Conceptos Generales

CONTROL Y PROGRAMACIÓN DE ROBOTS

Grado en Electrónica, Robótica y Mecatrónica

Índice

1. Introducción
2. Localización de objetos
3. Modelado cinemático
   - Modelo cinemático directo
   - Modelo cinemático inverso
   - Modelo diferencial. Singularidades.
4. Modelado dinámico
5. Control cinemático. Generación de trayectorias.
6. Control dinámico
7. Ejemplo de arquitectura (funcional) de control

Introducción

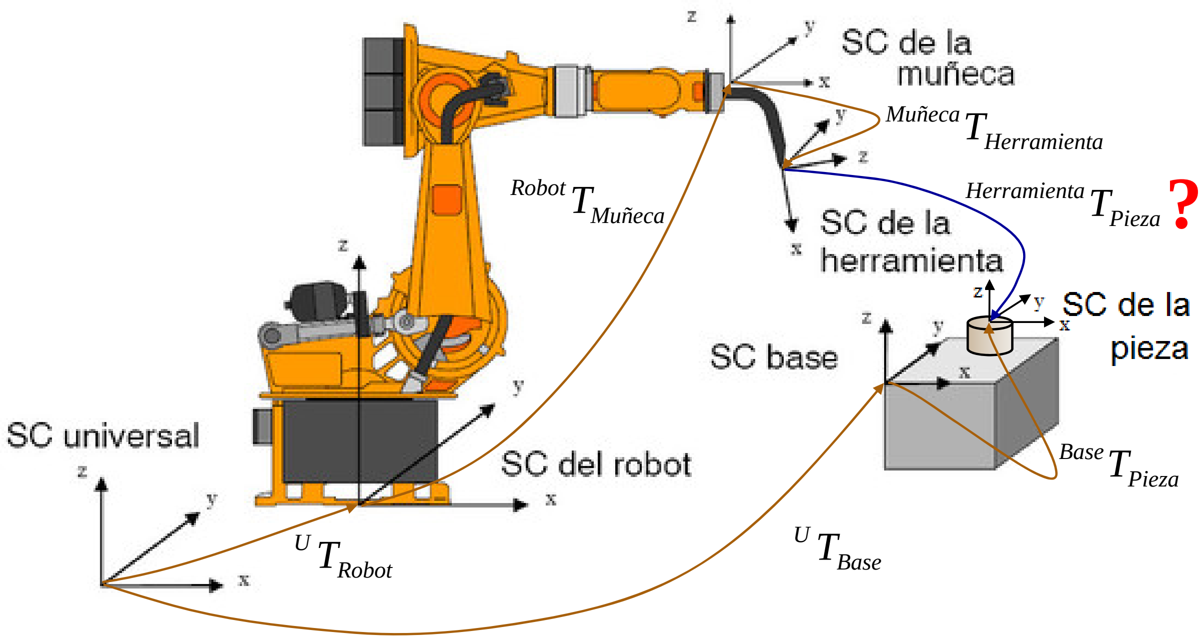
Objetivos:

- Exposición de problemas asociados al control de robots
- Conceptos a tener en cuenta en la programación

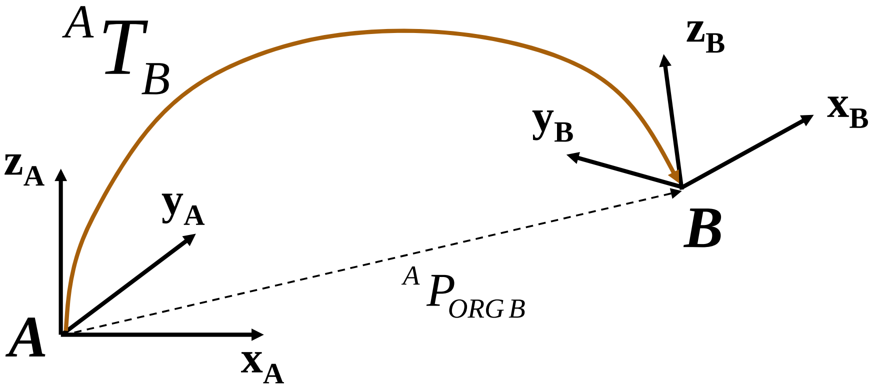
Repaso breve de la asignatura *Fundamentos de Robótica*

Localización de objetos

Necesidad:



Localización de objetos



$^{A}T_{B}$ debe contener datos de la localización de *B* con respecto a *A*

- **Translación** (origen de *A* distinto de origen de *B*)
- **Rotación** (*A* y *B* no están igualmente orientados)

Localización de objetos

Posibilidad:

$^{A}T_{B}$ como **Matriz de Transformación Homogénea**

$$^{A}T_{B} = \begin{pmatrix} ^{A}R_{B} & ^{A}P_{ORG\,B} \\ 0_{3\times 1} & 1 \end{pmatrix}$$

$^{A}P_{ORG\,B}$: Translación (coordenadas x,y,z del origen de B en ejes A)

$^{A}R_{B}$: Matriz de rotación de B con respecto a A
(sólo 3 parámetros independientes, p.e., ángulos de Euler)

Ángulos de Euler más utilizados: XYZ fijos, ZXZ móviles
y ZYZ móviles

Localización de objetos

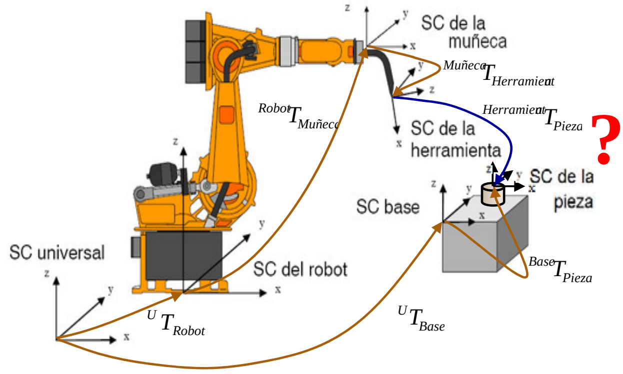
Propiedad de las MTH:

Camino desde U hasta pieza por robot:

$$^{U}T_{Pieza} = {}^{U}T_{Robot} \times {}^{Robot}T_{Muñeca} \times {}^{Muñeca}T_{Herramienta} \times {}^{Herramienta}T_{Pieza}$$

Camino desde U hasta pieza por base:

$$^{U}T_{Pieza} = {}^{U}T_{Base} \times {}^{Base}T_{Pieza}$$



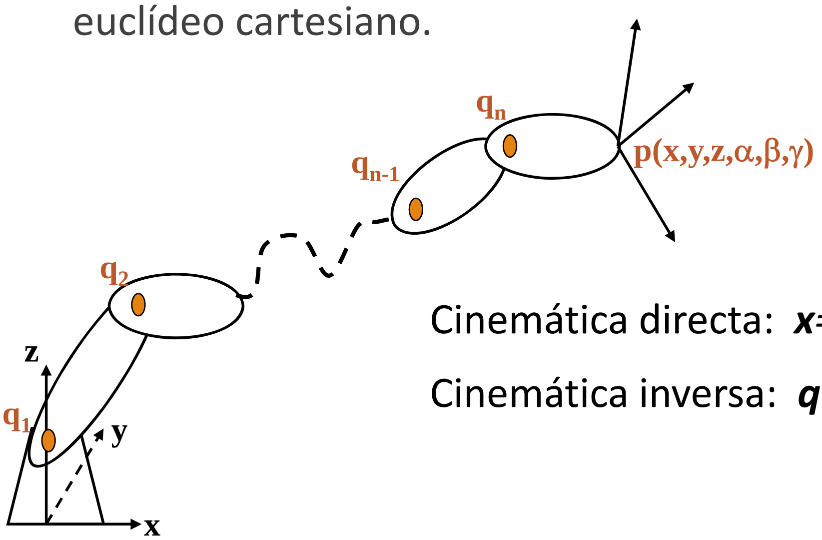
Igualando y despejando:

$$^{Herramienta}T_{Pieza} = \left({}^{U}T_{Robot} \times {}^{Robot}T_{Muñeca} \times {}^{Muñeca}T_{Herramienta}\right)^{-1} \times {}^{U}T_{Base} \times {}^{Base}T_{Pieza}$$

Modelado cinemático

Espacio articular y cartesiano (de la tarea):

- **q** = (q1, q2, ……, qn) : vector de variables articulares.
- **x** = (x,y,z,α,β,γ) : posición y orientación en el espacio euclídeo cartesiano.

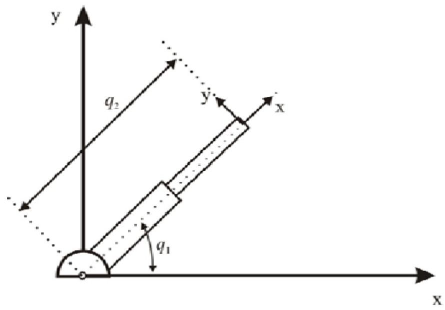


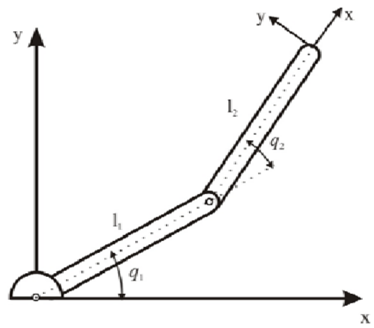
Cinemática directa: ***x***=*f*(***q***)

Cinemática inversa: ***q***=f^{-1}(***x***)

Modelo cinemático directo

- Conocer **posición y orientación** del extremo del robot **en función de sus coordenadas articulares** (GDL)
- Ejemplos:





$$
{}^{Muñeca}T_{Robot} = \left(\begin{array}{ccc|c} \cos(q_1) & -\sin(q_1) & 0 & q_2\cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & q_2\sin(q_1) \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

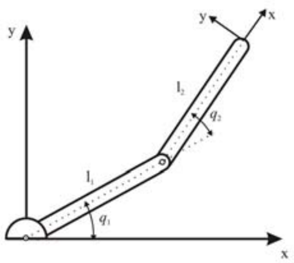
$$
\begin{aligned}
x &= q_2 \cos q_1 \\
y &= q_2 \,\mathrm{sen}\, q_1 \\
z &= 0 \\
[\mathbf{noa}] &= Rotz(q_1)
\end{aligned}
$$

$$
\begin{aligned}
x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\
y &= l_1 sen q_1 + l_2 sen(q_1 + q_2) \\
z &= 0 \\
[\mathbf{noa}] &= Rotz(q_1 + q_2)
\end{aligned}
$$

- Para cadenas cinemáticas más complejas, existen **métodos sistemáticos** para estos cálculos (*Denavit-Hartenberg*)

Modelo cinemático inverso

- Conocer **coordenadas articulares** del robot **en función de la posición y orientación** de su extremo.
- Problema **complejo sin solución analítica** en general.
- Ejemplo:

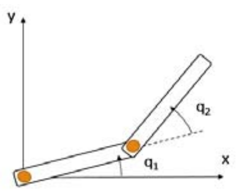


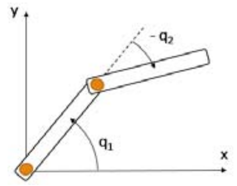
$$
\begin{aligned}
x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\
y &= l_1 sen q_1 + l_2 sen(q_1 + q_2)
\end{aligned}
$$

El cálculo de

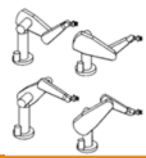
$(q_1,q_2)=f^{-1}(x,y)$

no es inmediato,
y proporciona
dos soluciones





- Posibilidad de **múltiples soluciones**, con necesidad de especificar en la programación cual es la deseada.

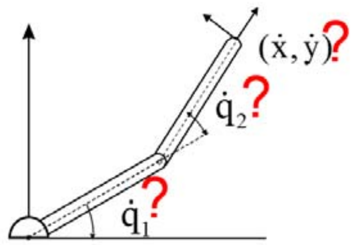


Modelo diferencial

- Relaciona las **velocidades** del extremo del robot en el **espacio cartesiano** (de la tarea) con las **velocidades** de las **articulaciones** del robot.

Modelo diferencial directo: $\dot{\boldsymbol{x}}=\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$

Modelo diferencial inverso: $\dot{\boldsymbol{q}}=\boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}$



- ***J(q)***: matríz **Jacobiana de las velocidades**
- No tiene por que ser cuadrada
  - En caso de no ser cuadrada inversa por pseudo inversa
- Necesario para el sistema de control.

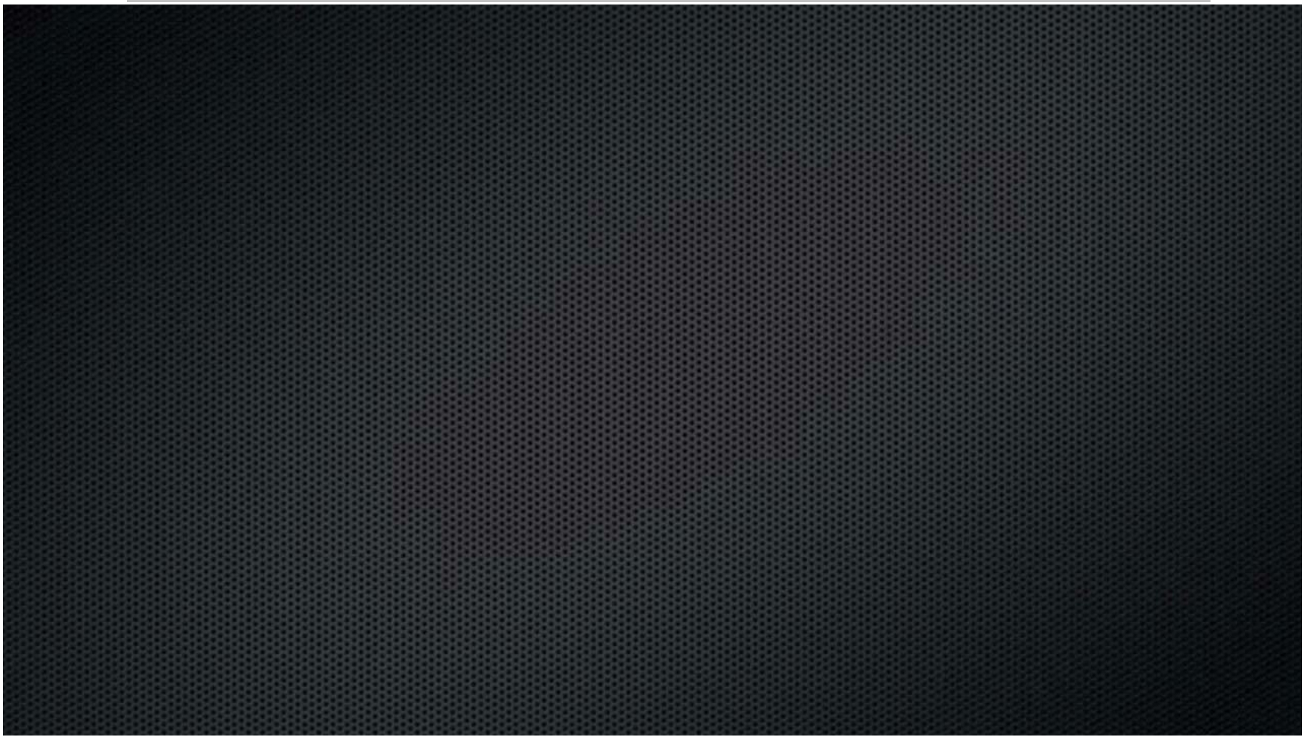
Singularidades

- Configuraciones de las articulaciones, ***q***, para las que la Jacobiana sea singular:

$$det(\boldsymbol{J}(\boldsymbol{q}))=0 \Rightarrow \boldsymbol{J}^{-1}(\boldsymbol{q}) \rightarrow \infty$$

- Para estas configuraciones, velocidades finitas en el espacio de la tarea demandan **velocidades infinitas de las articulaciones**.
- Clasificación según si se da en límites del espacio de trabajo o en su interior.
- Implica la **pérdida de algún grado de libertad**.
- **Evitar programar el robot en esas configuraciones**.
- Existen algoritmos que las detectan y las gestionan.

Singularidades



Modelado dinámico

Movimiento de las articulaciones en función de la fuerza/par que ejerce el actuador

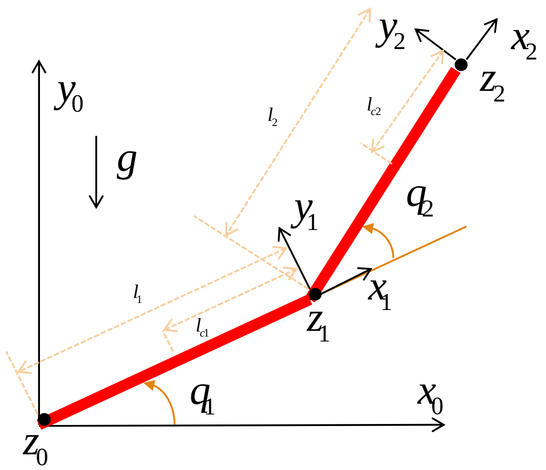
$$\boxed{R\tau_m = \left(M(q) + J_m R^2\right)\ddot{q} + \left(C(q,\dot{q}) + B_m R^2\right)\dot{q} + G(q) + F(\dot{q})}$$

Matriz de Reductoras (diagonal)
Par de motores
Matriz Dinámica
Matriz Inercias de motores
Matriz Centrípeta y Coriolis
Matriz Fricción viscosa de motores
Vector Gravitatorio
Vector Fricciones

- Obtención a partir de ecuaciones:
  - Lagrange: en términos energéticos
  - Newton/Euler: equilibrio de fuerzas y pares
- Sistema de ecuaciones diferenciales acopladas.
- **Relación alta en reductores (R>>1) ayuda al desacoplo**.

Modelado dinámico

Robot plano RR vertical



$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \qquad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

Eslabones

$m_1 = m_2 = 3\,Kg \qquad l_1 = l_2 = 1\,m$

$I_1 = I_2 = 0.2536\ Kg.m^2 \quad l_{c1} = l_{c2} = 0.5\ m$

Motores

$J_{m1} = J_{m2} = 0.025\ Kg.m^2$

$B_{m1} = B_{m2} = 3.6\times 10^{-6}\ \ Nm/(rad/s)$

$K_{t1} = K_{t2} = 10\ \ Nm/A$

$R_1 = R_2 = 25 \quad vs \quad R_1 = R_2 = 1$

Modelado dinámico

Robot plano RR vertical

$$K_t R I_m = \left(M(q) + J_m R^2\right)\ddot{q} + \left(C(q,\dot{q}) + B_m R^2\right)\dot{q} + G(q) + F(\dot{q})$$

$R_1 = R_2 = 25$

$$\begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix}\begin{bmatrix} \tau_{m1} \\ \tau_{m2} \end{bmatrix} = \begin{bmatrix} 20.8572 + 3.0\cos(q_2) & 1.0036 + 1.5\cos(q_2) \\ 1.0036 + 1.5\cos(q_2) & 16.8536 \end{bmatrix} * \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}$$

$$+ \begin{bmatrix} -1.5\sin(q_2)\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right) + 0.0022\dot{q}_1 \\ -1.5\sin(q_2)\dot{q}_1^2 + 0.0022\dot{q}_2 \end{bmatrix} + \begin{bmatrix} 44.1\cos(q_2) + 14.7(q_1 + q_2) \\ 14.7\cos(q_1 + q_2) \end{bmatrix}$$

$R_1 = R_2 = 1$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tau_{m1} \\ \tau_{m2} \end{bmatrix} = \begin{bmatrix} 5.0326 + 3.0\cos(q_2) & 1.0036 + 1.5\cos(q_2) \\ 1.0036 + 1.5\cos(q_2) & 1.029 \end{bmatrix} * \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$+ \begin{bmatrix} -1.5\sin(q_2)\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right) + 3.6\times 10^{-6}\dot{q}_1 \\ -1.5\sin(q_2)\dot{q}_1^2 + 3.6\times 10^{-6}\dot{q}_2 \end{bmatrix} + \begin{bmatrix} 44.1\cos(q_2) + 14.7(q_1 + q_2) \\ 14.7\cos(q_1 + q_2) \end{bmatrix}$$

Control cinemático

- **Generar las trayectorias** de referencia que debe seguir **cada articulación** del robot a lo largo del tiempo para las distintas órdenes de movimiento.
- Se debe tener en cuenta:
  - Punto de destino
  - Tipo de trayectoria del extremo
  - Tiempo invertido
  - etc..

MOVELD POS 300

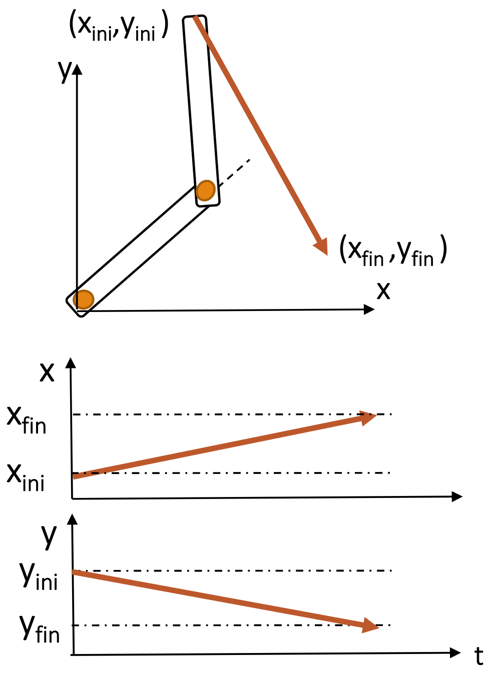
- Necesario **conocimiento de modelo cinemático**.
- Es necesario atender a las restricciones físicas de los accionamientos y criterios de calidad (suavidad, precisión...)

Control cinemático

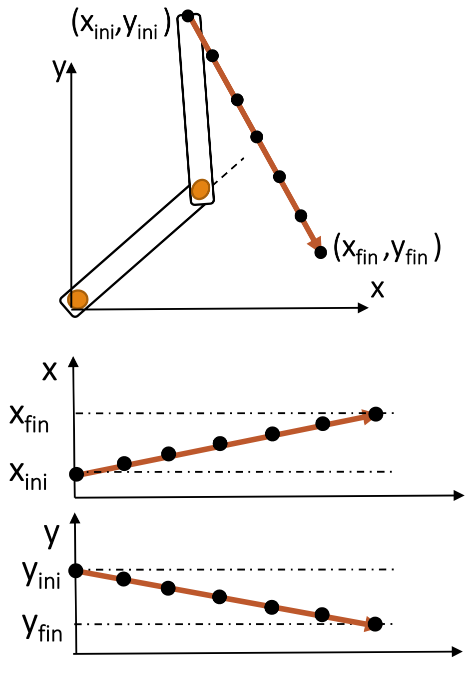
- **Pasos para generar trayectorias:**
  1. A partir de orden de movimiento, calcular ecuaciones de trayectorias en cartesianas
  2. Obtención de puntos concretos muestreando en le tiempo la trayectoria en cartesianas
  3. Conversión de puntos en cartesianas a espacio articular con modelo cinemático inverso
  4. Generación de trayectorias (**posición, velocidad y aceleración**) en espacio articular mediante interpolación de coordenadas articulares en el tiempo
     - Interpolador con splines
     - Interpolador con velocidad trapezoidal

Control cinemático

- **Paso 1:**

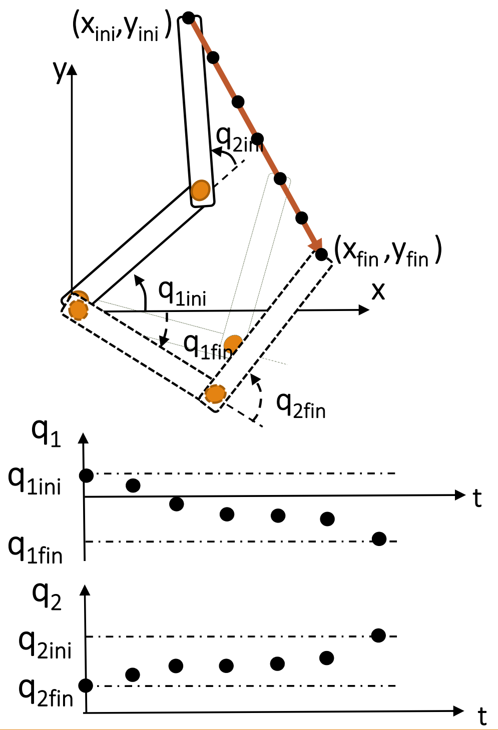


- **Paso 2:**

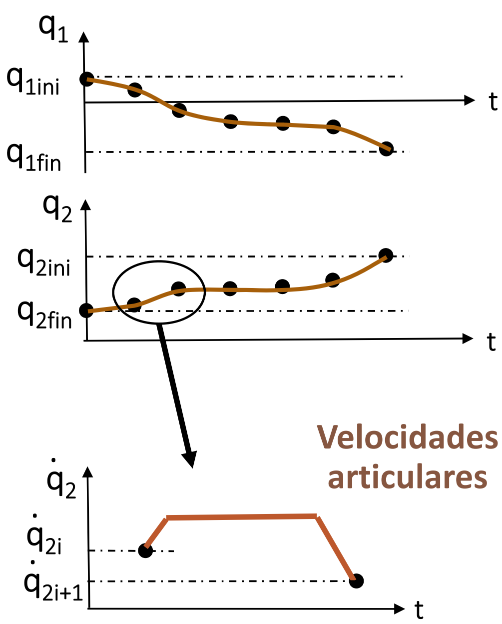


Control cinemático

- **Paso 3:**



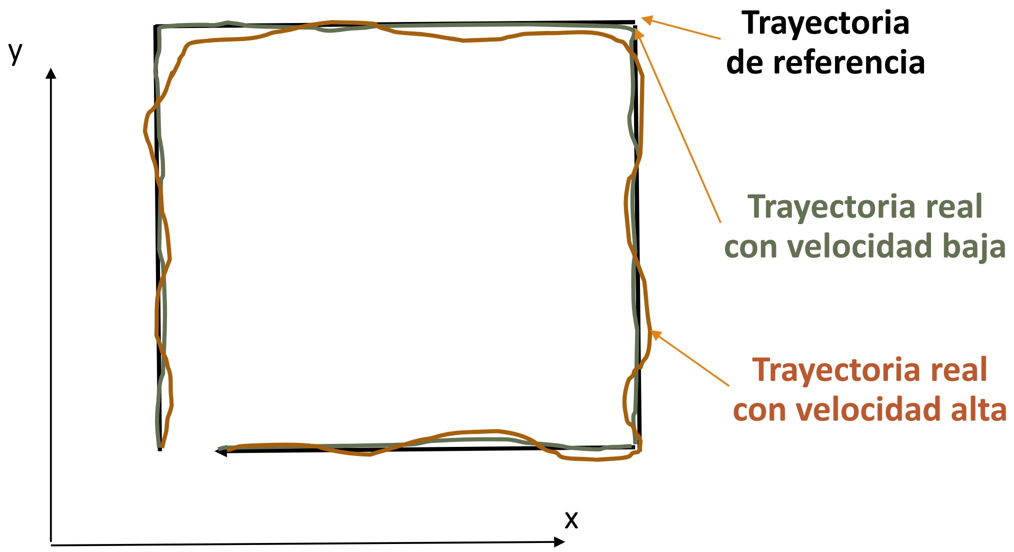
- **Paso 4:**



Control dinámico

- Gestionar de manera adecuada las **señales de control** de los actuadores (motores) de manera que las **posiciones articulares reales del robot se parezcan la más posible a las trayectorias de referencia** generadas por el control cinemático.
- **Control en bucle cerrado**: necesidad de sensores internos de posición y velocidad.
- **A mayor velocidad de movimiento, peor comportamiento (errores más grandes) en control del robot.**
- En general es necesario **estrategias de control avanzadas**, si bien el problema se simplifica con factores altos en los reductores.

Control dinámico



Arquitectura de control

• Arquitectura funcional

Ejemplo: