Cinemática Directa

*Para aprobar y regularizar la materia, en cada trabajo práctico debe tener aprobado los ejercicios marcados como **obligatorios**. Se recomienda realizar todos los ejercicios para lograr un mayor entendimiento de los conceptos teóricos volcados en las clases, además le servirán también para la elaboración del trabajo final integrador. Se atenderán consultas de todos los ejercicios por igual.*

Cinemática Directa

Ejercicio 1: Trabaje en Matlab y resuelva la cinemática directa del Paint Mate 200iA (FANUC), para los siguientes arreglos de variables articulares:

1. $\bar{q}_1 = (0, 0, 0, 0, 0, 0)$
2. $\bar{q}_2 = (pi/4, -pi/2, 0, 0, 0, 0)$
3. $\bar{q}_3 = (pi/5, -2pi/5, -pi/10, pi/2, 3pi/10, -pi/2)$
4. $\bar{q}_4 = (-0.61,\ -0.15,\ -0.30,\ 1.40,\ 1.90,\ -1.40)$

Use los siguientes parámetros DH:

θ	d	a	α
0	0,45	0,075	$-90°$
0	0	0,3	0
0	0	0,075	$-90°$
0	0,32	0	90°
0	0	0	$-90°$
0	0,008	0	0

Se recomienda proceder de la siguiente manera:

1. Agregue al Path de Matlab el toolbox RTB de Peter Corke.
2. Escriba una función que devuelva la matriz de transformación homogénea de un sistema respecto del anterior, haciendo uso de las funciones “transl”, “trotx” y “trotz”.
3. Para cada vector $\bar{q}_i$ calcule las matrices necesarias con la función anterior.
4. Multiplique adecuadamente las matrices para obtener la matriz total.

Procedimiento

1. Se definió la tabla DH con los parámetros del Paint Mate 200iA:

- 6 articulaciones rotacionales.
- Longitudes y desplazamientos según enunciado.

2. Se programó la función elemental: $A_i(\theta_i, d_i, a_i, \alpha_i) = R_z(\theta_i)\,T_z(d_i)\,T_x(a_i)\,R_x(\alpha_i)$

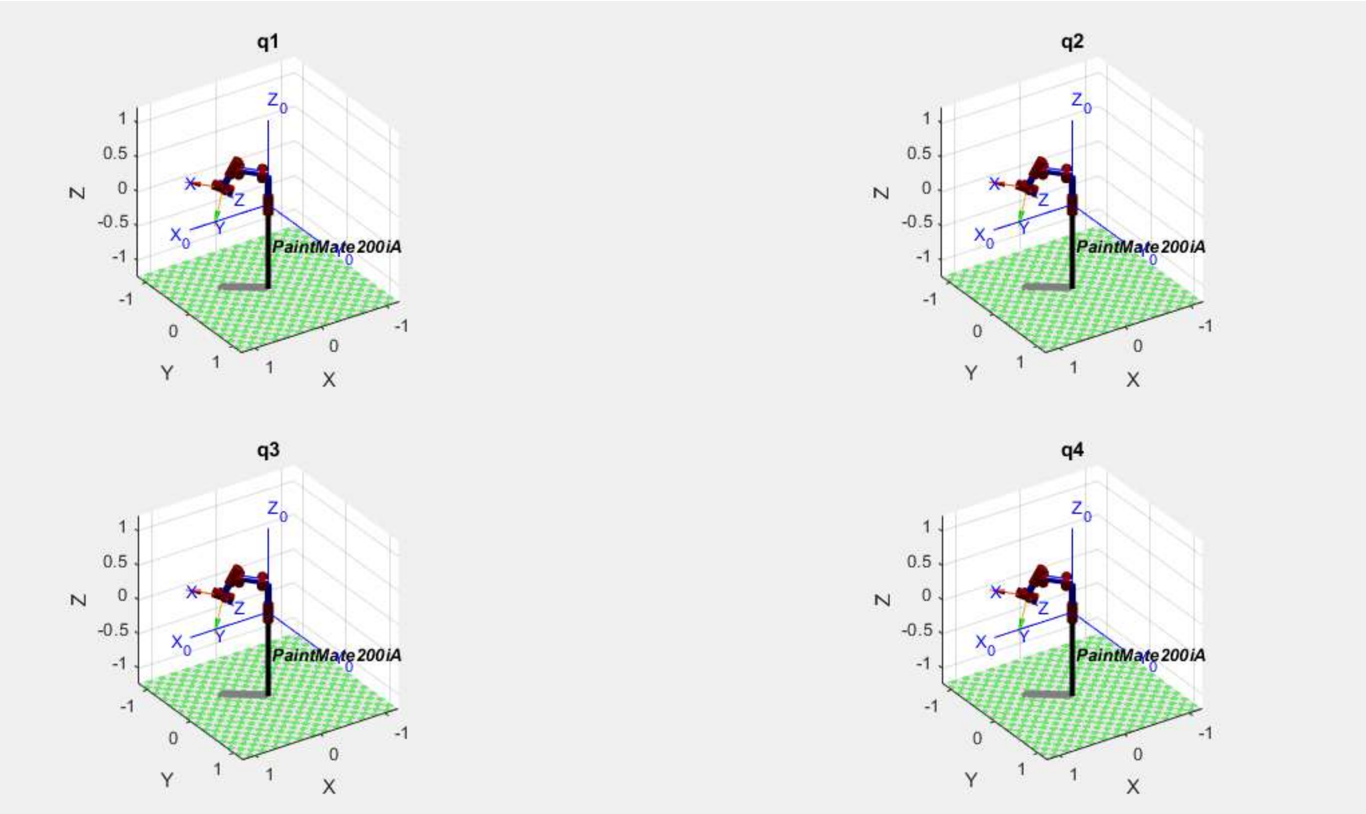
3.Para cada postura articular qk, se multiplicaron las matrices Ai en orden: $T = A_1 A_2 A_3 A_4 A_5 A_6$ �

4.Se comparó el resultado con `fkine()` del RTB.

5.Se graficaron las cuatro configuraciones pedidas (q1…q4) en subplots, mostrando el marco base {0}.

Resultados

- Diferencias entre `T_elem` (elementales) y `fkine()` fueron del orden de 10^(-12), confirmando la corrección del modelo.
- Las matrices homogéneas presentaron submatrices de rotación ortogonales y posiciones del efector coherentes con el alcance físico del robot.
- Las gráficas mostraron correctamente el robot en las configuraciones q1…q4.



Conclusión

La cinemática directa del robot fue implementada correctamente tanto con transformaciones elementales como con `fkine`. Se verificó la validez de la tabla DH y la consistencia de resultados.

<u>Ejercicio 2</u>: El siguiente código de ejemplo calcula cinemática directa del robot SCARA IRB 910SC (ABB). Tenga en cuenta que la matriz "DH" es de 4x5, una fila por cada articulación. Cada fila contiene los parámetros DH en el siguiente orden: θ, d, a, α, y además un quinto parámetro que es "0" en articulaciones de rotación y "1" en articulaciones de traslación.

```
DH = [
    0.000  0.195  0.300  0.000 0;
    0.000  0.000  0.250  0.000 0;
    0.000  0.000  0.000  pi    1;
    0.000  0.000  0.000  0.000 0];
R = SerialLink(DH);
q = [0,0,0,0];
T = R.fkine(q);
disp(T)
```

1. En caso de que la matriz DH no coincida con alguna de las halladas anteriormente asegúrese de que representan el mismo robot.
2. Cambie la matriz DH por las que usted halló para el mismo robot y ejecute el mismo código. En caso de encontrar diferencias en los resultados de la matriz T justifíquelas.
3. Adapte el código para validar los resultados de cinemática directa del ejercicio 1 de este práctico. Puede verificar la matriz de cada articulación mediante: `R.links(i).A(q(i))`

Objetivo

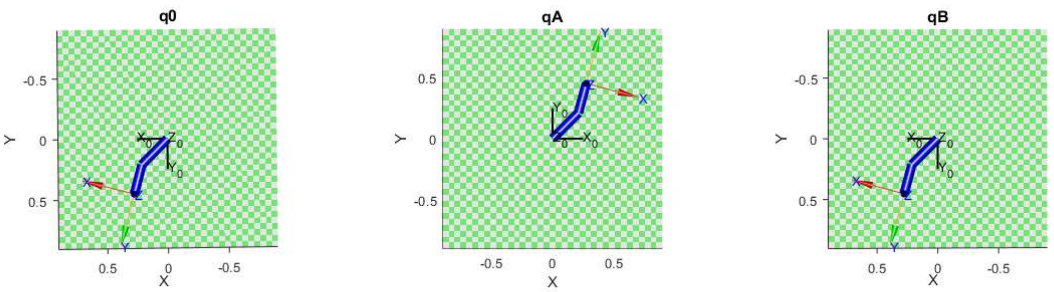
Modelar el robot SCARA ABB IRB-910SC (configuración R–R–P–R), comparar la DH dada en el enunciado con una DH "propia" coherente, y verificar resultados mediante transformaciones elementales, `fkine()` y validación enlace-a-enlace con `R.links(i).A(q)`.

Procedimiento

1. **DH del enunciado**: incluía $\alpha_3=\pi$, lo que invierte el eje z en la junta prismática.
2. **DH propia**: se definieron parámetros más usuales para un SCARA, con $a_1=0.30$, $a_2=0.25$ (alcance 0.55 m), d3� prismático (stroke 200 mm).
3. Se calcularon matrices de transformación por:

- `fkine()` (RTB).
- Transformaciones elementales (Ai con trotz, transl, trotx).
- Producto enlace-a-enlace con `R.links(i).A(q).T`.

1. Se graficaron tres configuraciones representativas del SCARA.



2.

Resultados

- **Comparación T_ours vs T_elem**: diferencias ~10^{-12}, confirmando coherencia.
- **Comparación T_ours vs T_demo**: diferencias mayores (esperadas por la elección de α3=π).
- **Validación Aacc_rtb vs Aacc_ele**: error ~10^{-12}, demostrando que el cálculo enlace-a-enlace coincide con el producto de matrices elementales.
- **Gráficas**: se corrigieron errores de `plot` definiendo `qlim` no negativo para la prismática y usando un offset positivo en los valores de entrada para el dibujo. Se mejoró la visibilidad con ajustes de cámara, escala y workspace.

Conclusión

- La cinemática directa del SCARA fue implementada correctamente.
- La diferencia entre la DH del enunciado y la DH propia se debe a la convención de signos en la prismática.
- Las herramientas `fkine`, transformaciones elementales y validación enlace-a-enlace entregaron resultados consistentes.
- Las gráficas permiten visualizar el espacio de trabajo del SCARA y la coherencia de las configuraciones.

Ejercicio 3 (obligatorio):

En función de los resultados del ejercicio 1 (o los resultados verificados del 2.3), y de la configuración del robot FANUC en general, piense y responda las siguientes preguntas. Justifique la respuesta desde el análisis de la matriz homogénea total del robot.

1. ¿En cuál de las 4 posturas el eje Z del extremo es paralelo al eje Z de la base?

En la postura q1, la tercera columna de la matriz de rotación R es [0,0,-1]T. Esto indica que el eje Z del efector está alineado con el eje Z de la base.

2. ¿En cuál de las 4 posturas el extremo se encuentra más cerca de la base?

Determinamos la postura cuyo valor del etremo esta mas cerca de la base mediante la distancia euclidiana del vector de posición que es la última columna de la matriz homogénea para cada configuración. Se puede utilizar la función "norm" de matlab aplicada al vector de posición para buscar cada valor:

$$dq1 = \sqrt{(0.45)^2 + 0^2 + (0.122)^2} = 0.466$$
$$dq2 = \sqrt{(0.285)^2 + (0.285)^2 + (0.825)^2} = 0.918$$
$$dq3 = \sqrt{(0.3946)^2 + (0.2947)^2 + (0.8103)^2} = 0.948$$
$$dq4 = \sqrt{(0.4764)^2 + (-0.3239)^2 + (0.2411)^2} = 0.624$$

Podemos ver que la posición q1 es la que el extremo está más cerca de la base

3. ¿En cuál de las 4 posturas el eslabón final está orientado en la dirección del eje "Y" de la base?

Para determinar si el eslabón final está orientado en la dirección del eje Y de la base se utilizó la matriz de rotación R de la matriz homogénea. En la posición q3, la columna correspondiente al eje Z del efector es [0 1 0]T, lo que indica que dicho eje está alineado con el eje Y de la base. Por tanto, en q3 el eslabón final está orientado en la dirección del eje Y de la base.

4. ¿En cuál de las 4 posturas el extremo no se encuentra en el primer cuadrante del sistema de la base?

En la postura $Q4$ el extremo no se encuentra en el primer cuadrante del sistema de la base

5. ¿Qué condición debe cumplirse en la matriz homogénea total del robot para que los ejes del sistema del extremo sean paralelos (sin importar orientación ni orden) a los del sistema de la base?

Que R sea una **matriz de permutación con signos**:

- cada columna y fila tiene exactamente un ±1y el resto **0**,
- (R^T)*R=I y detR=+1.Equivalentemente: los cosenos directores son {0,±1} y los ejes del efector son ejes de {0} reordenados y posiblemente invertidos.

6. ¿Por qué las siguientes matrices no pueden ser resultado de ningún vector de posiciones articulares?

$$T = \begin{bmatrix} 0 & 0.5 & 0.866 & 0.4971 \\ 0 & -0.866 & 0.5 & 0.4971 \\ 1 & 0 & 0 & 0.5250 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0.7030 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 1 & 0 & 0.3946 \\ 0 & 0 & 0 & 0.2947 \\ 1 & 0 & 1 & 0.8103 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 1.4971 \\ -0.7071 & 0 & 0.7071 & 1.4971 \\ 0 & -1 & 0 & 0.5250 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación debe cumplir las siguientes propiedades:

Propiedades

$$\left({}^{M}\mathbf{Rot}_{O}\right)^{T} = \left({}^{M}\mathbf{Rot}_{O}\right)^{-1} = \left({}^{O}\mathbf{Rot}_{M}\right)$$
$${}^{M}\mathbf{Rot}_{O} \times {}^{O}\mathbf{Rot}_{M} = I$$
$$\det(\mathbf{Rot}) = \pm 1$$

Estas propiedades aseguran que la matriz de rotación debe ser ortonormal. Esto significa que sus vectores columna (o fila) deben cumplir con dos condiciones: deben ser vectores unitarios (la magnitud de cada vector debe ser 1), y deben ser ortogonales entre sí, es decir, el producto punto de cualquier par de vectores distintos debe ser 0. Además el determinante debe ser 1; entonces:

1. submatriz de rotación correcta matemáticamente, pero la orientación resultante implica que el eje X del efector quede alineado con el eje Z de la base. Dicha alineación requiere una rotación de 90° alrededor del eje Y de la base, que el Paint Mate 200iA no puede realizar dentro de sus límites articulares.

2. El eje Z del efector está alineado con el eje X de la base(orientación posible) Además, la posición indicada 0.703 se encuentra dentro del rango máximo del robot. El problema se debe al usar ambas condiciones al mismo tiempo ya que el efector no puede adoptar esa orientación y, a la vez, llegue exactamente a ese punto.

3. La matriz no es ortonormal, ya que el producto escalar de sus columnas 1 y 3 es 1. Además su determinante es 0. Por lo que no cumple las propiedades

4.La traslación solicitada es excesiva, el vector de posición tiene un módulo mucho mayor que el alcance máximo del Paint Mate 200iA. Está fuera de su espacio de trabajo.

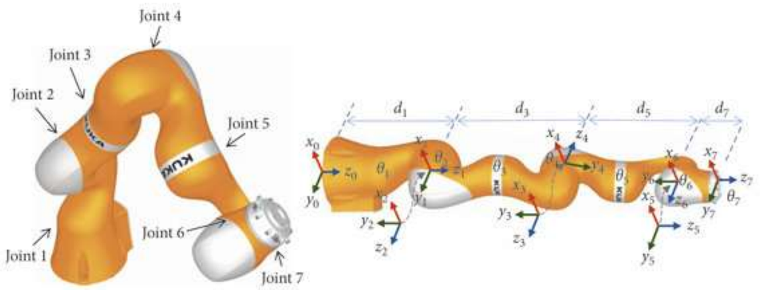
Ejercicio 4

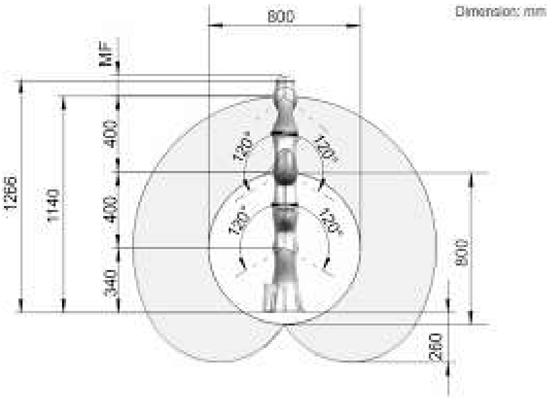
Calcule el máximo error de posición cartesiana en el extremo final que podría tener el LBR iiwa 7 R800 (KUKA). Para hacerlo, asuma que está en una postura totalmente vertical y extendida, y que todas sus articulaciones tienen un error de posición (delta de posición) de 0,1º.

1. ¿Cuál es el error en esta situación?

Para calcular el error vamos hacer una simplificacion del problema y un analisis del peor caso. Cada articulacion tiene un error de 0,1°.Vamos a suponer que la orientacion no se considera un error de desplazamiento. Por lo que en el robot LBR iiwa 7 R800 (KUKA) en posicion completamente vertical hay dos rotaciones sobre el propio eje del robot, por lo que estas no son consideradas al tener el cuenta el modulo del error de posicion.

Luego el resto de errores se calcula utilizando la longitud de la articulacion y el error angular, de esta forma obtenemos el arco que forma y su desplazamiento tanto horizontal como vertical del punto que se espera.





El desplazamiento lateral que tendra resulta de sumar:

$a = d_3 * \text{sen}(\alpha) + d_5 * \text{sen}(2 * \alpha) + d_7 * \text{sen}(3 * \alpha)$

$a = 400 * \text{sen}(0,1°) + 400 * \text{sen}(2 * 0,1°) + 126 * \text{sen}(3 * 0,1°)$

$a = 2,75\text{mm}$

El desplazamiento vertical que tendra resulta de sumar:

$b = \ d_3 + d_5 + d_7 - (d_3 * \cos(\alpha) + d_5 * \cos(2 * \alpha) + d_7 * \cos(3 * \alpha))$

$b = 400 + 400 + 126 - (400 * \cos(0,1°) + 400 * \cos(2 * 0,1°) + 126 * \cos(3 * 0,1°))$

$b = 0,0047\text{mm}$

Por el modulo del error de posicion será:

$e = \sqrt{a^2 + b^2}$

$e = \sqrt{2,75^2 + 0,0047^2}$

$e = 2,75\text{mm}$

2. ¿El error es el mismo para cualquier posición? (suponiendo constante el error de posición articular).

No, el error depende fuertemente de la postura en la que este el robot.

3. ¿La respuesta anterior es válida para todo tipo de robots serie?

Si, el razonamiento es el mismo y el calculo tambien.

Ejercicio TF (obligatorio)

Trabaje con el robot (o los robots) seleccionado para el Trabajo Final de la materia y desarrolle lo siguiente:

1. Cree un archivo de tipo función denominado "esp_trab.m". Este archivo deberá realizar al menos 2 gráficas del espacio de trabajo del robot (figuras 2D en 2 vistas representativas). No deberá contener la definición del robot en sí, para eso se debe ejecutar el archivo "robot.m" del TP3 en su primera línea.

1.1 Una forma de hacerlo es determinando las posiciones extremas del robot (extensiones máximas y mínimas de acuerdo a los límites articulares), calcular el punto correspondiente en el espacio cartesiano, y graficar dichos puntos. Luego, puede unir esos puntos con rectas, dejando áreas representativas aproximadas.

1.2 Otra forma es unir los puntos anteriores con curvas interpoladas dadas de por "jtraj" (función presentada en la unidad 5).

1.3 Puede ayudarse de la función SerailLin.teach() para visualizar y probar posiciones antes de graficar.

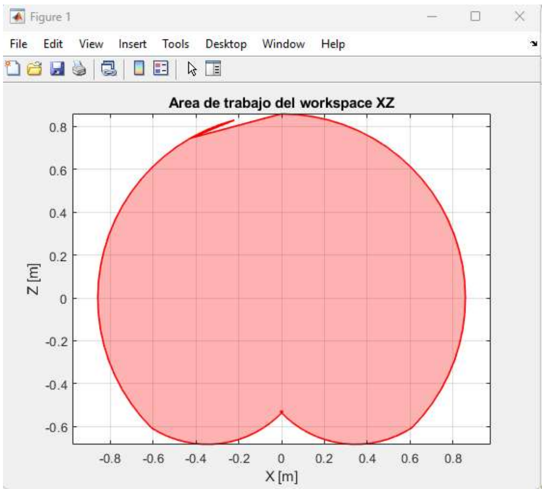
1.4 Recuerde que con los comandos "hold on" y "hold off" de Matlab, puede activar y desactivar el "guardado" de los gráficos anteriores en un Figure.

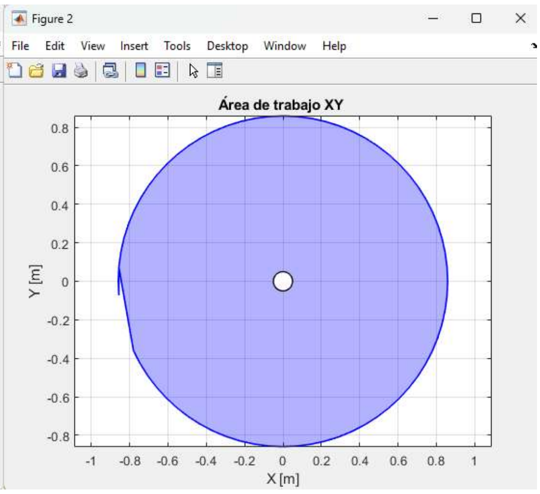
2. Verifique que el espacio de trabajo determinado sea adecuado para la aplicación propuesta. Si la tarea no es definida (ejemplo: "soldadura por arco"), indique aproximadamente qué tareas podría realizar (ejemplo: "arcos lineales de 800mm como máximo y arcos circulares de hasta 300mm de radio"). Tenga en cuenta que las tareas pueden requerir orientación.

Segun nuestra area de trabajo, el robot puede escanear de forma completa objetos pequeños y medianos (como ser piezas mecanicas, moldes, etc.) del orden de los 100 a 400 mm.

El área de trabajo que definimos (en planos XZ y XY) cubre un volumen útil de aproximadamente:

- **Alcance lineal**: hasta unos 1600 mm en X (máxima extensión del brazo).
- **Altura útil**: unos 1200 mm en Z.
- **Cobertura angular**: completa en XY gracias a q1 y q6.





Eso significa que el efector final puede **rodear y cubrir completamente un objeto** de tamaño pequeño/mediano colocado en una mesa o soporte.

Si se trata de objetos grandes, puede escanearlos pero no en su totalidad, sino que cubre un area del objeto y luego habria que rotar el mismo para cubrir otro area, por ejemplo un auto.

Para escaneo con un **láser o sonda óptica**, lo importante no es solo el alcance, sino la posibilidad de **colocar el sensor en diferentes ángulos** respecto a la superficie.

El espacio de trabajo obtenido permite acercar el efector desde arriba, desde los costados, e incluso con cierta inclinación, lo es suficiente para cubrir un objeto sin puntos ciegos, siempre que se coloque en el centro del área accesible.

Por lo tanto, el workspace definido asegura la viabilidad de la aplicación propuesta de escaneo con un Scan Arm

3. Defina matrices de “base” y de “tool” acordes a su aplicación en el archivo “robot.m”.

Realizado en el archivo robot.m