

INTRODUCCIÓN A PROGRAMACIÓN DE ROBOT

Departamento de Ingeniería de Sistemas y Automática (DISA)
Escuela de Ingenierías Industriales (EII)
Universidad de Valladolid (UVa)

Curso 2023/2024

CONTENIDOS

1 Introducción a los Lenguajes de Programación de Robots

- **Componentes mecánicos-eléctrico:** Son todos lo brazos, ejes y motores del robot.
- **Componentes electrónicos (controlador):**
 - **Ordenador:** Se dispone de un ordenador con su sistema operativo conectado a todos los actuadores y sensores del robot y otras conexiones al exterior.
 - **Pantalla de operario:** Pantalla táctil con ventanas para poder programar los movimientos y conexiones del robot.
Ejemplos: **ABB:** Flex Pendant **RAPID FANUC:** Teach Pendant **UR:** PolyScope
- **Software de simulación:** La mayoría de robot disponen de un software de simulación para PC del robot y controlador para poder programar el mismo fuera de linea.
ABB: RobotStudio **FANUC:** RoboGuide **UR:** URSim_VIRTUAL
- **Software generales de simulación:** Existen algunos programas que pueden simular robot de distintas marcas.
 - **RoboDK:** Simula muchos robot industriales con un lenguaje propio que luego puede ser traducido al del robot.
 - **RobCad:** Tiene módulos de muchos robot industriales para poderlos programar fuera de linea.
 - **Matlab-Simulink:** Librerías “Robotic Toolbox” (matlab) y “Multi-Body” (Simulink).

LENGUAJES DE PROGRAMACIÓN DE ROBOT

- **Lenguajes de programación:** Los movimientos y conexiones del robot deben ser programados desde el software de simulación y/o desde la pantalla. Todo lenguaje tiene dos tipos de sentencias,
 - **Sentencias de movimiento:** Para mover el robot de un punto a otro respecto de una determinada referencia.
 - **Sentencias de control y comunicación:** Son muy similares a las de cualquier otro lenguaje de programación.
- Ejemplo **ABB:** RAPID **FANUC:** KAREL **UR:** Script
- **Lenguajes del programación simplificados:** Algunos fabricantes usan lenguajes simplificados para programar desde la pantalla de operario.
ABB: RAPID (por comandos) **FANUC:** lenguaje TP **UR:** Lenguaje PolyScope

- **Definición:** Un eje viene determinado por una posición en el espacio y una orientación con respecto a otro eje definido.
- **Posición:** Coordenadas x, y, z , medido en metros o milímetros respecto a una referencia.
- **Orientación:** Existen muchas maneras de medir la orientación del eje x , eje y y eje z respecto a una referencia (ver apéndice).
 - **Ángulo de Euler:** Rotar el eje de referencia un ángulo en z , R_z , otro en y , R_y , y nuevamente en z , R_z para conseguir el nuevo eje (3 grados de libertad, R_{zyz}). Simple pero con problemas de singularidad.
 - **Ángulo de Cardano:** Rotar el eje de referencia un ángulo en x , R_x , otro en y , R_y , y un ángulo en z , R_z para conseguir el nuevo eje (3 grados de libertad, R_{xyz}).
 - **Matriz de rotación ortogonal (3×3):** (9 componentes, 3 grados de libertad),
 - Obtenida por el vector del eje z (aproximación $[a_x, a_y, a_z]'$), el eje y (orientación $[o_x, o_y, o_z]'$) y eje x (normal $[n_x, n_y, n_z]'$), $R = [n, o, a]$.
 - Obtenida por la composición de las matrices asociadas a la rotación R_z , R_y y R_x u otro posible combinación.
 - **Vector-ángulo:** Se define un vector de giro v (unitario) y un ángulo θ (4 componentes, 3 grados de libertad).
 - **Cuaternios:** A partir de v y θ se define un vector de 4 componentes $[\cos(\theta/2), v * \sin(\theta/2)]$. Se trabaja con cuaternios unitarios (4 componentes, 3 grados de libertad).



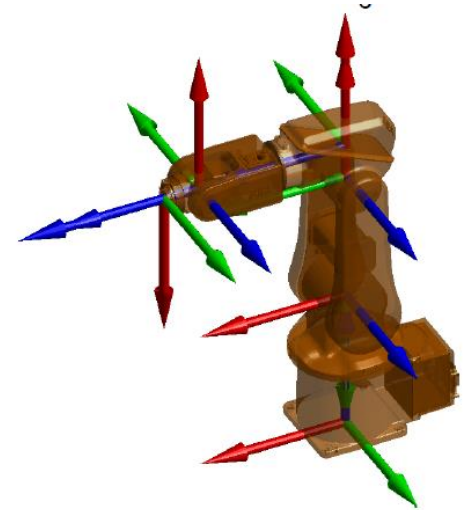
REPRESENTACIÓN DE EJES Y PUNTOS OBJETIVOS (II)

- **Posición y Orientación en los lenguajes de programación:**
 - Todos los métodos indicados tienen 6 grados de libertad, 3 de posición y 3 de orientación, y son equivalentes.
 - Ejemplos de unidades y métodos en diferentes marcas (Torre de Babel):
ABB: RAPID (mm y cuaternios) RobotStudio (mm y R_{xyz} grados) **UR:** PolyScope (mm y R_{xyz} grados) Script (m y R_{xyz} radianes).
- **Matriz homogénea:** Incluye en una matriz (4×4) el vector desplazamiento y la matriz rotación, ${}^A T_B = \begin{bmatrix} {}^A R_B & t \\ 0_{1 \times 3} & 1 \end{bmatrix}$, donde $t = [x, y, z]'$ es la traslación y ${}^A R_B$ la rotación entre el sistema B respecto del A .
- **Composición de sistema:** Si ${}^A T_B$ es la matriz homogénea entre B y A , y ${}^B T_C$ entre C y B , la matriz homogénea entre C y A será, ${}^A T_C = {}^A T_B * {}^B T_C$.
- **Paso de un eje a otro:** Un punto $t = [x, y, z]'$ del espacio visto desde A y B guardan la relación, $\begin{bmatrix} {}^A t \\ 1 \end{bmatrix} = {}^A T_B * \begin{bmatrix} {}^B t \\ 1 \end{bmatrix}$.



- **Tipos de ejes de un robot:**
 - **Rotacionales:** Un brazo gira respecto del brazo anterior en un nodo.
 - **Prismáticos:** Un brazo se desplaza respecto al brazo anterior.
- **Robot industriales antropomórficos:**
 - Simulan un brazo humano, con antebrazo, brazo y muñeca.
 - Suelen tener 6 grados de libertad, normalmente, 6 ejes rotacionales.
- **Ejes del robot:** Posición y rotación de cada nodo respecto al anterior.
Ejemplo ABB irb120,

- **Eje base:** Define la referencia de la base del robot.
- **Ejes de los nodos:** Definen la posición de cada nodo.
- **Eje z del nodo:** Debe definir el giro o traslación del nodo (variable).
- **Punto de trabajo (TCP):** Eje del último nodo.
- **Codos:** Ejes [1,4,6], definen la posición del antebrazo, brazo y muñeca. Puede haber 8 formas de ir a un punto.



CINEMÁTICA DIRECTA E INVERSA (I)

- **Método DenavitHartenberg (D-H) para obtener los ejes de los nodos:** De un eje $n - 1$ al siguiente eje n ,

- Rotar en z (Variable)
- Trasladar en z (Fijo)
- Trasladar en x (Fijo)
- Rotar en x (Fijo)

$${}^{n-1}T_n(R_{z_n}) = T(R_{z_n}) * T(t_{z_n}) * T(t_{x_n}) * T(R_{x_n})$$

Ejemplos:

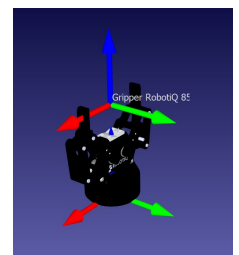
- De $n = 0$ a $n = 1$ en el ABB irb120:
 R_{z_1} (variable), $t_{z_1} = 290$, $t_{x_1} = 0$, $R_{x_1} = -90$.
- De $n = 1$ a $n = 2$ en el ABB irb120:
 $R_{z_2} - 90$ (variable, giro x), $t_{z_2} = 0$, $t_{x_2} = 270$, $R_{x_2} = 0$.
- **Punto de trabajo (TCP):** Posición final del robot, 6º eje o la herramienta.
- **Coordenadas cartesianas del TCP:** Se obtiene por composición de ejes, ${}^0T_6(R_{z_1}, \dots, R_{z_6}) = {}^0T_1(R_{z_1}) * {}^1T_2(R_{z_2}) * \dots * {}^5T_6(R_{z_6})$.
- **Coordenadas articulares del TCP:** Ángulo (rotacional) de cada uno de los 6 ejes del robot $[R_{z_1}, \dots, R_{z_6}]$.



- **Cinemática directa:**
 - Paso de las coordenadas articulares a las cartesianas del TCP.
 - Se obtiene multiplicando las matrices homogéneas de todos los ejes.
- **Cinemática inversa:**
 - Paso de las coordenadas cartesianas a articulares.
 - Puede haber 8 posibles posiciones, dependiendo de la orientación de los codos.
 - Se obtiene por métodos analíticos o de optimización computacional.
- **Punto objetivo:** Punto y orientación a la que se quiere llevar el TCP del robot.
 - **IMPORTANTE:** La orientación del eje z del punto debe ser coherente con el eje z del robot.
 - **Ejemplo:** El punto $t = [200, 0, 0]$, $R_{xyz} = [0, 0, 0]$ no es alcanzable. Se debe girar para orientar el eje z hacia abajo, $R_{xyz} = [180, 0, 0]$.

HERRAMIENTA DE TRABAJO

- **Elementos:** Normalmente está anclada en el robot.
 - Se compone de dos ejes:
 - **Base:** normalmente asociada al 6º
 - **Extremo:** Punto de trabajo de la herramienta (TCP)
 - T_{tool} : Relación entre ejes
En este caso, $t = [0, 0, 180]$, $R_{xyz} = [0, 0, 0]$ Solo varia z.
 - **TCP herramienta:** Las coordenadas del robot más la herramienta, $T_{TCP} = {}^0T_6 * T_{tool}$
- **Regla de los 4 puntos:** Para buscar la $t_{tool} = [x, y, z]$ de forma geométrica.
 - Se coloca el TCP herramienta en un mismo punto con 4 orientaciones diferentes.
 - El punto final t_{TCP} es el mismo, pero el punto del 6º eje (t_6) es diferente.
 - La distancia entre los 4 puntos t_6 y el t_{TCP} es la misma y generan 4 ecuaciones con 3 incógnitas. La cuarta se usa como test.



-

FUNCIONES Y ARGUMENTOS DE LOS MOVIMIENTOS DEL ROBOT

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ▶ ↺ 🔍 ↻

- **Programación básica:** Los lenguajes de programación tienen,
 - **Variables, funciones, módulos:** Tienen la estructura de un lenguaje modular normal.
 - **Sentencias de control:** Bucles y condicionales.
 - **Interrupciones:** Algunos lenguajes tienen interrupciones para procesar módulos en paralelo.
- **Comunicaciones:** Los lenguajes de programación tiene funciones específicas para comunicarse por otros dispositivos,
 - **Tarjeta entrada-salida:** Posibles variables analógicas y digitales.
 - **Puerto Internet:** Para comunicaciones con PCs basadas en TCP/IP, OPC, ROS.
 - **Puerto Profibus-Profinet:** Para comunicaciones industriales con autómatas.