S_1C_2S_3 + S_1C_3)(C_4C_5C_6 - S_4S_6) + S_1S_2(S_4C_5C_6 + C_4S_6) + (-S_1C_2C_3 - C_1S_3)S_5C_6 \\
n_z &= (-S_2S_3)(C_4C_5C_6 - S_4S_6) + C_2(S_4C_5C_6 + C_4S_6) + S_2C_3S_5C_6 \\
o_x &= (C_1C_2S_3 + S_1C_3)(-C_4C_5C_6 - S_4S_6) + C_1S_2(-S_4C_5C_6 + C_4S_6) + (-C_1C_2C_3 + S_1S_3)(-S_5C_6) \\
o_y &= (-S_1C_2S_3 + S_1C_3)(-C_4C_5C_6 - S_4S_6) + S_1S_2(-S_4C_5C_6 + C_4S_6) + (-S_1C_2C_3 - C_1S_3)(-S_5C_6) \\
o_z &= (-S_2S_3)(-C_4C_5C_6 - S_4S_6) + C_2(-S_4C_5C_6 + C_4S_6) + S_2C_3(-S_5C_6) \\
p_x &= (C_1C_2S_3 + S_1C_3)(l_4C_4S_5) + C_1S_2(l_4S_4S_5) + (C_1C_2C_3 + S_1S_3)(-l_4C_5 + l_3) + \\
&\qquad (-l_2C_1C_2S_3 - l_2S_1C_3 - l_1S_1) \\
p_y &= (-S_1C_2S_3 - C_1C_3)(l_4C_4S_5) + S_1S_2(l_4S_4S_5) + (-C_1C_2C_3 - C_1S_3)(-l_4C_5 + l_3) + \\
&\qquad (-l_2S_1C_2S_3 - l_2C_1C_3 + l_1C_1) \\
p_z &= (-S_2S_3)(l_4C_4C_5) + C_2(l_4S_4S_5) + S_2C_3(-l_4C_5 + l_3) + l_2S_2S_3
\end{aligned}
$$

Inverso:
q1,..q6 incógnitas
nx,…px datos

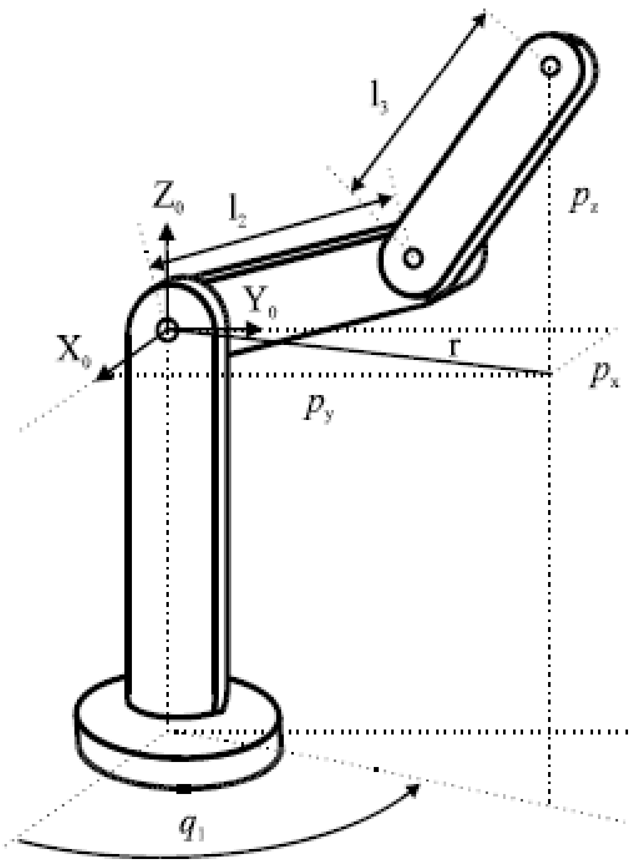
¿Cómo despejar las incógnitas?

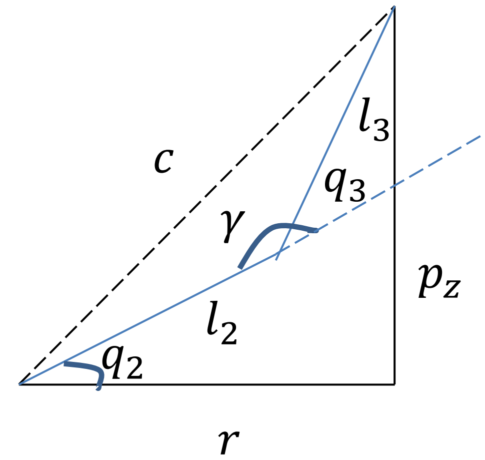
Modelo cinemático inverso
Resolución mediante métodos geométricos

- Aplicable a robots con poco grados de libertad.
- Busca relaciones geométricas (típicamente trigonométricas) entre las coordenadas en el espacio de la tarea y el de las articulaciones.

Modelo cinemático inverso

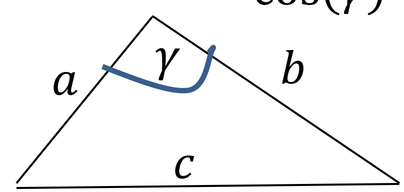
Resolución mediante métodos geométricos





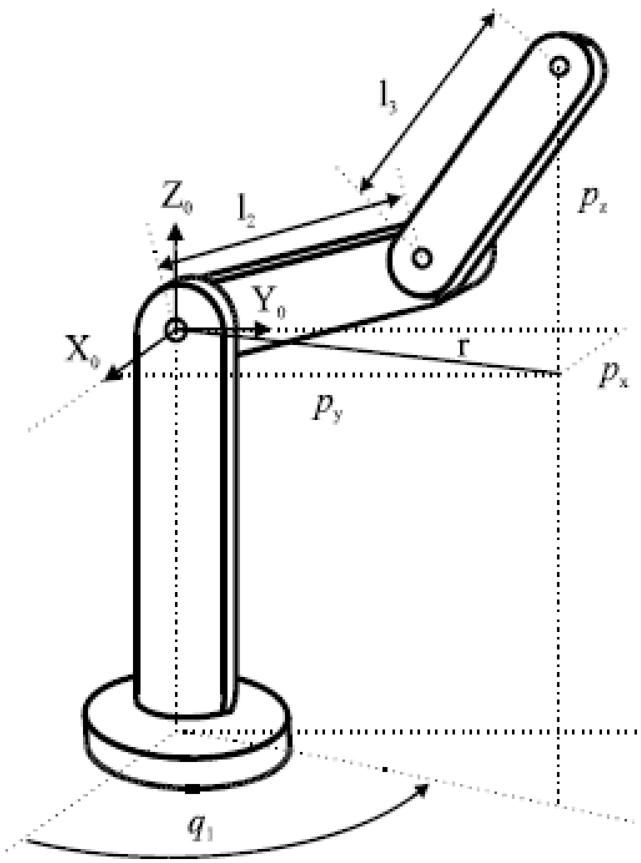
$$r^2 + p_z^2 = l_2^2 + l_3^2 - 2 l_2 l_3 \cos(\gamma)$$

$$\cos(\gamma) = -cos(q_3)$$



Teorema del coseno $\quad c^2 = a^2 + b^2 - 2abcos(\gamma)$

Modelo cinemático inverso
Resolución mediante métodos geométricos



$$\boxed{q_1 = \operatorname{arctg}\left(\frac{p_y}{p_x}\right)}$$

$$\left.\begin{array}{l} r^2 = p_x^2 + p_y^2 \\ r^2 + p_z^2 = l_2^2 + l_3^2 + 2 l_2 l_3 \cos q_3 \end{array}\right\} \Rightarrow$$

$$\cos q_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

$$\operatorname{sen} q_3 = \pm\sqrt{1 - \cos^2 q_3}$$

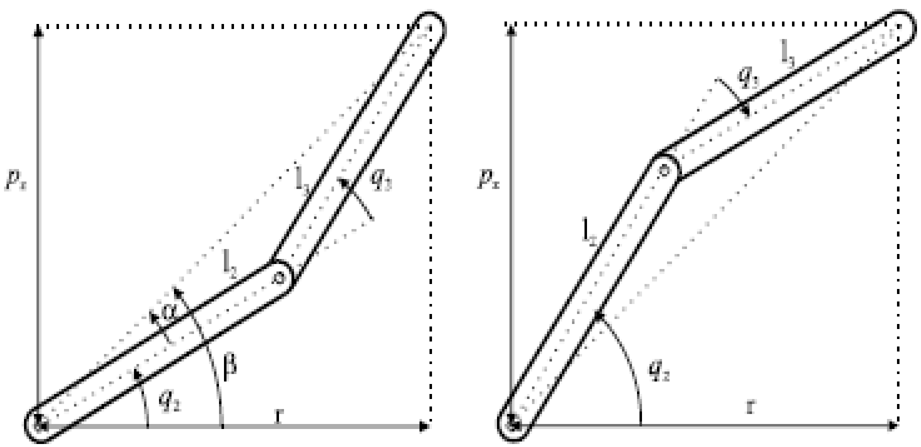
Solución Doble

$$q_3 = \operatorname{arctg}\left(\frac{\pm\sqrt{1-\cos^2 q_3}}{\cos q_3}\right)$$

$$\text{con} \quad \cos q_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

Modelo cinemático inverso

Resolución mediante métodos geométricos

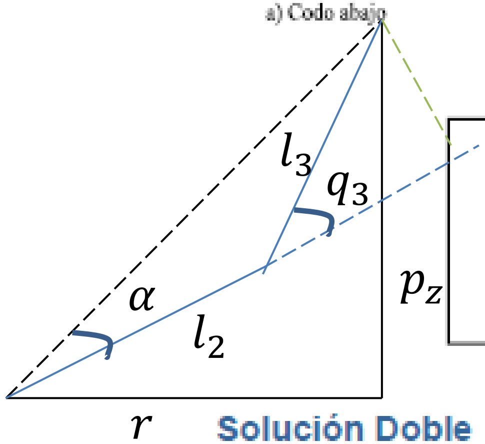


a) Codo abajo   b) Codo arriba

$$q_2 = \beta - \alpha$$

$$\beta = \operatorname{arctg}\left(\frac{p_z}{r}\right) = \operatorname{arctg}\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right)$$

$$\alpha = \operatorname{arctg}\left(\frac{l_3 \operatorname{sen} q_3}{l_2 + l_3 \cos q_3}\right)$$



$$q_2 = \operatorname{arctg}\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \operatorname{arctg}\left(\frac{l_3 \operatorname{sen} q_3}{l_2 + l_3 \cos q_3}\right)$$

Solución Doble

Modelo cinemático inverso

Resolución mediante MTH

- Se conoce la localización del robot T=[noap] (o equivalente) a donde se le quiere llevar
- Se conoce la cinemática del robot definida por sus parámetros DH y por lo tanto:

$$ {}^{0}A_{n}(q_{1},..,q_{n})={}^{0}A_{1}(q_{1}) \dots {}^{n-1}A_{n}(q_{n}) $$

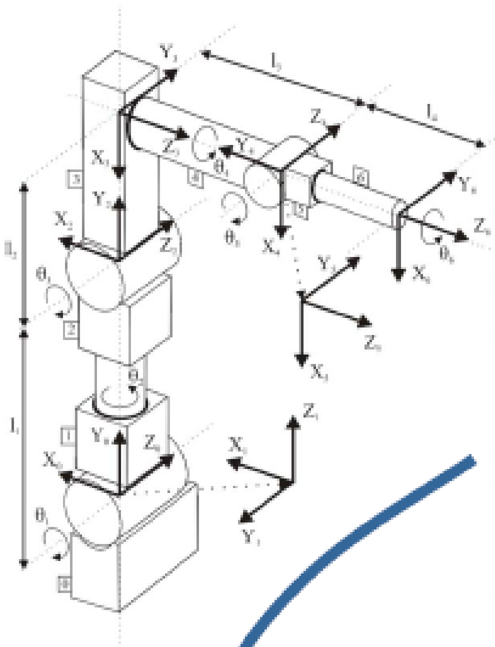
- Se trata de encontrar q1,…,qn que satisfagan:

$$ {}^{0}A_{n}(q1,..,qn)=[noap] $$

- Esta ecuación corresponde a 12 ecuaciones no lineales con n incógnitas

Modelo cinemático inverso

Resolución mediante MTH



$$n_x = (C_1C_2S_3 + S_1C_3)(C_4C_5C_6 - S_4S_6) + C_1S_2(S_4C_5C_6 + C_4S_6) + (-C_1C_2C_3 + S_1S_3)S_5C_6$$

$$n_y = (-S_1C_2S_3 + S_1C_3)(C_4C_5C_6 - S_4S_6) + S_1S_2(S_4C_5C_6 + C_4S_6) + (-S_1C_2C_3 - C_1S_3)S_5C_6$$

$$n_z = (-S_2S_3)(C_4C_5C_6 - S_4S_6) + C_2(S_4C_5C_6 + C_4S_6) + S_2C_3S_5C_6$$

$$o_x = (C_1C_2S_3 + S_1C_3)(-C_4C_5C_6 - S_4S_6) + C_1S_2(-S_4C_5C_6 + C_4S_6) + (-C_1C_2C_3 + S_1S_3)(-S_5C_6)$$

$$o_y = (-S_1C_2S_3 + S_1C_3)(-C_4C_5C_6 - S_4S_6) + S_1S_2(-S_4C_5C_6 + C_4S_6) + (-S_1C_2C_3 - C_1S_3)(-S_5C_6)$$

$$o_z = (-S_2S_3)(-C_4C_5C_6 - S_4S_6) + C_2(-S_4C_5C_6 + C_4S_6) + S_2C_3(-S_5C_6)$$

$$p_x = (C_1C_2S_3 + S_1C_3)(l_4C_4S_5) + C_1S_2(l_4S_4S_5) + (C_1C_2C_3 + S_1S_3)(-l_4C_5 + l_3) + (-l_2C_1C_2S_3 - l_2S_1C_3 - l_1S_1)$$

$$p_y = (-S_1C_2S_3 - C_1C_3)(l_4C_4S_5) + S_1S_2(l_4S_4S_5) + (-C_1C_2C_3 - C_1S_3)(-l_4C_5 + l_3) + (-l_2S_1C_2S_3 - l_2C_1C_3 + l_1C_1)$$

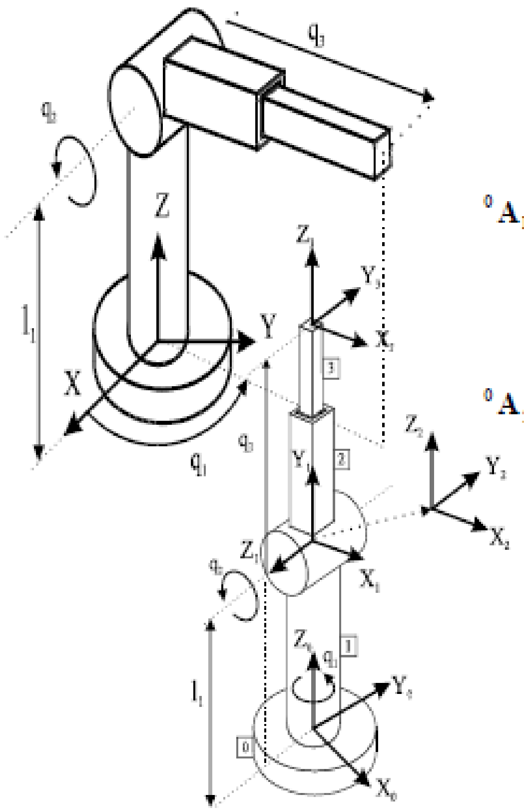
$$p_z = (-S_2S_3)(l_4C_4C_5) + C_2(l_4S_4S_5) + S_2C_3(-l_4C_5 + l_3) + l_2S_2S_3$$

q1,..q6 incógnitas

nx,…px datos

Modelo cinemático inverso

Resolución mediante MTH



Artic.	θ	d	a	α
1	q_1	l_1	0	90°
2	q_2	0	0	-90°
3	0	q_3	0	0

$$^0\mathbf{A}_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^1\mathbf{A}_2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^2\mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0\mathbf{A}_2 = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & 0 \\ S_1C_2 & C_1 & -S_1S_2 & 0 \\ S_2 & 0 & C_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} = {}^0\mathbf{A}_3 = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -q_3C_1S_2 \\ S_1C_2 & C_1 & -S_1S_2 & -q_3S_1S_2 \\ S_2 & 0 & C_2 & q_3C_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Obtener el MCD

Modelo cinemático inverso
Resolución mediante MTH

2. Aislar q_1

$$\mathbf{T} = {}^0\mathbf{A}_1 \, {}^1\mathbf{A}_2 \, {}^2\mathbf{A}_3$$

$$\left({}^0\mathbf{A}_1\right)^{-1} \mathbf{T} = {}^1\mathbf{A}_2 \, {}^2\mathbf{A}_3$$

$$\underbrace{\left({}^0\mathbf{A}_1\right)^{-1}}_{({}^0A_1)^{-1}} \, \underbrace{{}^0\mathbf{T}_3}_{{}^0T_3} = {}^1\mathbf{A}_2 \, {}^2\mathbf{A}_3 = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ \boxed{S_1 \quad -C_1 \quad 0 \quad 0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \underbrace{\begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^1A_2} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^2A_3} = \underbrace{\begin{bmatrix} C_2 & 0 & -S_2 & -S_2 q_3 \\ S_2 & 0 & C_2 & C_2 q_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^1A_2 \, {}^2A_3}$$

Elemento 3,4

$$S_1 p_x - C_1 p_y = 0 \quad \Rightarrow \tan(q_1) = \left(\frac{p_y}{p_x}\right) \quad \Rightarrow q_1 = \arctan\left(\frac{p_y}{p_x}\right)$$

Modelo cinemático inverso

Resolución mediante MTH

3. Continuar aislando el resto de las incógnitas

$$\mathbf{T} = {}^0\mathbf{A}_1\,{}^1\mathbf{A}_2\,{}^2\mathbf{A}_3$$

$$\left({}^0\mathbf{A}_1\right)^{-1}\mathbf{T} = {}^1\mathbf{A}_2\,{}^2\mathbf{A}_3$$

$$\left({}^1\mathbf{A}_2\right)^{-1}\left({}^0\mathbf{A}_1\right)^{-1}\mathbf{T} = {}^2\mathbf{A}_3$$

$$\begin{bmatrix} C_2 & S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} C_2C_1 & C_2S_1 & S_2 & -l_1S_2 \\ -S_1 & C_1 & 0 & 0 \\ -S_2C_1 & -S_2S_1 & C_2 & -C_2l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \boxed{0} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \boxed{q_3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Elementos 1,4 y 3,4

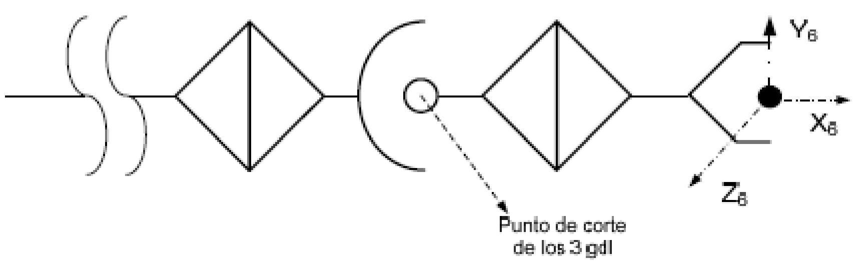
$$q_2 = \arctan\frac{\sqrt{p_x^2 + p_y^2}}{l_1 - p_z}$$

$$q_3 = C_2(p_z - l_1) - S_2\sqrt{p_x^2 + p_y^2}$$

UNIVERSIDAD DE BURGOS

Modelo cinemático inverso
Resolución mediante desacoplo cinemático

- Los procedimientos anteriores son complejos, o inutilizables, cuando el número de grados de libertad es elevado
- El desacoplo cinemático divide un problema de 6 gdl a 2 problemas de 3gdl
- **OJO! Solo puede aplicarse cuando los ejes de las 3 últimas articulaciones se cortan en un punto**



UNIVERSIDAD DE BURGOS

Modelo cinemático inverso
Resolución mediante desacoplo cinemático

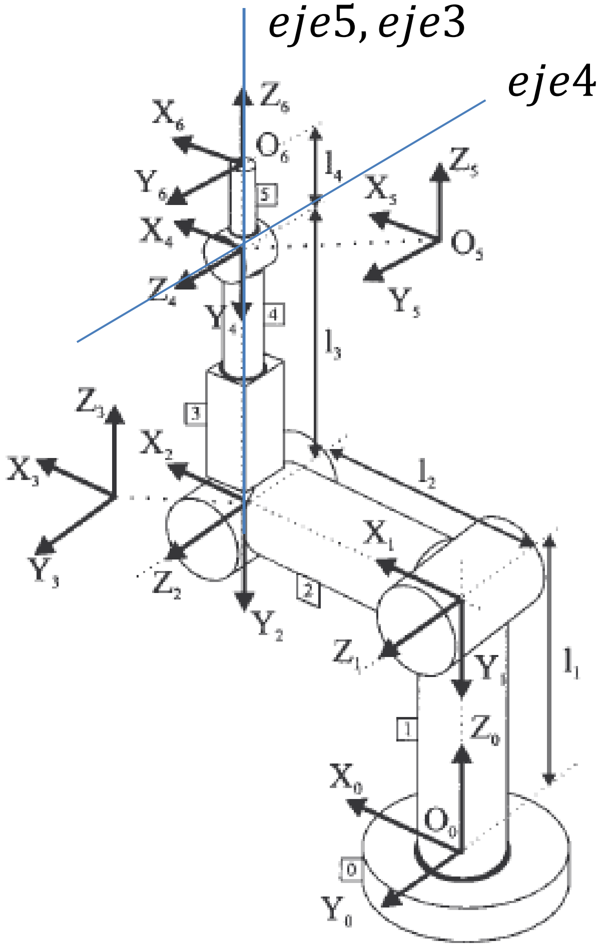
- A partir de a posición y orientación deseadas [noap], se obtiene el punto de corte de los 3 últimos ejes (punto de la muñeca: p_m)
- Se resuelve el modelo cinemático inverso para el robot de 3 gdl (q_1, q_2, q_3) que va desde la base hasta p_m
- La resolución de (q_1, q_2, q_3), condiciona la posición y orientación del robot en p_m
- Se resuelve el problema cinemático inverso para el robot que va desde pm hasta el punto final p_r, encontrando (q_4, q_5, q_6).

Modelo cinemático inverso
Resolución mediante desacoplo cinemático

- El punto p_m corresponde al centro del sistema O_5
- El punto final del robot p_r corresponde al centro del sistema O_6
- Como la dirección del eje Z_6 coincide con Z_5 , y la distancia entre O_5 y O_6 es $d_4, p_m = p_r - l_4\, Z_6$
- **Se calculan los valores (q_1, q_2, q_3), a partir de un método geométrico**
- Si ${}^0R_6$ es la matriz de rotación de ${}^0T_6$, entonces

$$ {}^0R_6 = [noa] = {}^0R_3 \ {}^3R_6 $$

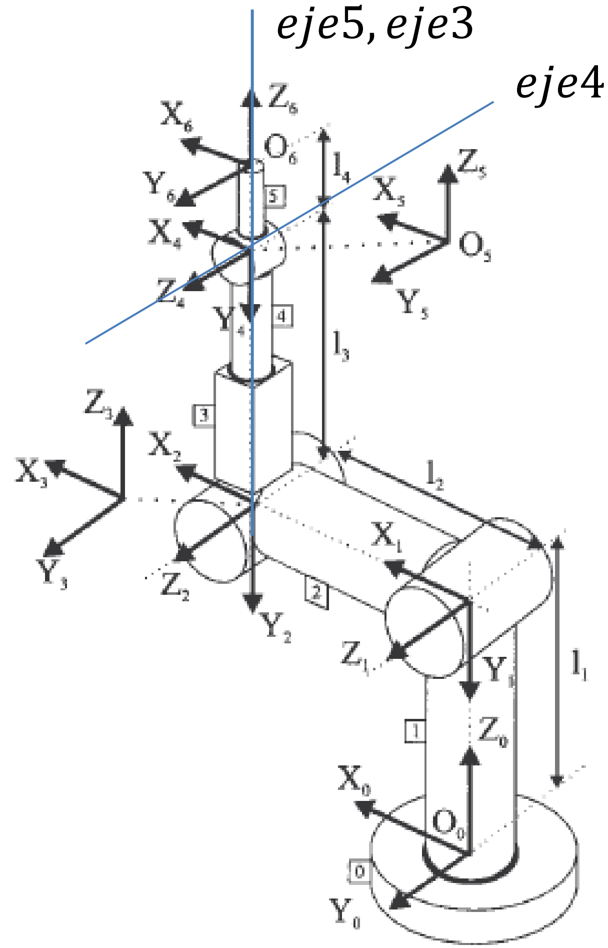
$$ {}^3R_6 = ({}^0R_3)^{-1} \, [noa] = [r_{ij}] $$



Modelo cinemático inverso
Resolución mediante desacoplo cinemático

$$
{}^{3}\mathbf{R}_4 = \begin{bmatrix} C_4 & 0 & -S_4 \\ S_4 & 0 & C_4 \\ 0 & -1 & 0 \end{bmatrix} \quad {}^{4}\mathbf{R}_5 = \begin{bmatrix} C_5 & 0 & S_5 \\ S_5 & 0 & -C_5 \\ 0 & 1 & 0 \end{bmatrix} \quad {}^{5}\mathbf{R}_6 = \begin{bmatrix} C_6 & -S_6 & 0 \\ S_6 & C_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
\left[r_{ij}\right] = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & \boxed{C_4S_5} \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & \boxed{-S_4C_5} \\ \boxed{-S_5C_6} & \boxed{S_5S_6} & \boxed{C_5} \end{bmatrix}
$$



- A partir de r13,r23,r31,r32,r33 se obtienen las ecuaciones

$$
r_{13} = C_4S_5 \qquad r_{23} = -S_4C_5 \qquad r_{33} = C_5
$$
$$
r_{31} = -S_5C_6 \qquad r_{32} = S_5S_6
$$

- A partir de ellas los ángulos

$$
\boxed{\begin{aligned} q_4 &= \operatorname{arcsen}\left(\frac{r_{23}}{r_{33}}\right) \\ q_5 &= \operatorname{arcos}\left(r_{33}\right) \\ q_6 &= \arctan\left(-\frac{r_{32}}{r_{31}}\right) \end{aligned}}
$$

Modelo cinemático inverso

Resolución mediante métodos numéricos

- Se modifican las coordenadas articulares hasta que la cinemática directa $\boldsymbol{\mathcal{K}}(\boldsymbol{q})$ se ajusta a la pose deseada $\boldsymbol{\zeta}$
- Se trata de un problema de optimización multivariable, en el que se busca minimizar el error:

$$q^* = \underset{q}{argMIN} \|\mathcal{K}(q) - \zeta\|$$

- La solución encontrada dependerá del valor inicial $(q_1, q_2, \ldots q_N)$ y del algoritmo de optimización empleado
- Algunos algoritmos de optimización pueden detenerse en algún mínimo local y no encontrar el mínimo absoluto

Modelo cinemático inverso
Resolución mediante métodos numéricos

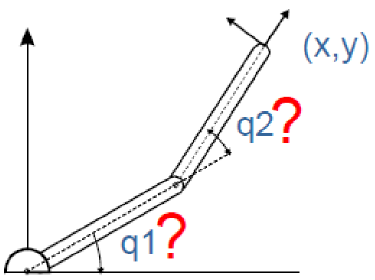
Tipos de algoritmos de optimización:

- Algoritmos clásicos
  - Directos: solo usan la función objetivo: random, simplex, Powel…
  - Indirectos: usan función objetivo y sus derivadas: gradiente, newton..
- Algoritmos de búsqueda
  - Voraz, A*, Hill-Climbing…
  - Minimax, Poda alfa-Beta
  - Algoritmos genéticos
- Ejemplos: Nelder-Mead Simplex Method (*fminsearch*)
- Ejemplos: BFGS Quasi-Newton (*fminunc*)
- Ejemplos: Trust Region Algorithm (*fminunc*)

Modelo cinemático inverso
Resolución mediante métodos numéricos

- Ejemplo con Matlab

Ejemplo de robot con 2 gdl



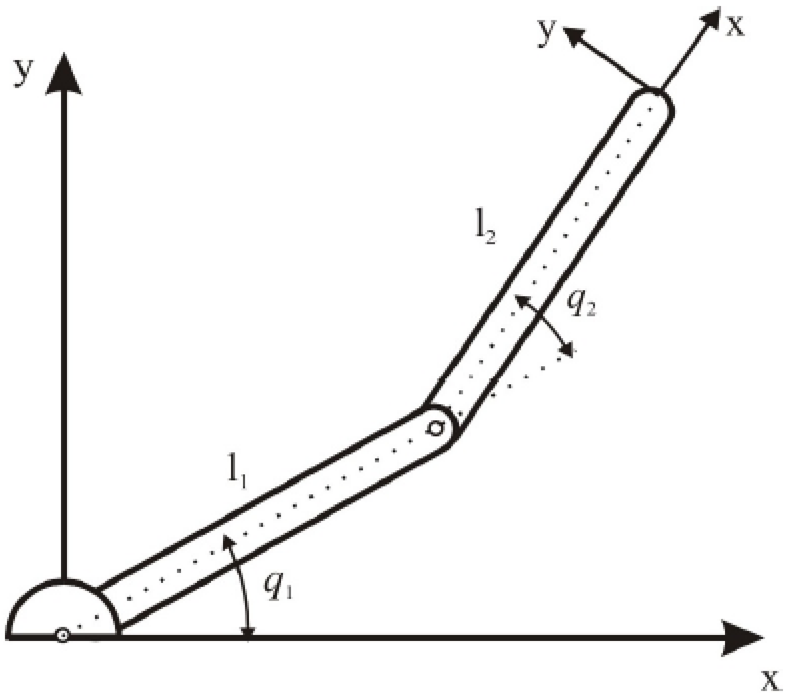

```
>>R=SerialLink([L1,L2],'name','robot')
>>posExt=[0.6;0.7]
>>q=fminsearch(@(q) norm(R.fkine(q).t-posExt), [0 0])
 q= -0.2295 2.1833
>>R.fkine(q).print
t=(0.6, 0.7), theta =112.9 deg
```

- Usa el algoritmo Nelder-Mead Simplex Method

Modelo diferencial

- Establece la relación entre las velocidades de las articulaciones, con las velocidades del extremo del robot
- Es utilizado por el sistema de control del robot para establecer qué velocidades debe imprimir a cada articulación (a través de sus respectivos actuadores) para conseguir que el extremo desarrolle una trayectoria temporal concreta.
- El modelo diferencial queda concretado en el jacobiano o matriz jacobiana.

Modelo diferencial
Jacobiano



b

$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2)$$

$$y = l_1 sen q_1 + l_2 sen(q_1 + q_2)$$

$$z = 0$$

$$p = (x, y)$$

Derivando p respecto de las coordenadas articulares q,

$$\frac{dp}{dq} = J(q)$$

$J(q)$ es el jacobiano

$$\boldsymbol{J(q)} = \begin{bmatrix} \boldsymbol{-l_1 S_1 - l_2 S_{12}} & \boldsymbol{-l_2 S_{12}} \\ \boldsymbol{l_1 C_1 + l_2 C_{12}} & \boldsymbol{l_2 C_{12}} \end{bmatrix}$$

$$dp = J(q) \cdot dq, \frac{dp}{dt} = J(q) \cdot \frac{dq}{dt}$$

$$\dot{\boldsymbol{p}} = \boldsymbol{J(q)} \cdot \dot{\boldsymbol{q}}$$

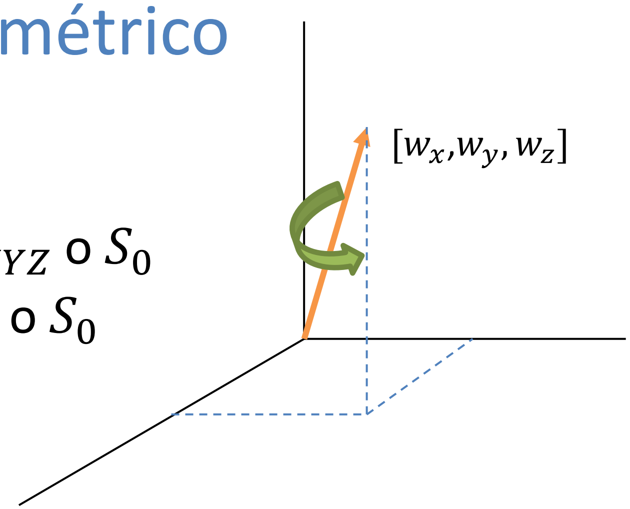
Modelo diferencial
Jacobiano

- Generalizando la expresión anterior para recoger la orientación se obtiene:

$$v_{XYZ} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{pmatrix} = J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdots \\ \dot{q}_N \end{pmatrix}$$

J(q)
- 1 columna por articulación
- 1 fila por componente espacial de velocidad
- Normalmente matriz 6xN

Jacobiano geométrico



$[v_x, v_y, v_z]$ vector de velocidad traslacional en O_{XYZ} o S_0

$[w_x, w_y, w_z]$ vector de velocidad angular en O_{XYZ} o S_0

Velocidad de giro es $\sqrt{w_x^2 + w_Y^2 + w_Z^2}$

Modelo diferencial

Jacobiano

- El jacobiano así definido proporciona la velocidad del extremo del robot respecto del sistema de coordenadas O_{XYZ} o S_0
- Para obtenerlo en el sistema de coordenadas del extremo O_{UVW} o S_N

$$v_{UVW} = \begin{pmatrix} {}^{N}R_0 & 0_{3x3} \\ 0_{3x3} & {}^{N}R_0 \end{pmatrix} J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdots \\ \dot{q}_N \end{pmatrix} = {}^{N}J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdots \\ \dot{q}_N \end{pmatrix}$$

Modelo diferencial
Jacobiano analítico

- Para algunas aplicaciones puede ser más intuitivo utilizar el jacobiano considerando las velocidades de roll, pitch y yaw $[\dot{\theta}_{r,}\dot{\theta}_{p,}\dot{\theta}_{y,}]$, $[\dot{\phi},\dot{\theta},\dot{\psi}]$ o $[\dot{r},\dot{p},\dot{y}]$ en vez del vector $\overline{w}$.

$$R(r,p,y) = R_x(r)R_y(p)R_z(y)$$

$$\frac{d}{dt}R(r,p,y) = \frac{d}{dr}R_x(r)\dot{r}R_y(p)R_z(y) + \frac{d}{dp}R_y(p)\dot{p}R_x(r)R_z(y) + \frac{d}{dy}R_z(y)\dot{y}R_x(r)R_y(p)$$

$$\frac{d}{dt}R(r,p,y) = S(w)\{R_x(r)R_y(p)R_z(y)\}$$

Resolviendo $S(w)$

$$w = B(r,p,y)\begin{pmatrix}\dot{r}\\ \dot{p}\\ \dot{y}\end{pmatrix}$$

Modelo diferencial

Jacobiano analítico

$$\frac{d}{dt}R(r,p,y) = \boldsymbol{S}(\boldsymbol{w})\{R_x(r)R_y(p)R_z(y)\}$$

Resolviendo $S(w)$

jacobiano

$$w = B(r,p,y)\begin{pmatrix}\dot{r}\\ \dot{p}\\ \dot{y}\end{pmatrix} = B(\Gamma)\dot{\Gamma} \qquad B(r,p,y) = \begin{pmatrix}1 & 0 & \sin(p)\\ 0 & \cos(r) & -\cos(p)\sin(r)\\ 0 & \sin(r) & \cos(p)\cos(r)\end{pmatrix}$$

geométrico $\quad v = J(q)\dot{q} = (v_x, v_y, v_z, w_x, w_y, w_z)^T$

analítico $\quad v' = J_A(q)\dot{q} = (v_x, v_y, v_z, \dot{r}, \dot{p}, \dot{y})^T$

$$\boldsymbol{J_A(q)} = \begin{pmatrix}\boldsymbol{I_{3x3}} & \boldsymbol{0_{3x3}}\\ \boldsymbol{0_{3x3}} & \boldsymbol{B^{-1}(\Gamma)}\end{pmatrix}\boldsymbol{J(q)} \qquad \Gamma = (r,p,y)^T$$

Modelo diferencial

Jacobiano inverso

Permite obtener las velocidades articulares, a partir de las velocidades en el espacio de la tareas

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_N \end{pmatrix} = J^{-1}(q) \begin{pmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{pmatrix} \qquad \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_N \end{pmatrix} = J_A^{-1}(q) \begin{pmatrix} v_x \\ v_y \\ v_z \\ \dot{r} \\ \dot{p} \\ \dot{y} \end{pmatrix}$$

Modelo diferencial
Jacobiano inverso

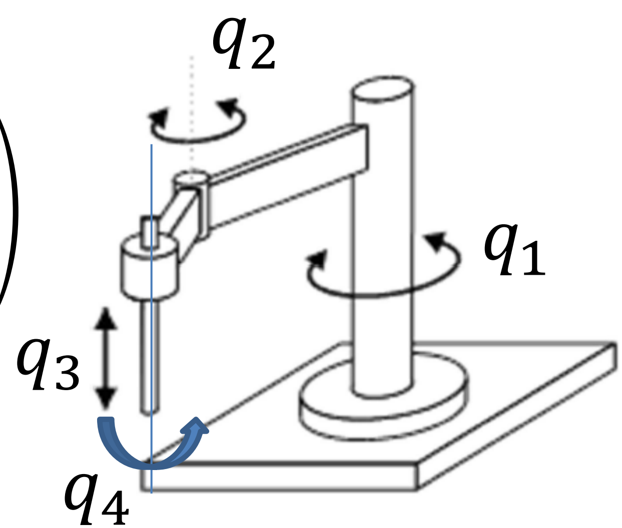
Posibles métodos de cálculo:

- Inversión simbólica de la matriz jacobiana
  → complejo 6x6
- Derivando el modelo cinemático inverso
- Evaluación numérica de J e inversión numérica
  - Necesidad de recómputo continuo
  - En ocasiones J no es cuadrada → pseudoinversa
  - En ocasiones $J = 0$ → puntos singulares

Modelo diferencial
Manipuladores infra y sobre actuados

- Si el robot es infra-actuado (underactuated): N<6

$$v_{XYZ} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ \cancel{w_x} \\ \cancel{w_y} \\ w_z \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ \cancel{J_{41}} & \cancel{J_{42}} & \cancel{J_{43}} & \cancel{J_{44}} \\ \cancel{J_{51}} & \cancel{J_{52}} & \cancel{J_{53}} & \cancel{J_{54}} \\ J_{61} & J_{62} & J_{63} & J_{64} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix}$$



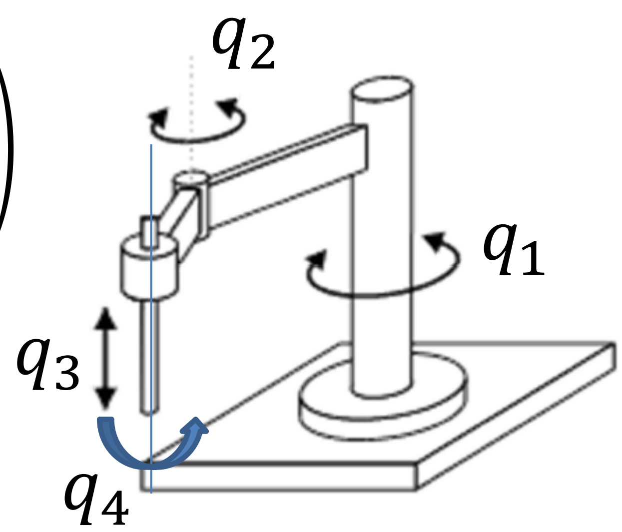
- Jacobiano no es cuadrado→ no puede ser invertido
- $[w_x, w_y]$ no se pueden controlar

Modelo diferencial

Manipuladores infra y sobre actuados

- Si el robot es infra-actuado (underactuated): N<6

$$v_{XYZ} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ w_z \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{61} & J_{62} & J_{63} & J_{64} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix}$$



- **Ahora el Jacobiano ya puede ser invertido**

Modelo diferencial

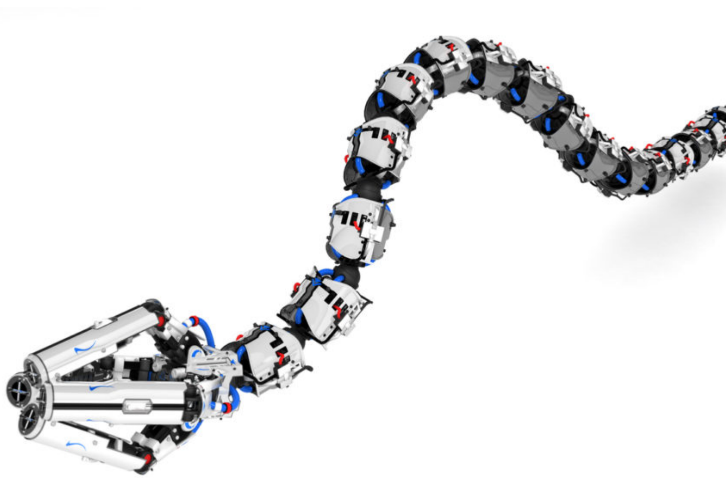
Manipuladores infra y sobre actuados

- Si el robot es sobre-actuado o redundante(overactuated):

N>6

$$v_{XYZ} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} & \cdots & J_{1N} \\ J_{21} & J_{22} & \cdots & J_{2N} \\ \cdots & \cdots & \ddots & \cdots \\ J_{N1} & J_{N2} & \cdots & J_{NN} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdots \\ \dot{q}_N \end{pmatrix}$$

Se pueden eliminar columnas



Snake-robot de OC 20GDL

https://www.youtube.com/watch?v=cV7aT8nvUL4

- No puede ser invertido el jacobiano
- Pero se puede usar la pseudoinversa

$$\dot{q} = J(q)^{+} v \qquad J^{+} = (J^T J)^{-1} J^T$$

Modelo diferencial
Manipuladores infra y sobre actuados

- Si el robot es sobre-actuado o redundante(overactuated):

N>6

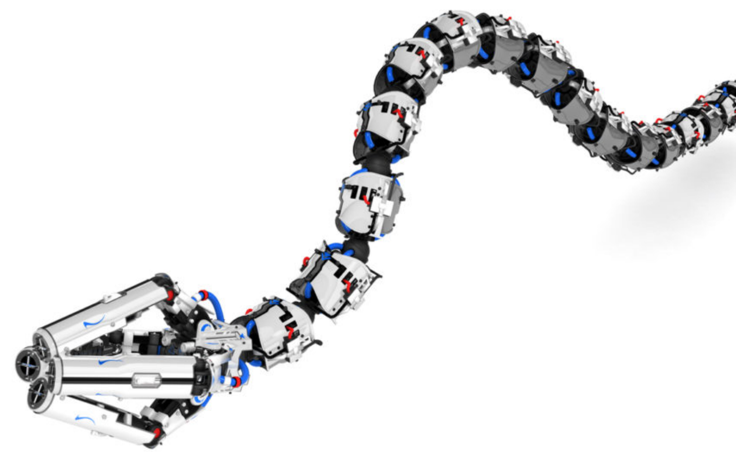
$$v = J(q)\dot{q}$$

$\dot{q}$ tiene infinitas soluciones

- Aplicando la pseudoinversa

$$\dot{q} = J(q)^{+} v \qquad J^{+} = (J^T J)^{-1} J^T$$

$\dot{q}$ **es la solución con menor norma** $\|\dot{q}\|$



Snake-robot de OC 20GDL

https://www.youtube.com/watch?v=cV7aT8nvUL4

Modelo diferencial

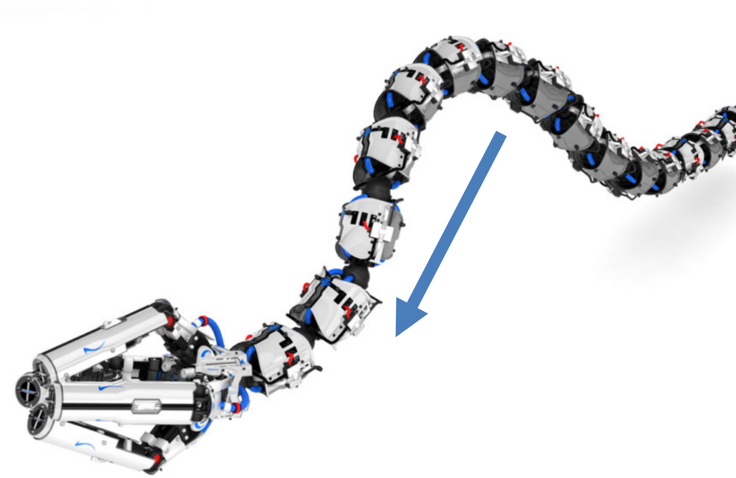
Manipuladores infra y sobre actuados

- Si el robot es sobre-actuado o redundante(overactuated):

N>6

$$\dot{q} = J(q)^{+}v + \boxed{NN^{+}\,\dot{q}_{NS}}$$

Articulaciones que no afectan a la posición final



Estas articulaciones son útiles para evitar obstáculos
Cambian la forma del manipulador

Snake-robot de OC 20GDL
https://www.youtube.com/watch?v=cV7aT8nvUL4

N **es el espacio nulo de J(q) o Ker{J(q)}**

- Conjunto de vectores $N = \{n_1, n_2, \ldots n_M\}$
- Tal que $J(q)n_i = 0$
- Número de vectores=nº articulaciones -6

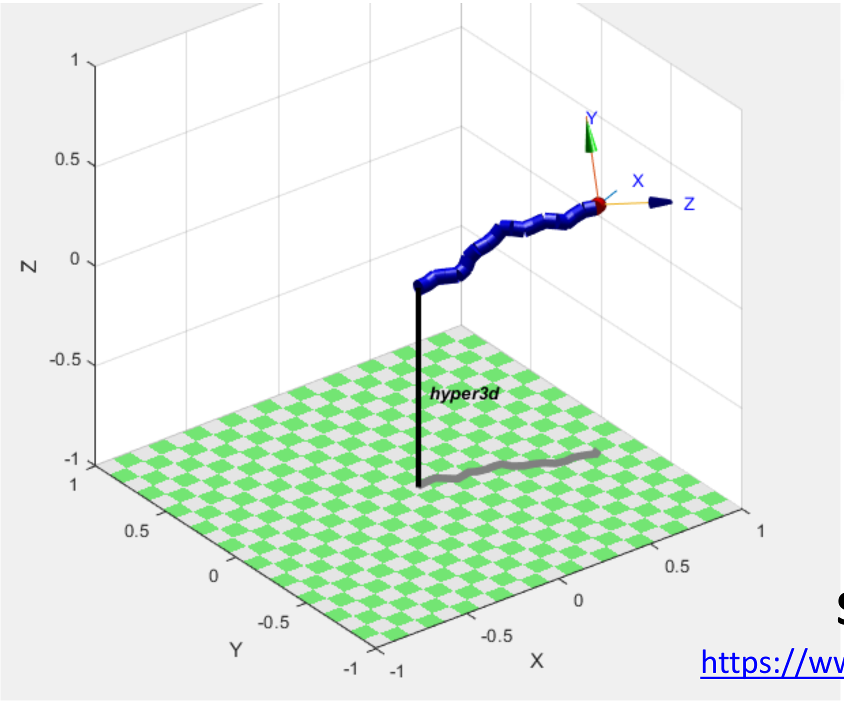
Modelo diferencial
Manipuladores infra y sobre actuados

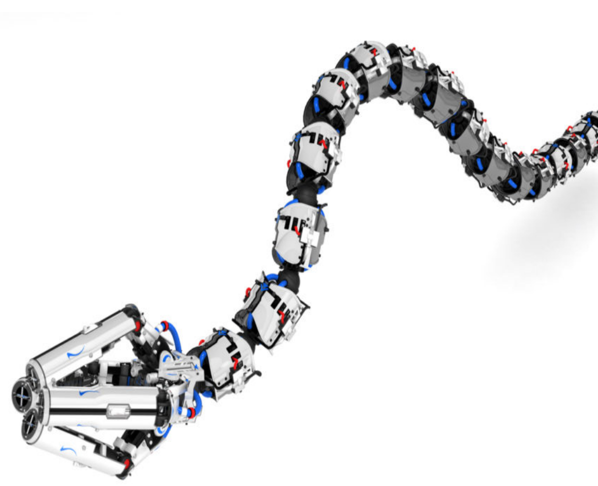
- Si el robot es sobre-actuado o redundante(overactuated):

Ejemplo Matlab

```
>>mdl_hyper3d(20)
>>q=rand(1,20)
>>h3d.plot(q)
```

N=20





Snake-robot de OC 20GDL

https://www.youtube.com/watch?v=cV7aT8nvUL4

Configuraciones singulares

- Cuando $|J(q)| = 0$ → $\mathrm{Ran}(J(q)) \neq Nfilas$
- $dp = J(q) \cdot dq$ →Incremento infinitesimal de las coordenadas del robot necesita un incremento infinito de las coordenadas articulares
- Implica pérdida de algún grado de libertad

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix}, \quad \boldsymbol{dx} = dq_1 + dq_2, \boldsymbol{dy} = 0,$$

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix}, \quad \boldsymbol{dx} = dq_1 + dq_2, \boldsymbol{dy} = 2dq_1 + 2dq_2 = 2dx,$$

Tipos:

- En los límites del espacio de trabajo→ bloqueo
- En el interior del espacio de trabajo

- **Al calcular las trayectorias debemos evitar pasar por los puntos singulares**

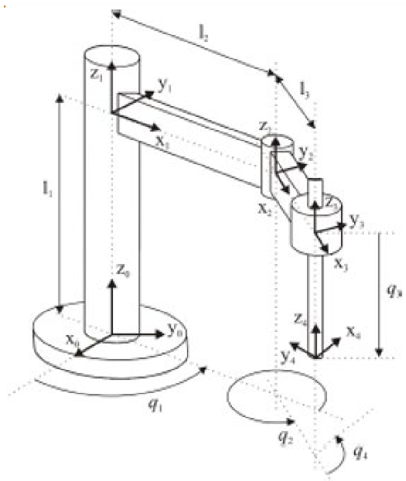
Configuraciones singulares

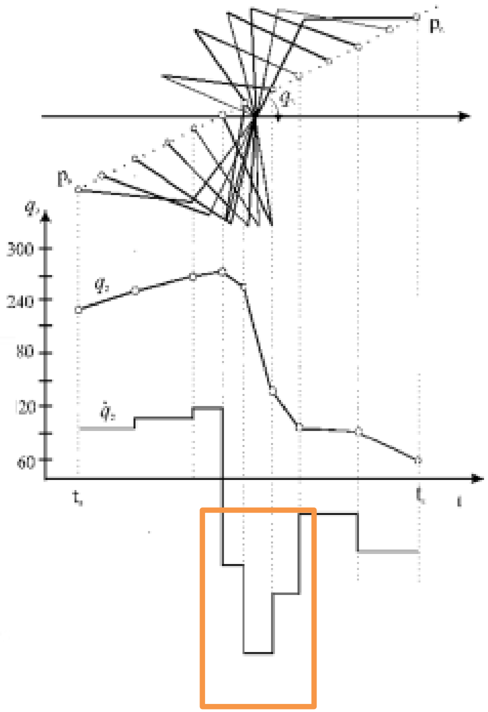
- Ejemplo Scara

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\left(l_3 S_{12} + l_2 S_1\right) & -l_3 S_{12} & 0 \\ l_3 C_{12} + l_2 C_1 & l_3 C_{12} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$Det\big(J(q)\big) = l_2 l_3 (S_{12} C_1 - S_1 C_{12}) = l_2 l_3 \sin(q_1 + q_2 - q_1) = l_2 l_3 \sin(q_2)$$

$Det\big(J(q)\big) = 0$ para $\boldsymbol{q_2=0}$ o $\boldsymbol{q_2 = \pi}$





La velocidad crece conforme q se acerca a π , en π vale ∞

Configuraciones singulares

- Ejemplo Scara

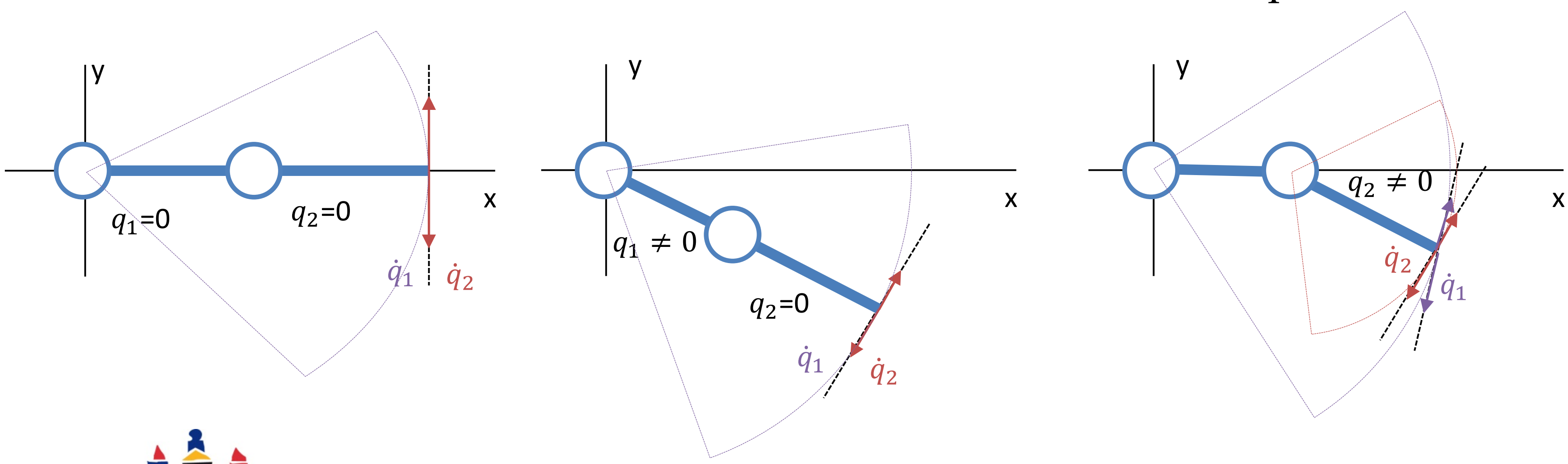
para $\boldsymbol{q_2}$=0 o $\boldsymbol{q_2 = \pi}$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-(l_3S_{12}+l_2S_1) & -l_3S_1\\ l_3C_{12}+l_2C_1 & l_3C_{12}\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-(l_3S_1+l_2S_1) & -l_3S_1\\ l_3C_1+l_2C_1 & l_3C_1\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}$$

$$\dot{x} = -(l_3S_1+l_2S_1)\dot{q}_1 - l_3\,S_1\dot{q}_2$$
$$\dot{y} = (l_3C_1+l_2C_1)\dot{q}_1 + l_3C_1\dot{q}_2$$

$$\dot{x} = -S_1[(l_3+l_2)\dot{q}_1 + l_3\,\dot{q}_2]$$
$$\dot{\boldsymbol{y}} = C_1[(l_3+l_2)\dot{q}_1 + l_3\dot{q}_2] = -\frac{\boldsymbol{C_1}}{\boldsymbol{S_1}}\,\dot{\boldsymbol{x}}$$



Bibliografía

- Antonio Barrientos, (2007) Fundamentos de Robótica, 2ª, Mc Graw Hill,
- Anibal Ollero Baturone, (2001) ROBOTICA Manipuladores y Robots Móviles,Marcombo, 84-267-1313-0,
- Hartenberg, R. S., & Denavit, J. (1955). A kinematic notation for lower pair mechanisms based on matrices.
- Cinemática del Brazo articulado PUMA. Jose Cortes Parejo
- Jaber A.A. (2017) PUMA 560 Robot and Its Dynamic Characteristics. In: Design of an Intelligent Embedded System for Condition Monitoring of an Industrial Robot. Springer Theses (Recognizing Outstanding Ph.D. Research). Springer, Cham
- Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright. “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions.” SIAM Journal of Optimization. Vol. 9, Number 1, 1998, pp. 112–147.
- Shanno, D. F. “Conditioning of Quasi-Newton Methods for Function Minimization.” Mathematics of Computing, Vol. 24, 1970, pp. 647–656.
- Coleman, T. F. and Y. Li. “An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds.” SIAM Journal on Optimization, Vol. 6, 1996, pp. 418–445.
- https://robotacademy.net.au/