

Proyecto Final Grupo 11: Scan Arm.

Estudio de brazo robot de 6 GDL basado en el robot "Quantum Max" de Faro.



CARRERA Ingeniería Mecatrónica

SEMESTRE Segundo Semestre 2025

CÁTEDRA Robótica I

ALUMNOS Barrios Retta Federico , 14101
Campo Camila , 13667
Giménez Mariano , 13853
Salinas Eduardo , 13900
Videla Jeronimo , 14233

FECHA 03 de noviembre de 2025

Contents

1	Abstract	3
1.1	Resumen (ES)	3
1.2	Abstract (EN)	3
2	Introducción	4
3	Tareas a resolver y robot seleccionado	4
3.1	Problema a resolver	4
3.2	Robot seleccionado y pertinencia	4
3.3	Conveniencia frente a alternativas	5
3.4	Alcance verificable del proyecto	6
3.5	Nota sobre las demostraciones	6
4	Información del Scan Arm	6
4.1	Concepto y principio de operación	7
4.2	Arquitectura mecánica y GDL	7
4.3	Sensado, referencia y herramientas	7
4.4	Definición y tipos de articulaciones	7
4.5	Modelado geométrico (DH)	8
4.6	Workspace, ergonomía y límites	8
	WORKSPACE	9
4.7	WORKSPACE CON FS	10
5	Cinematica Directa	10
	Método de solución	11
6	Cinematica Inversa	11
7	Relacion de velocidades (Jacobiano)	12
	Parte lineal (J_v)	13
	Parte angular (J_ω)	13
	Determinante del Jacobiano	13
	Puntos singulares	14
8	Planificación y generación de trayectoria	14
8.1	Precisión y repetibilidad: qué afecta y cómo se cuida	15
8.2	Modos de uso en el proyecto	15
8.3	Muestreo de una placa de dimensiones $H \times W \times e$	15
	Línea de Escaneo:	15
	Transición (U-Turn):	16
8.4	Gráficos obtenidos	16
9	Sensores y actuadores	20
9.1	Sensores	20
	Sensores Internos (Estado Articular y Cinemática)	20
	Sensores Externos (Adquisición de Tarea)	21
9.2	“Actuadores” en brazos pasivos	21
9.3	Variante cooperativa actuada (enfoque académico)	21
9.4	Selección concreta para el proyecto	22

10	Conclusiones	22
11	Referencias	23
12	Anexos	24

1 Abstract

1.1 Resumen (ES)

Palabras clave: cinemática directa e inversa, DH estándar, jacobiano, singularidades, LSPB, tpoly, trayectoria cartesiana, ScanArm, MATLAB. ([GitHub](#))

Este proyecto presenta el modelado, análisis y simulación de un brazo articulado tipo **ScanArm** de **6 grados de libertad** parametrizado mediante **Denavit–Hartenberg estándar** (unidades: **m**). A partir de la tabla DH propuesta, se desarrolla la **cinemática directa** para reconstruir la pose del efector y la **cinemática inversa** para planificar movimientos hacia poses objetivo. Se estudia la **relación velocidad-articular** mediante el **jacobiano**, identificando **singularidades** y zonas de bajo condicionamiento relevantes para tareas de medición. En planificación, se implementan trayectorias **P2P** con perfiles **LSPB** y transiciones **quintic (tpoly)**, y trayectorias **cartesianas en SE(3)** para barridos de escaneo. El trabajo contempla dos escenarios: **uso pasivo** (medición guiada por operario, cinemática directa con encoders) y una **variante cooperativa actuada** para evaluación académica, donde la CI habilita la ejecución automática de vías en un entorno controlado. Se reportan simulaciones del **workspace**, perfiles cinemáticos y animaciones del recorrido, así como criterios de seguridad y límites articulares. El desarrollo se integra en **MATLAB** dentro del repositorio del curso (*Simulation - Final/*), con scripts reproducibles y figuras exportables para el informe. ([GitHub Repository](#))

1.2 Abstract (EN)

This project presents the modeling, analysis, and simulation of a **ScanArm-type 6-DoF** articulated manipulator parameterized with **standard Denavit–Hartenberg** notation (units in **meters**). Based on the proposed DH table, we derive **forward kinematics** to reconstruct the end-effector pose and **inverse kinematics** to plan motions toward target poses. We analyze the **joint-to-task velocity mapping** through the **Jacobian**, highlighting **singularities** and poor-conditioning regions that matter for measurement tasks. For motion planning, we implement **point-to-point** profiles using **LSPB** and **quintic (tpoly)** timing, and **Cartesian SE(3) paths** tailored to scanning sweeps. Two use cases are considered: a **passive** operation (human-guided measurement, forward kinematics via encoders) and a **cooperative actuated** variant for academic evaluation, where IK enables automatic execution of puntos de referencia under safe limits. We report simulations of the **workspace**, joint/Cartesian profiles, and rendered animations, together with safety criteria and joint bounds. The implementation is provided in **MATLAB** within the course repository (*Simulation - Final/*), including reproducible scripts and exportable figures for the final report. ([GitHub Repository](#))

Keywords: forward and inverse kinematics, standard DH, Jacobian, singularities, LSPB, tpoly, Cartesian trajectory, ScanArm, MATLAB. ([GitHub](#))

2 Introducción

La inspección geométrica asistida por brazos articulados de medición ha ganado relevancia en entornos donde se requiere movilidad, alcance y rapidez de despliegue. En este proyecto se modela y analiza un manipulador tipo ScanArm de **6 grados de libertad**, parametrizado mediante la convención **Denavit–Hartenberg estándar** con unidades en **metros**. El objetivo general es establecer una base cinemática sólida para tareas de escaneo y palpado, evaluando el desempeño geométrico del sistema y su capacidad para generar trayectorias que preserven la calidad metrológica.

El trabajo aborda tanto la **cinemática directa** como la **cinemática inversa** del modelo propuesto, junto con el **jacobiano** y el estudio de **singularidades** y condicionamiento. Si bien muchos dispositivos comerciales operan en modo **pasivo** (movidos por el operario y sensados por encoders), aquí se incorpora una **variante cooperativa actuada** con fines académicos: la CI se utiliza para planificar y, de ser pertinente, ejecutar vías cartesianas y articulares bajo límites de posición, velocidad y aceleración.

Para la **planificación de movimiento**, se emplean perfiles **LSPB** y polinomios **quintic** para movimientos punto a punto, y trayectorias **cartesianas en SE(3)** adecuadas a barridos de escaneo. Se analizan criterios de seguridad, límites articulares y regiones mal condicionadas del jacobiano, evitando configuraciones cercanas a singularidades. Las simulaciones incluyen estimación del **espacio de trabajo**, perfiles cinemáticos y animaciones de recorridos representativos.

El desarrollo se implementa en **MATLAB** y se integra en el repositorio del curso (RoboticaI/Simulation - Final), con scripts reproducibles y exportables para documentar resultados.

Este documento se organiza de la siguiente manera: primero se presenta la parametrización DH y la validación de la cinemática directa; luego la cinemática inversa y el análisis de velocidades; a continuación la planificación de trayectorias en espacio articular y cartesiano; se muestran las simulaciones y resultados; y finalmente se discuten conclusiones y líneas de trabajo futuro.

3 Tareas a resolver y robot seleccionado

3.1 Problema a resolver

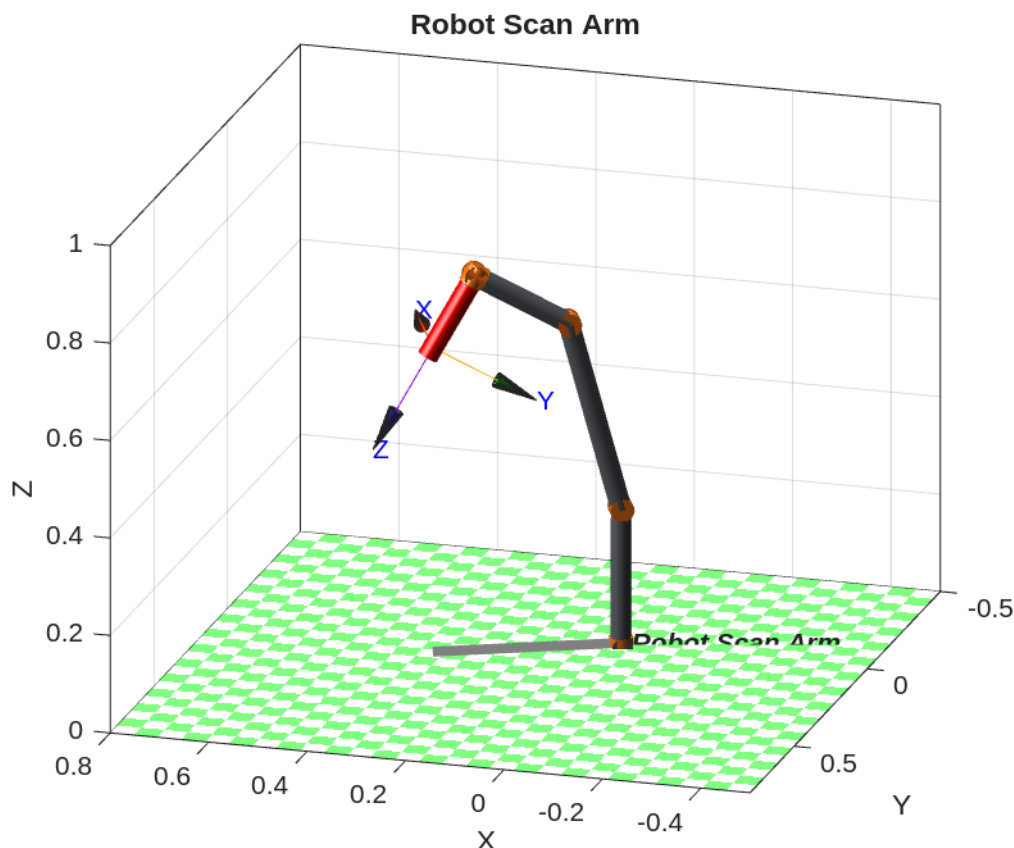
La célula requiere medición y escaneo dimensional con agilidad: piezas cambiantes, sin utillajes específicos, sin una CMM fija y sin depender cada semana de la agenda y el presupuesto de un tercero. El objetivo es disponer de un medio con alcance volumétrico medio sobre mesa, capacidad de orientar la normal del efector para barridos, repetibilidad suficiente para práctica académica e inspección ligera, y un flujo reproducible que permita simular y documentar resultados de forma consistente.

3.2 Robot seleccionado y pertinencia

Se adopta un brazo articulado tipo ScanArm de seis grados de libertad, modelado con la convención Denavit–Hartenberg (**DH**) estándar en metros. Esta arquitectura en serie ofrece un equilibrio favorable entre alcance y compacidad de la celda; los seis ejes permiten posicionar el TCP (**Tool Center Point**) y gobernar su orientación para seguir superficies, condición necesaria en barridos de escaneo. En operación pasiva, los encoders de las juntas y el modelo DH proporcionan la cinemática directa para reconstruir la pose; en la variante cooperativa actuada,

la cinemática inversa habilita la ejecución de vías planificadas y el análisis de límites, singularidades y perfiles de movimiento.

La planificación articular y cartesiana con trayectoria con perfil de velocidad trapezoidal (**LSPB**) y polinomios de quinto orden (**quintic**) evita sacudidas y respeta límites de posición, velocidad y aceleración, preservando la calidad de la medición.



3.3 Conveniencia frente a alternativas

Frente a la contratación de un metrólogo externo, esta solución reduce la dependencia de agenda y el coste recurrente, a la vez que internaliza el conocimiento y garantiza la reproducibilidad mediante scripts y simulación previa.

Una CMM puente superaría la precisión, pero su coste, huella espacial y falta de portabilidad la vuelven desproporcionada para una célula didáctica o procesos ágiles.

Los instrumentos manuales (calibres, plantillas) son económicos y veloces, aunque poco versátiles y sin capacidad para capturar geometría 3D completa.

Un robot industrial estándar aporta potencia y automatización, pero exige integración de seguridad y suele ser excesivo en costo y espacio para inspección flexible.

El 6R tipo ScanArm, en cambio, es portátil, versátil, orientable y totalmente guionable, además de alinearse con los criterios de la cátedra (DH, CD/CI, jacobiano, trayectorias y simulación). Si se desea avanzar a control cooperativo, la transición es natural sin modificar el modelo geométrico.

Resumiendo las características de cada uno:

Criterio	Metrólogo	CMM fija/puente	Calibres/Plantillas	Robot industrial	Faro ScanArm
Costo recurrente	—	Excelente precisión	Rápidos según quien los use	Potente	Portátil
Dependencia de disponibilidad	—	Mala portabilidad	Baratos	Requiere integración de seguridad	Versátil
Transferencia de <i>know-how</i>	Poca transferencia a la empresa	Sobredimensionado para la célula didáctica	Bueno movimiento versátiles	Alto costo relativo a la aplicación	Didáctico
Reproducibilidad del flujo	Cero control sobre reproducibilidad del <i>pipeline</i> y la simulación previa	—	Producción ágil	Plano de diseño complejo	Se puede programar

3.4 Alcance verificable del proyecto

El trabajo demostrará, con base en el repositorio, la definición y justificación de la tabla DH; la cinemática directa validada en posturas de prueba; la cinemática inversa numérica para planificar poses objetivo; el jacobiano con análisis de manipulabilidad y singularidades aplicadas al escaneo; la planificación de trayectorias articulares y cartesianas con temporización LSPB y quintic; y simulaciones del espacio de trabajo, perfiles cinemáticos y animaciones. Se incorporan criterios de seguridad y límites de juntas en todos los casos.

3.5 Nota sobre las demostraciones

Las validaciones se realizarán mediante trayectorias simples y reproducibles: movimientos punto a punto entre referencias, un arco manteniendo la normal del TCP y un raster lineal sobre un plano virtual. En modo pasivo, estas trayectorias sirven como guía para el operario; en modo cooperativo actuado, se ejecutan con cinemática inversa respetando los límites definidos.

4 Información del Scan Arm

El Scan Arm considerado es un manipulador 6R concebido para **medición y escaneo** guiados por el usuario (modo pasivo) y, en nuestra variante académica, también para **planificación/ejecución asistida** (modo cooperativo). Su mérito no es mover cargas paletizadas ni soldar chasis; es **posicionar** con libertad la punta y **orientarla** con precisión sobre superficies, requiriendo un costo financiero y espacial menor al de una CMM.

4.1 Concepto y principio de operación

El brazo trabaja como **AACMM** (*Articulated Arm Coordinate Measuring Machine* o *Máquina de Medición por Coordenadas de brazo articulado*): una cadena cinemática en serie con **encoders** y **motores** en cada junta. En modo pasivo, el operario desplaza el brazo y el sistema reconstruye la pose del TCP por **cinemática directa**, es decir con los motores apagados. Pero en nuestra aplicación utilizamos el modo activo, en la que los motores están encendidos.

Para el proyecto adoptamos **DH estándar** en **metros** y demostramos, además, cinemática inversa para planificar vías que el operario puede seguir o que un sistema actuado podría ejecutar.

4.2 Arquitectura mecánica y GDL

Se modela una estructura 6 GDL: tres ejes “de brazo” para posicionamiento grueso y tres “de muñeca” para orientar la herramienta. Esta partición permite mantener la normal del sensor durante barridos, que es donde este tipo de brazo brilla. Un **7.º eje** es habitual cuando se agrega un **escáner láser** o se requieren orientaciones más ricas sin sacrificar ergonomía, pero el alcance del trabajo integrador queda cubierto con 6.

4.3 Sensado, referencia y herramientas

Cada articulación reporta posición angular; el marco de referencia está en la base. El **TCP** puede ser un **palpador táctil** o el **soporte de una cabeza de escaneo**. La precisión práctica depende de tres patas:

- (i) **calibración geométrica** del modelo
- (ii) **resolución/ruido** de encoders
- (iii) **rigidez** del conjunto.

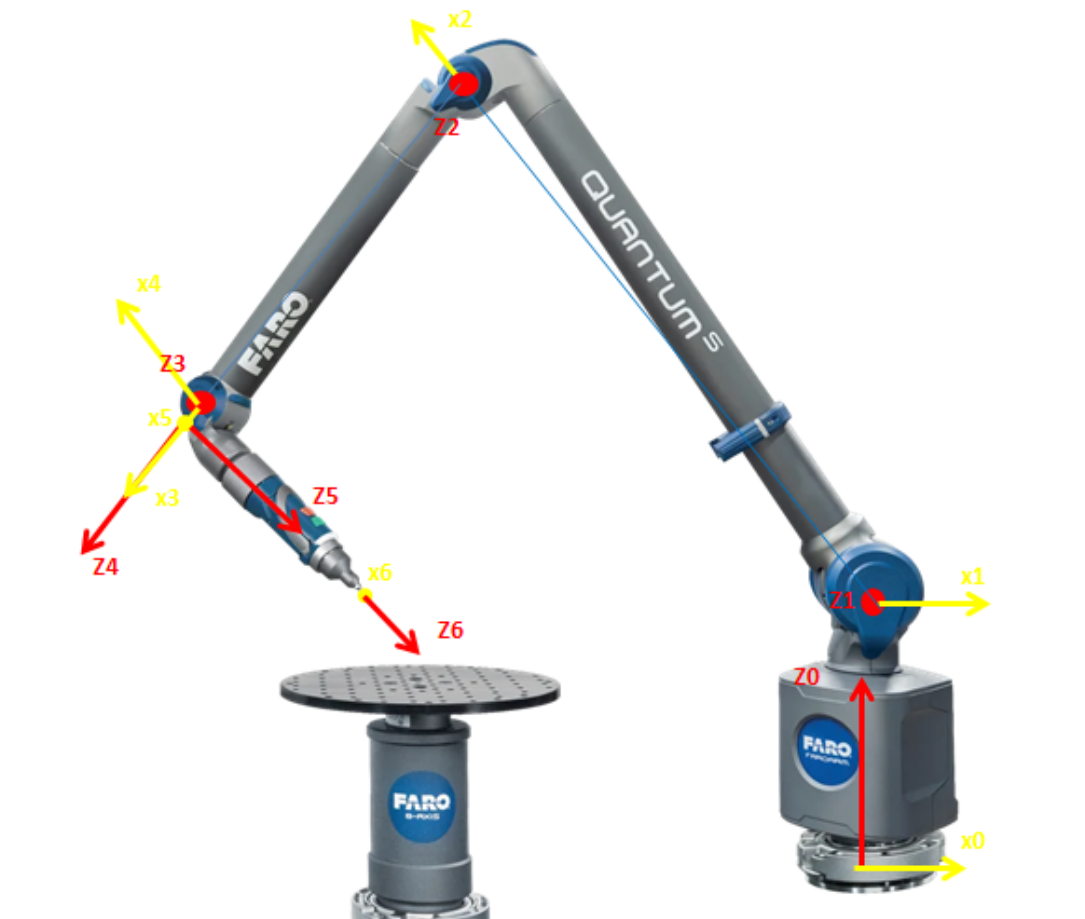
Por eso los flujos serios incluyen **compensación de sonda** y pruebas periódicas tipo **esfera/artefacto** para verificar que la CD refleje la realidad. Esta calibración repetitiva es lo que permite asegurar la precisión y validez de las mediciones realizadas, para evitar corrimientos de los motores y articulaciones de sus posiciones esperadas.

4.4 Definición y tipos de articulaciones

Juntura	Tipo	Movimiento	Eje	Notas
J1 (Base)	Revolución (R)	Yaw (rotación alrededor de z)	z_0	Base: orientación de toda la cadena.
J2 (Hombro)	Revolución (R)	Pitch (rotación alrededor de y)	y_1	Primer eje de brazo para posicionamiento grueso.
J3 (Codo)	Revolución (R)	Yaw (rotación alrededor de z)	z_2	Segundo eje de brazo; define alcance del TCP.
J4 (Muñeca 1)	Revolución (R)	Yaw (rotación alrededor de z)	z_3	Primer eje de muñeca para orientar herramienta.
J5 (Muñeca 2)	Revolución (R)	Roll (rotación alrededor de x)	x_4	Segundo eje de muñeca; controla inclinación del TCP.
J6 (Muñeca 3)	Revolución (R)	Yaw (rotación alrededor de z)	z_5	Eje de orientación fina del efector.

4.5 Modelado geométrico (DH)

El brazo se parametriza con la tabla DH provista (convención estándar). El orden de ejes y los signos de α_i se fijan siguiendo la regla clásica: z_i alineado al eje de la junta (i), x_i hacia el común perpendicular, y α_i como el giro de x_i a x_{i+1} . Con esto se obtienen las matrices homogéneas A_i y la pose 0T_6 . La **CD** valida la geometría; la **CI** se usa para planear movimientos hacia poses objetivo sin caer en configuraciones mal condicionadas.



θ	d	a	α
q_1	0.283	0	$-\frac{\pi}{2}$
q_2	0	0.398	0
q_3	0	0.213	0
q_4	0	0	$\frac{\pi}{2}$
q_5	0	0	$\frac{\pi}{2}$
q_6	0.166	0	0

Nota: Si bien el robot del proyecto esta basado en el 'Quantum Max ScanArm' de Faro, el cuál NO posee muñeca esférica (el que se encuentra en la imagen de arriba), el robot propuesto en este proyecto (robot Scan Arm G11) SI posee muñeca esférica.

4.6 Workspace, ergonomía y límites

El **espacio de trabajo** se estima barriendo **q** dentro de los **límites articulares**, y muestra el espacio en el cual el robot es capaz de trabajar. Este se descompone para mas facil visualizacion

en sus planos XZ (Rojo) y XY (azul).

Los límites de los parámetros articulares, que definen nuestro workspace son:

Límite	q_1	q_2	q_3	q_4	q_5	q_6
Superior	170	80	140	140	120	360
Inferior	-170	-150	-110	-140	-120	-360

Debido a la naturaleza no actuada del robot elegido, los límites articulares no son provistos por el fabricante, y no se encuentran en su datasheet. Por lo tanto, se eligieron límites articulares conservativos, que permitan recrear los movimientos del robot con fidelidad y sin movimientos inesperados.

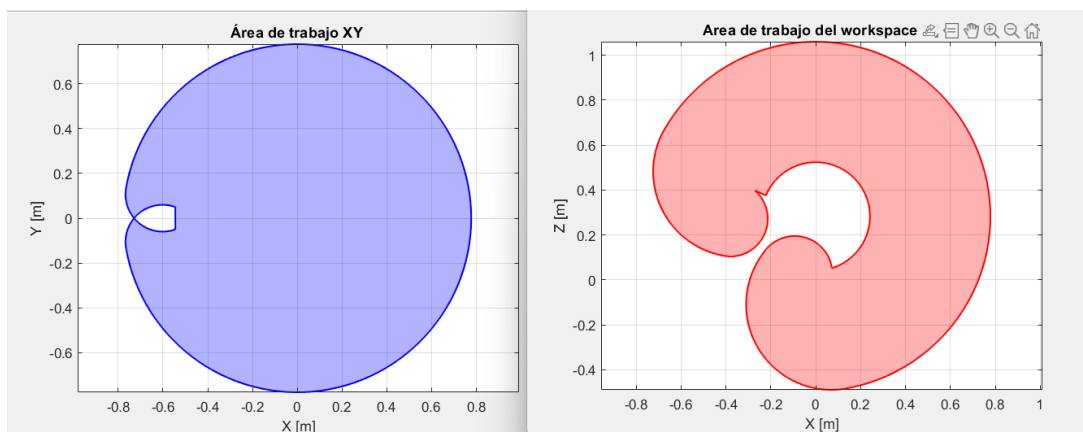
La diferencia entre ambas imágenes se deben a la forma en la que se hace volver al efector.

Si se observa la vista superior XY, se puede notar que es capaz de alcanzar prácticamente todos los puntos a su alrededor, menos en la dirección opuesta al marco de referencia de base, por esto no nos genera problemas ya que va a ser preferible trabajar en el frente de este.

Por otro lado, si se observa la vista lateral XZ, vemos que el brazo es capaz de alcanzar su propia base, puntos sobre y detrás de sí mismo. Esto puede generar problemas si no se tiene cuidado con las trayectorias, ya que una mala combinación de instrucciones articulares podrían causar un choque del robot con su base. En la proyección XZ, el alcance real del robot es mayor que el mostrado, pero se realizó una reducción para mostrar su espacio de trabajo óptimo y útil.

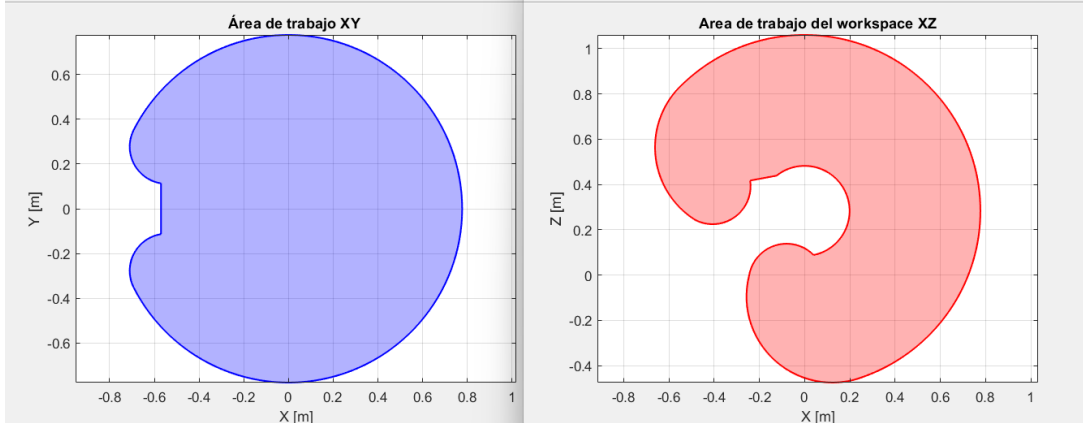
Analizando las dimensiones, vemos que tiene un alcance máximo de 0.8m, y una distancia mínima ideal de 0.2m a altura 0.2m. Además, si se busca trabajar de forma segura, es recomendable no colocarse cerca de los límites del espacio, por lo que será necesario trabajar con piezas de dimensiones que no excedan los 0.5m sobre el plano XZ.

WORKSPACE



Si se busca evitar forzar las articulaciones y los motores, se puede trabajar con factor de seguridad, tal que el valor máximo alcanzado por las articulaciones no supere un cierto porcentaje del límite, asegurando que siempre se tenga un margen de movilidad segura. Utilizando un FS de 90%, se obtiene el nuevo workspace

4.7 WORKSPACE CON FS



En la sección del Jacobiano se abordarán las zonas donde se debe evitar trabajar para evitar problemas con la cinemática inversa.

5 Cinemática Directa

A partir de la parametrización obtenida mediante el método **Denavit–Hartenberg**, se desarrolló el cálculo de la **cinemática directa** para el robot **ScanArm 6R**, utilizando tanto formulaciones teóricas como herramientas computacionales.

El objetivo fue determinar la posición y orientación del efector final respecto al sistema base a partir de las variables articulares del robot, representadas por el vector:

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]$$

Mediante el producto de las matrices homogéneas de transformación $({}^{i-1}T_i)$, se obtiene la matriz total:

$${}^0T_6 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6$$

Esta matriz describe completamente la **pose del gripper** respecto al sistema base, permitiendo conocer en todo momento la posición y orientación del efector dentro del entorno de trabajo. Para eso se usó la función `fkine(q)` del toolbox de Peter Corke, con eso y con la matriz DH podemos obtener $({}^0T_6)$

En nuestro caso, la cinemática directa se utilizó principalmente para determinar la posición inicial del gripper y su orientación en distintas configuraciones, lo que permitió validar que el modelo DH adoptado era coherente con la geometría real del brazo.

A través de este análisis, fue posible ubicar con precisión el efector final en el espacio de trabajo y comprobar que el robot podía alcanzar los puntos requeridos sin violar los límites articulares ni las restricciones físicas del modelo.

Además, la cinemática directa nos permitió **determinar el espacio de trabajo efectivo** del ScanArm, ayudando a definir las zonas de alcance y los límites de movimiento de cada articulación. Este resultado fue esencial para el diseño del **layout de la celda**, ya que permitió posicionar el robot de forma que optimizara su rango operativo y evitara interferencias con otros elementos del entorno.

La cinemática directa además es fundamental para el funcionamiento del robot en modo cooperativo, ya que en este modo se miden las posiciones articulares y el punto de medición es determinado por la cinemática directa.

Por último, la validación del modelo se realizó comparando los resultados teóricos obtenidos mediante el algoritmo DH con las posiciones estimadas en el modelo 3D del robot.

Las diferencias observadas fueron mínimas, confirmando la correcta definición de los parámetros y el buen ajuste entre el modelo matemático y la estructura física.

De esta forma, se garantizó que las trayectorias generadas posteriormente se desarrollaran dentro de los límites seguros del sistema y del espacio disponible en la celda de trabajo.

Método de solución

1. Tabla DH

Se utiliza la tabla DH definida en el punto (i), con los parámetros $(a_i, \alpha_i, d_i, \theta_i)$.

Para las articulaciones revolutas, el parámetro variable es el ángulo $(\theta_i = q_i + \theta_{i0})$.

2. Transformación elemental

Cada matriz ${}^{i-1}T_i$ se construye según:

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Composición de transformaciones

La matriz total $({}^0T_6)$ se obtiene multiplicando secuencialmente las seis matrices elementales.

4. Obtención de la pose

- **Posición:** coordenadas del efector $(x, y, z) = \text{última columna de } ({}^0T_6)$.
- **Orientación:** matriz de rotación $(R = {}^0R_6)$.

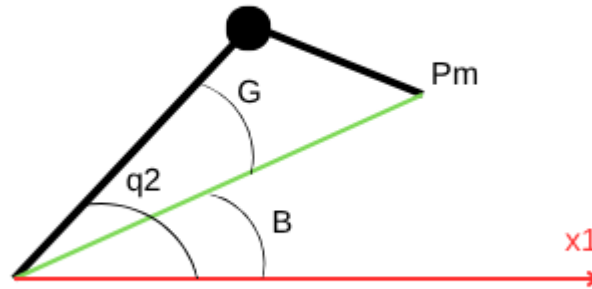
6 Cinemática Inversa

Para el cálculo de la cinemática inversa de nuestro robot utilizamos la función de cinemática inversa `cin_inv_Faro` que definimos en el trabajo práctico N° 5B, que utiliza el cálculo analítico para encontrar los q_1, q_2, q_3, q_4, q_5 y q_6 para llegar al punto inicial de la trayectoria y luego a los siguientes puntos de la misma.

Es importante destacar que pudimos realizar el desacople de la muñeca del resto del robot por el hecho de haber considerado que nuestro robot es del tipo muñeca esférica, sin esta simplificación no podríamos haber utilizado este método.

Llamaremos p al punto al que queremos llegar, p_m al punto del sistema de la muñeca, en un principio referenciados al sistema S_0 (base de nuestro robot)

- Para calcular q_1 : como rota el brazo entero, y sabemos que la muñeca se debe alinear con el punto p , entonces basta con calcular $q_1 = \tan\left(\frac{p_{my}}{p_{mx}}\right)$ (2 soluciones)
- Para calcular q_2 nos trasladamos el sistema a S_1 y referenciamos todo al mismo, entonces para cada q_1 , donde se puede calcular 2 valores diferentes de q_2 (codo arriba y codo abajo):



de donde calculamos r , G , B y $q_2 = B \pm G$

- Para el calculo de q_3 volvemos a areferenciarse todo al sistema S_2 y calculamos q_3 con las proyecciones de p_m
- Para el calculo de q_4 , q_5 y q_6 se construye la matriz de transformacion directa 0T_6 y obtenemos la transformada relativa muñeca 3T_6 necesaria para alcanzar T, como la muñeca solo nos aporta la orientacion, usamos solo la parte \mathbb{R} de 3T_6 , y calculamos q_4 , q_5 y q_6 . Para verificar que el robot llegue al punto indicado con los q calculados usamos la funcion `fkine` para verificar que llegamos a la matriz T del punto, o simplemente comparamos que los q coincidan. Depende de que parametro de entrada le demos a la funcion.

Ademas, en el calculo de la cinemtica inversa, nuestro robot llega al punto indicado, pero el q_2 esta 2π adelantado, esto es debido al calculo trigonometrico ya que las soluciones matematicas son infinitas, por lo que tuvimos que restarle 2π , para asegurarnos de que los valores de q_2 esten dentro de los limites articulares.

7 Relacion de velocidades (Jacobiano)

El Jacobiano es una transformación matemática que permite relacionar las velocidades en el espacio articular con las velocidades lineales y angulares en el espacio cartesiano. Además, nos permite evaluar puntos donde el robot puede fallar hallando los puntos singulares, aquellos donde el determinante del Jacobiano se vuelven 0. Ademas, si se conocen las fuerzas aplicadas sobre el efector final, nos permite encontrar los torques aplicados sobre las articulaciones, aunque esto ultimo no sera desarrollado en detalle en este informe. Esto se representa como:

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \omega \end{bmatrix} = J(q) \dot{\mathbf{q}} \quad \text{y} \quad \tau = J(q)^T \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}$$

Siendo que nuestro robot posee 6 articulaciones, y ademas 6 grados de libertad, su Jacobiano tiene dimension 6×6 . Este lo descompondremos en 2 partes, una de velocidades lineales (J_v), y otra de velocidades angulares (J_ω), ambas de 3×6 , tal que el Jacobiano completo resulte:

$$J(q) = \begin{bmatrix} J_v^{(\text{brazo})} & J_v^{(\text{muñeca})} \\ J_\omega^{(\text{brazo})} & J_\omega^{(\text{muñeca})} \end{bmatrix}, \quad J_v, J_\omega \in \mathbb{R}^{3 \times 6}.$$

Utilizaremos ademas las siguientes abreviaciones:

$$\begin{aligned} s_i &= \sin(q_i), & c_i &= \cos(q_i), & \text{para } i &= 1, \dots, 6, \\ s_{23} &= \sin(q_2 + q_3), & c_{23} &= \cos(q_2 + q_3), \\ s_{234} &= \sin(q_2 + q_3 + q_4), & c_{234} &= \cos(q_2 + q_3 + q_4). \end{aligned}$$

Parte lineal (J_v)

$$J_v^{(\text{brazo})} = \begin{bmatrix} -a_2 s_1 - a_3 \sin(q_1 + q_2) & c_1 d_6 \cos q_5 s_{234} & c_1 d_6 \cos q_5 s_{234} \\ a_2 c_1 + a_3 \cos(q_1 + q_2) & s_1 d_6 \cos q_5 s_{234} & s_1 d_6 \cos q_5 s_{234} \\ 0 & a_2 + a_3 \cos q_2 + d_6 c_{234} \cos q_5 & a_3 \cos q_2 + d_6 c_{234} \cos q_5 \end{bmatrix},$$

$$J_v^{(\text{muñeca})} = \begin{bmatrix} c_1 d_6 \cos q_5 s_{234} & -d_6 (s_1 \cos q_5 + c_1 c_{234} \sin q_5) & 0 \\ s_1 d_6 \cos q_5 s_{234} & -d_6 (s_1 c_{234} \sin q_5 - c_1 \cos q_5) & 0 \\ d_6 c_{234} \cos q_5 & d_6 s_{234} \sin q_5 & 0 \end{bmatrix}.$$

Parte angular (J_ω)

$$J_\omega^{(\text{brazo})} = \begin{bmatrix} 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}, \quad J_\omega^{(\text{muñeca})} = \begin{bmatrix} s_1 & c_1 s_{234} & -s_1 \sin q_5 + c_1 c_{234} \cos q_5 \\ -c_1 & s_1 s_{234} & s_1 c_{234} \cos q_5 + c_1 \sin q_5 \\ 0 & c_{234} & -\cos q_5 s_{234} \end{bmatrix}.$$

Determinante del Jacobiano

Aprovechando el desacople por la muñeca esférica ($J = J_{\text{brazo}} \oplus J_{\text{muñeca}}$):

$$\det(J) = \det(J_v^{(\text{brazo})}) \det(J_\omega^{(\text{muñeca})})$$

y, para una muñeca esférica de tres ejes concurrentes:

$$\det(J_\omega^{(\text{muñeca})}) = \sin(q_5)$$

Debido al offset de 90° aplicado a q_5 , $\sin(q_5)$ pasa a ser $\cos(q_5)$ por identidad trigonométrica

Por lo tanto, el determinante total del Jacobiano geométrico del efector final es:

$$\det(J_{\text{ef}}) = \cos(q_5) \det(J_v^{(\text{brazo})})$$

donde $\det(J_v^{(\text{brazo})})$ depende sólo de la geometría del brazo. La fórmula final del determinante resulta:

$$\det J(q) = \cos(q_5) \left[\underbrace{0.033740052}_{\frac{8435013}{250000000}} \cos q_2 \sin q_3 - \underbrace{0.018056862}_{\frac{9028431}{500000000}} \sin q_2 + \underbrace{0.018056862}_{\frac{9028431}{500000000}} \cos q_3 \sin q_2 + \underbrace{0.018056862}_{\frac{9028431}{500000000}} \right]$$

Y de forma simplificada:

$$\det J(q) = \cos(q_5) \left[\cos q_2 \sin q_3 \left(0.033740052 + 0.018056862 \cos q_3 \right) + 0.018056862 \sin q_2 (\cos q_3 - 1) \right].$$

Donde podemos observar que q_1 no tiene influencia sobre el determinante del brazo, solo q_2 , q_3 , q_5 y las dimensiones del robot.

Puntos singulares

Desarrollando las formulas, llegamos a:

$$\det J_{ef}(q) = 0 \iff \begin{cases} \sin(q_3) = 0 \\ \cos(q_5) = 0 \\ O_c = (0,0,Z); \text{ (Muñeca sobre base) (Caso 3)} \end{cases} \Rightarrow \begin{matrix} q_3 = 0, \pi & \text{(Codo estirado/plegado)} \\ q_5 = \frac{\pi}{2}, \frac{3\pi}{2} & \text{(Muñeca alineada) (Caso 2)} \end{matrix}$$

Observamos que hay 3 casos donde $\det(J)$ se vuelve 0, los cuales definen cada uno un tipo de singularidad que se debe evitar.

El primer caso sucede cuando el brazo se encuentra completamente estirado (o flexionado, pero debido a los limites articulares esto no es posible). Su solución es sencilla: reducir ligeramente el área de trabajo, lo cual además evita crear torques excesivos sobre el brazo.

El segundo caso sucede cuando la muñeca del robot se ubica a 90° respecto del antebrazo de este. Causa que los ejes Z de q_4 y q_6 se alineen, lo que elimina un grado de libertad. Esto se puede solucionar trabajando con unos limites articulares más reducidos, y evitando doblar la muñeca hacia "adentro".

El tercer y ultimo caso es causado cuando la muñeca (O_c) se ubica sobre el eje Z de la base (coordenadas 0,0). Esto depende de una combinación de valores de los parametros articulares q_2 y q_3 , por lo que no hay una única limitación articular que lo solucione. Pero, se puede implementar una verificación en el espacio cartesiano que marque a la posición como no válida. Además esto se soluciona facilmente al ver que el espacio de trabajo no incluirá tal zona, ya que implicaría trabajar sobre el robot o directamente en colisión con este.

8 Planificacion y generacion de trayectoria

La planificación de trayectorias en robótica parte de tres piezas conceptuales: **la tarea, la representación geométrica y las restricciones**. La tarea define qué debe hacer el efector (posicionar, seguir una curva, inspeccionar una zona, evitar un volumen). La representación geométrica especifica dónde debe hacerlo: un camino en el espacio articular q o en el espacio de tarea $SE(3)$ mediante poses objetivo y tolerancias. Las restricciones delimitan lo físicamente ejecutable: cotas de junta (q_{\min}, q_{\max}), límites de velocidad y aceleración ($\dot{q}_{\max}, \ddot{q}_{\max}$), restricciones de orientación, zonas prohibidas y márgenes de seguridad.

Conviene distinguir **camino** y **trayectoria**. El camino es puramente geométrico (una curva en q o en $SE(3)$ sin noción de tiempo). La trayectoria es el camino **temporalizado** por una ley $t \mapsto s(t)$ que fija posición, velocidad y aceleración en cada instante. En práctica, se propone primero un camino que cumpla la tarea y evite colisiones y, luego, se aplica una **parametrización temporal** que respete límites y suavidad. Esta temporalización puede ser polinómica (quintic con condiciones en extremos), tipo LSPB (trapezoidal), S-curve con acotación de jerk, o cualquier ley que garantice continuidad al orden requerido por el actuador y el sensor.

Si el camino se define en espacio de tarea, cada muestra $T(s)$ se proyecta a juntas resolviendo cinemática inversa con semillas continuas para preservar suavidad y minimizar saltos. La factibilidad cinemática se contrasta evaluando el jacobiano $J(q)$: la relación $\dot{x} = J(q) \dot{q}$ permite estimar \dot{q} inducida por la velocidad deseada en tarea y detectar saturaciones antes de ejecutar. El condicionamiento de J guía el **evitado de singularidades** y las re-orientaciones del efector cuando correspondan. Si el camino se define directamente en espacio articular, la verificación es análoga pero sin resolver IK, y el control de colisiones/alcanzabilidad se hace mapeando a $(SE(3))$ mediante cinemática directa.

8.1 Precisión y repetibilidad: qué afecta y cómo se cuida

Las fuentes de error típicas incluyen **offsets geométricos**, **backlash**, **compliance**, **deriva térmica** y el propio **ruido de medida**. En análisis, el **jacobiano** permite entender cómo pequeños errores articulares se amplifican en el TCP. En práctica, se atenúa con buena calibración, perfiles de movimiento suaves (LSPB/quintic) y evitando regiones cercanas a singularidades.

8.2 Modos de uso en el proyecto

En modo pasivo (realista) el operador guía el brazo siguiendo las **trayectorias recomendadas** generadas por el planificador. El sistema registra poses vía **cinemática directa** a partir de las lecturas de junta y la nube resultante se referencia al marco de la placa. La utilidad de planificar no desaparece: el raster aporta repetibilidad, uniformidad de espaciamiento y un marco para comparar ejecuciones y detectar desviaciones.

En la variante cooperativa actuada (académica), el mismo pipeline se ejecuta automáticamente: la pasada recta se sintetiza en cartesiano y se proyecta a juntas con cinemática inversa; los u-turns se generan mediante waypoints y se interpolan en juntas, todo bajo límites de posición, velocidad y aceleración. El sistema puede trabajar en control de admitancia para permitir guía humana sin perder la capacidad de **seguir perfiles**. En ambos modos las **demonstraciones** se circunscriben a casos canónicos y reproducibles (pasadas rectas, arcos simples y raster plano), suficientes para validar el modelo, ilustrar el papel del jacobiano y documentar con claridad los resultados de simulación.

8.3 Muestreo de una placa de dimensiones $H \times W \times e$

El pipeline de planificación se implementa como un programa monolítico que genera la trayectoria completa, tanto las líneas de escaneo como las transiciones, de forma secuencial y acumulativa. La estrategia es definir todos los movimientos en el espacio cartesiano y resolverlos punto a punto mediante cinemática inversa. El programa itera por el número de líneas de escaneo (`num_lines`). En cada iteración:

Línea de Escaneo:

Se definen los puntos cartesianos de inicio P_0 y fin P_f de la línea. La dirección de avance en X se alterna para líneas pares e impares.

Se calcula la duración total del segmento T_{total} basándose en una velocidad cartesiana constante vel_{scan} . Esto genera un perfil de velocidad rectangular, no trapezoidal.

Se genera una trayectoria de poses cartesianas T_{path} usando $ctraj(T0, Tf, N)$, que interpola linealmente desde la pose de inicio a la de fin.

Se genera la trayectoria articular Q_{seg} iterando sobre cada pose T_k de T_{path} y resolviendo la cinemática inversa (ikine o ikcon). Para asegurar continuidad, la solución articular previa q_{prev} se usa como semilla q_0 para la siguiente iteración.

Transición (U-Turn):

Si no es la última línea, la transición (maniobra en “U”) no se genera como un movimiento único y suave en juntas, sino como tres segmentos cartesianos rectos e independientes:

Elevar P_{lift} : Un movimiento recto en +Z.

Desplazar P_{shift} : Un movimiento recto en +Y.

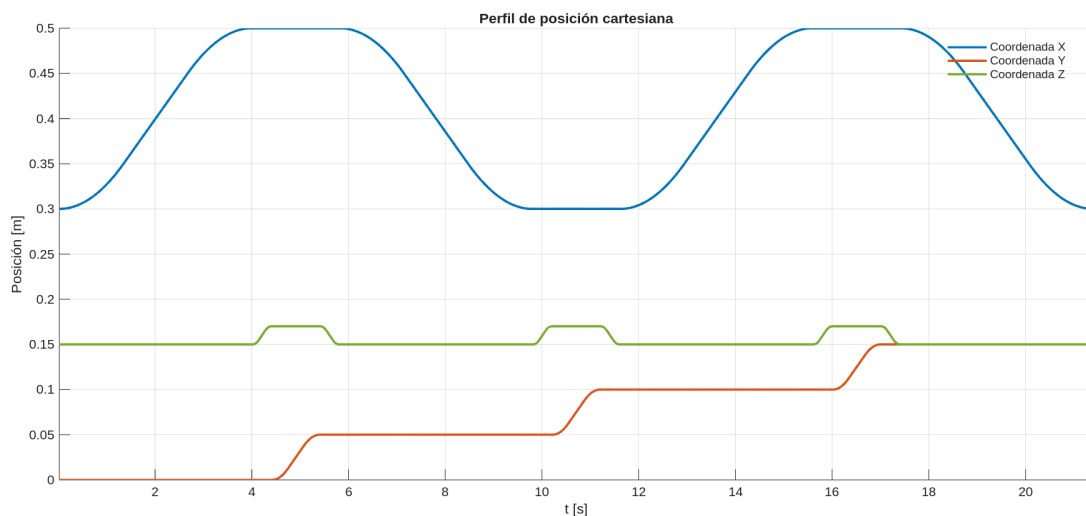
Descender $P_{descend}$: Un movimiento recto en -Z hasta la posición inicial de la siguiente línea.

Cada uno de estos tres segmentos se genera usando exactamente el mismo método que la línea de escaneo: se calcula un T_{total} (basado en vel_{move}), se genera un T_{path} con $ctraj$, y se resuelve la cinemática inversa punto a punto en un bucle.

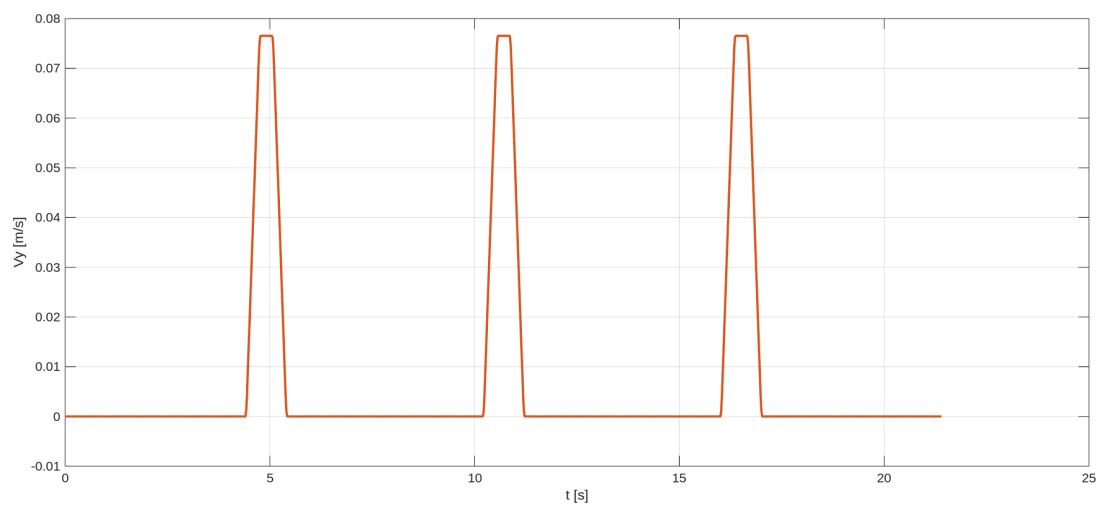
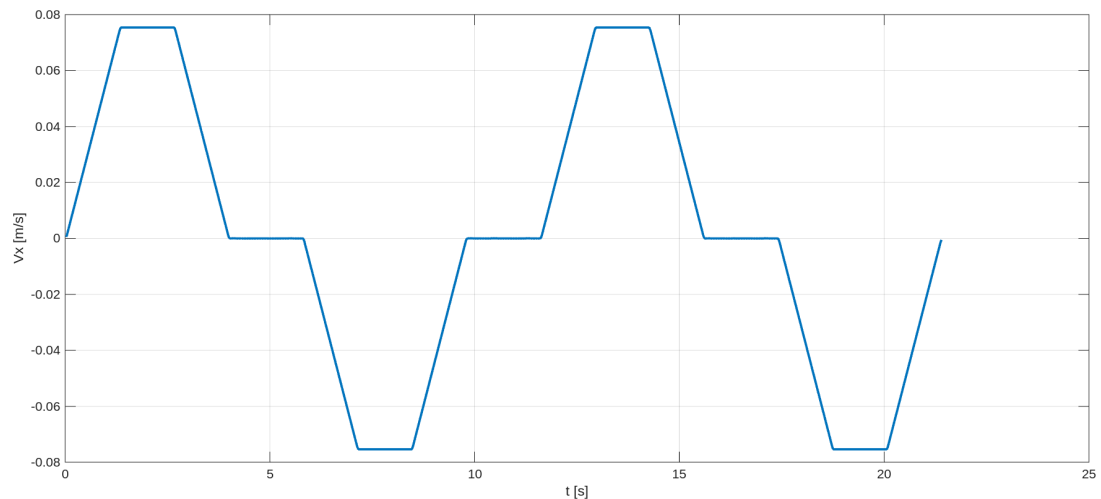
Todos los segmentos articulares Q_{seg} se concatenan en el arreglo Q_{all} y sus vectores de tiempo en t_{all} . Finalmente, las velocidades Qd y aceleraciones Qdd articulares no se planifican explícitamente, sino que se calculan a posteriori de forma numérica (usando $gradient()$) a partir de la trayectoria de posición Q_{all} resultante. El programa calcula y grafica:

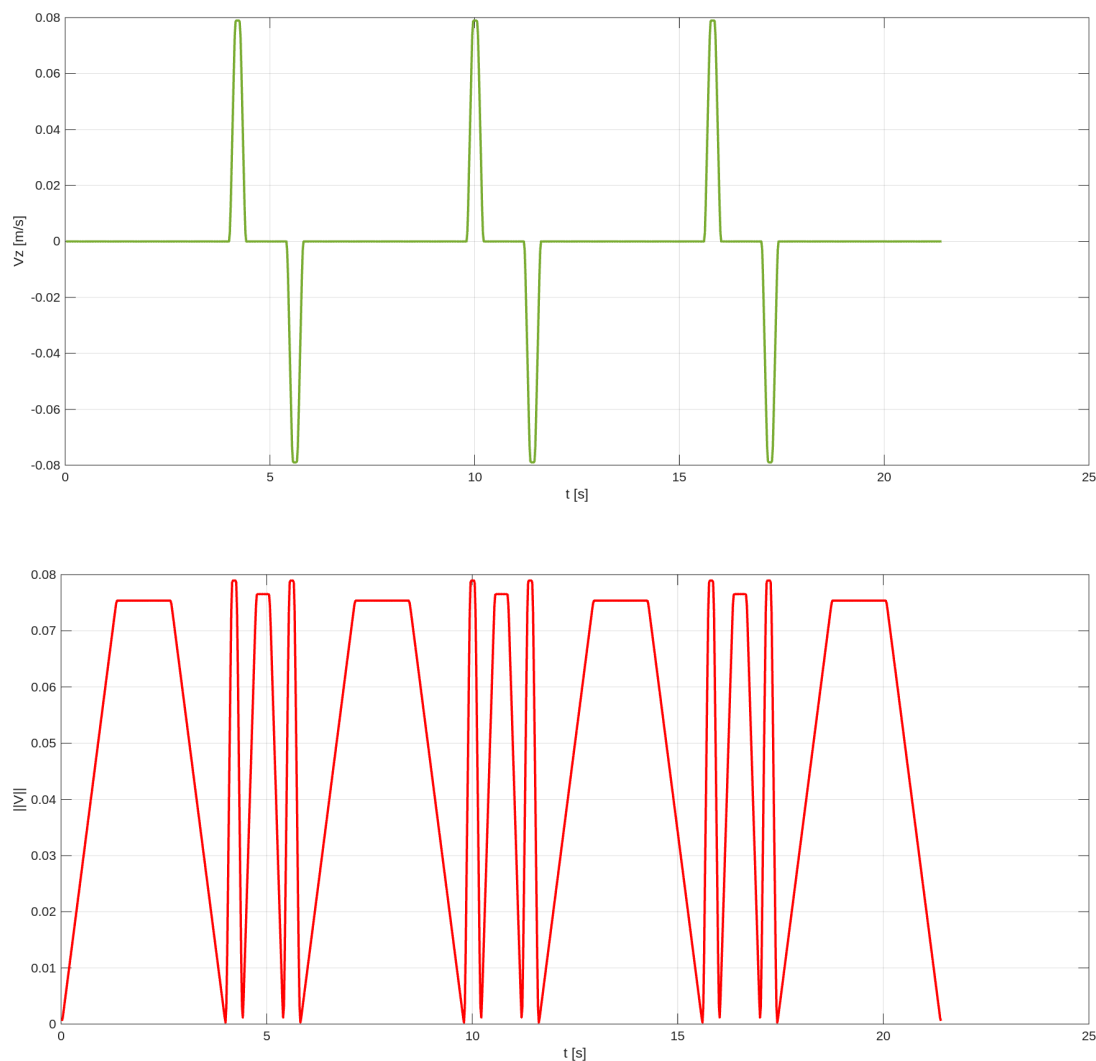
- (Q, \dot{Q}, \ddot{Q}) (articular);
- (\dot{x}, \ddot{x}) (TCP) vía (J) y (\dot{J}) ;
- $(\kappa(J))$ y (w) (manipulabilidad);
- **Traza del TCP** en `R.plot(..., 'trail', 'r-')`.

8.4 Gráficos obtenidos



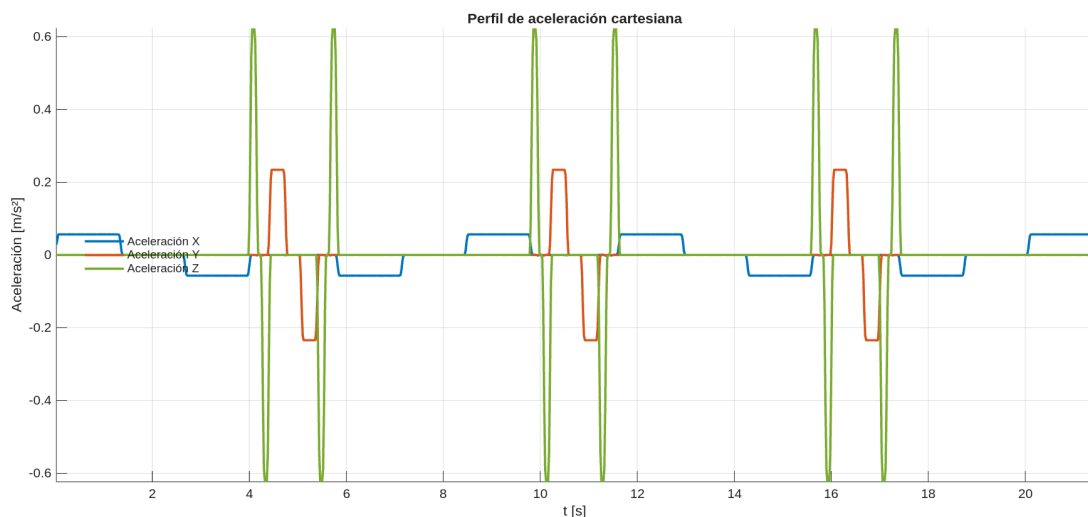
La **posición cartesiana** confirma el esquema de raster: en cada pasada el eje $x(t)$ evoluciona **monótonamente** con tramos lineales que corresponden al crucero a velocidad casi constante, separados por breves fases de aceleración y frenado; el eje $y(t)$ permanece **constante** durante la medición y cambia por **escalones** de Δy únicamente en los retornos; el eje $z(t)$ se mantiene en la cota de escaneo z_0 y solo realiza el **ciclo de despeje** (subir-cruzar-bajar) entre líneas. La trayectoria resultante es continua en posición y, dentro de cada pasada, también en velocidad, lo que garantiza un espaciamiento uniforme de muestras sobre la superficie medida y separa claramente los tramos útiles de las transiciones.





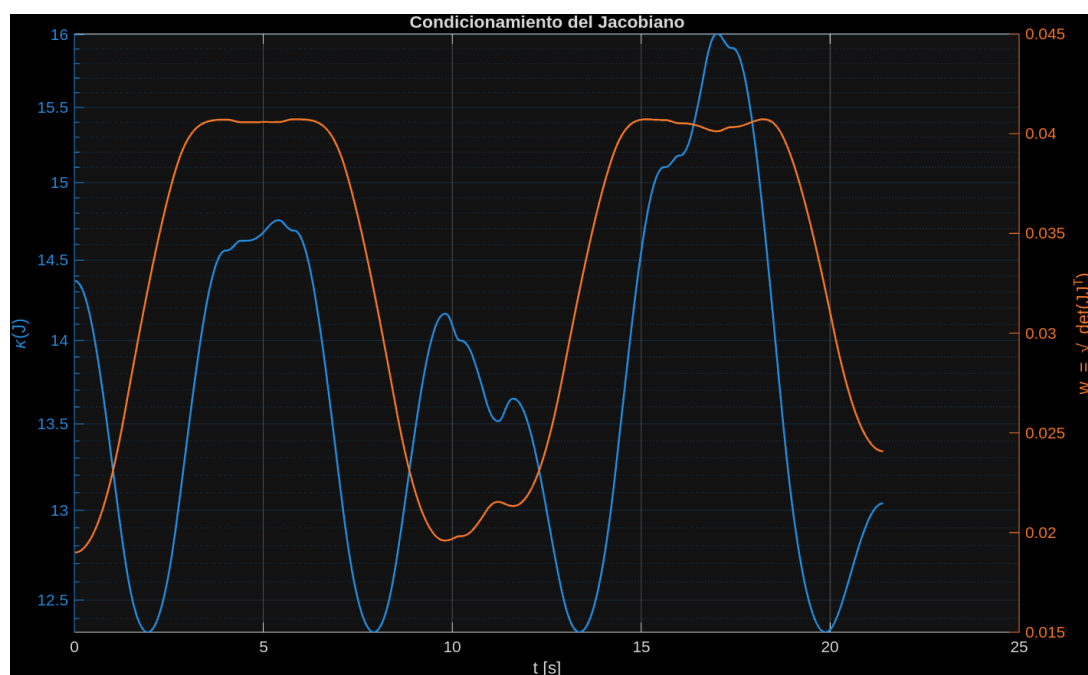
Los perfiles de **velocidad** muestran que las pasadas de escaneo en x se realizan con **velocidad prácticamente constante**: v_x presenta mesetas cercanas a 0.075–0.08 m/s y el **módulo** $\|\mathbf{v}\|$ replica el mismo valor durante los tramos activos. Esto confirma que la temporalización adoptada garantiza avance uniforme sobre cada línea, condición clave para muestreo homogéneo y para evitar sesgos por variación de velocidad.

Las **transiciones entre líneas (U-turn)** se manifiestan como pulsos bien definidos en v_y y, sobre todo, en v_z : breve elevación, desplazamiento lateral y descenso. La separación temporal de estos pulsos coincide con el fin de cada pasada en x , lo que respalda que el plan se está ejecutando según el pipeline propuesto (elevar-mover-bajar). La duración corta de estas maniobras reduce el tiempo improductivo entre pasadas, pero concentra los cambios de estado en ventanas pequeñas.

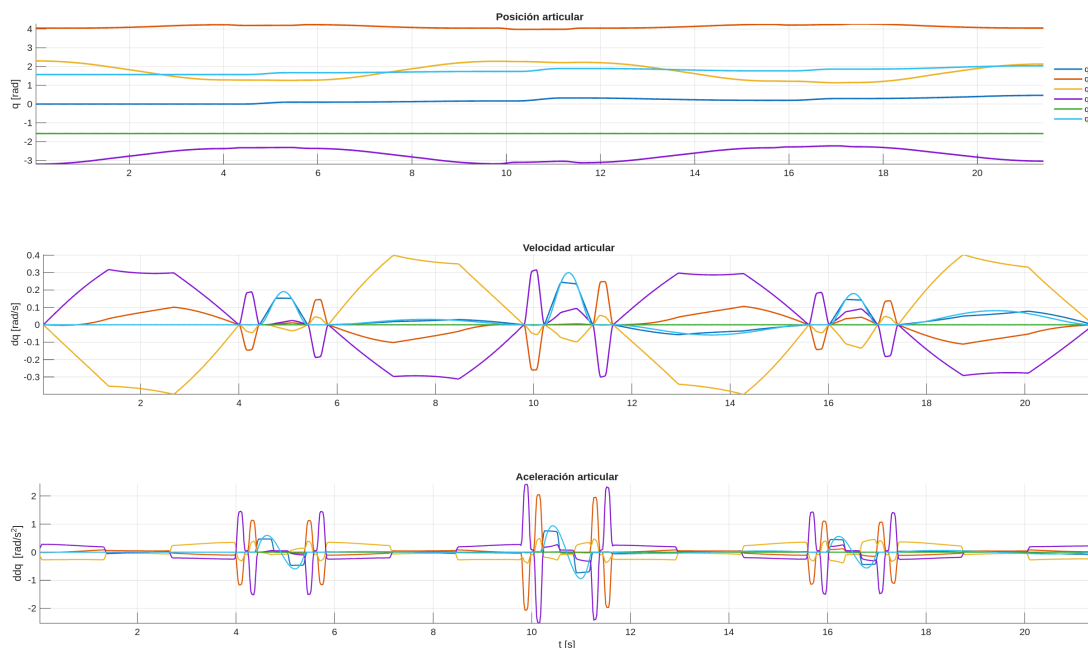


En **aceleración**, se observa el comportamiento esperado de un perfil tipo LSPB: a_x es casi por tramos constante durante aceleración y frenado, y nula en el crucero. En cambio, a_z exhibe picos angostos que superan a los de a_x y a_y por un orden de magnitud (del orden de 0.6 m/s^2 según la escala de la figura), coherentes con el “salto” vertical de los U-turn. Estos picos no afectan el muestreo en los tramos de escaneo, pero sí **incrementan la exigencia dinámica** en las articulaciones durante la transición.

Desde el punto de vista de **calidad metrológica**, los gráficos validan dos puntos: i) el **avance constante** en las pasadas, que asegura espaciamiento uniforme de muestras; ii) **transiciones discretas** fuera de la zona de medición, por lo que las variaciones de \mathbf{v} y \mathbf{a} no contaminan los datos útiles. Para robustecer aún más el lazo, conviene: suavizar las transiciones en z con **perfil S-curve o quintic** (limitando jerk), usar una **velocidad de movimiento** v_{move} menor que v_{scan} en las fases de elevar/bajar, y añadir un **pequeño dwell** antes de iniciar cada nueva pasada para disipar transitorios.



Finalmente, las variables articulares tienen gráficos:



9 Sensores y actuadores

En un Scan Arm de grado metrológico el operario es el motor. La arquitectura base es pasiva: cada articulación incorpora sensado angular y el sistema reconstruye la pose por cinemática directa. “Actuadores”, en este contexto, no significa servomotores que mueven el brazo, sino mecanismos y ayudas activas que mejoran ergonomía, seguridad y calidad de dato. Para cumplir con la cátedra también contemplamos una variante cooperativa actuada en simulación, donde la CI cobra sentido operativo.

9.1 Sensores

Clasificamos los sensores del sistema en internos (aquellos que miden el estado propio del robot) y externos (aquellos en el efector final, dedicados a medir la tarea).

Sensores Internos (Estado Articular y Cinemática)

Este es el conjunto de sensores más críticos para el análisis cinemático (CD/CI) del robot. El sistema se basa en una serie de codificadores angulares (encoders) de alta resolución, montados en el eje de cada una de las 6 articulaciones (GDL). La elección de encoders absolutos es deliberada y fundamental para la metrología. A diferencia de otros tipos, estos reportan la posición angular real de la junta desde el momento del encendido, sin necesidad de un ciclo de “homing” (búsqueda de cero). Esto garantiza que la referencia global del brazo sea robusta ante, por ejemplo, un corte de energía, que de otra manera invalidaría una medición en curso.

Estos sensores son la fuente de datos primaria que conecta el modelo geométrico (DH) con el mundo físico. Miden con precisión el ángulo de cada junta, proporcionando en tiempo real el vector de variables articulares:

$$q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$$

El rol de estos sensores para este proyecto:

- Rol en la Cinemática Directa (CD): En la operación normal (pasiva), donde el operario mueve el brazo, los valores leídos de estos encoders son la entrada directa al modelo de Cinemática Directa. El software utiliza este vector q para resolver la ecuación ${}^0T_6 = f(q)$ y así calcular la pose (posición y orientación) del efector final (TCP) en el espacio.
- Rol en la Cinemática Inversa (CI): En la “variante cooperativa actuada” (evaluada académicamente en este informe), la CI calcula el vector q deseado para alcanzar una pose objetivo. Los encoders se convierten entonces en la señal de retroalimentación para el lazo de control de posición de cada motor, asegurando que el robot alcance y mantenga la configuración articular calculada.

Sensores Externos (Adquisición de Tarea)

Estos sensores se ubican en el efector final (TCP) y su función es digitalizar la pieza de trabajo. Su propia precisión depende de que el modelo de CD, alimentado por los sensores internos, sea correcto.

- Palpador Táctil (Touch Probe): Es un sensor de contacto discreto. Al tocar la superficie de la pieza, genera un trigger (disparo) eléctrico de alta precisión.. En ese instante, el sistema “congela” la lectura de los encoders internos y calcula la CD para registrar la coordenada 3D exacta de ese único punto.
- Escáner Láser (LLP): Es un sensor de adquisición masiva. Utiliza triangulación óptica (una línea láser proyectada y una cámara CMOS) para capturar millones de puntos (nube de puntos) por segundo . La calidad de la nube de puntos depende de factores de la superficie (reflectividad, ángulo de incidencia) y, crucialmente, de la precisión y velocidad con que el sistema calcula la pose del escáner (CD) en cada instante, basándose en los encoders de las juntas. Para el TP alcanza con describir la física, el pipeline de registro y los límites: reflectividad, ángulo de incidencia, halo especular.

9.2 “Actuadores” en brazos pasivos

En un Scan Arm de grado metrológico, el operario es el “motor” primario. Por lo tanto, en el contexto pasivo (su uso industrial más habitual), el término “actuador” no se refiere a servomotores que mueven el brazo, sino a los mecanismos que asisten al operario y garantizan la calidad de la medición. Los componentes clave son:

- Contrapeso Interno: Un mecanismo pasivo (resortes o masas) que neutraliza la gravedad. Esto reduce la fatiga del operario y evita que el brazo caiga si se suelta, permitiendo mediciones más estables.
- Bloqueo/Freno de Junta: Sistemas de sujeción (a menudo electromagnéticos) que permiten al operario fijar una o todas las articulaciones en una pose específica. Esto es útil para soltar las manos, cambiar la sonda o parquear el brazo con seguridad.
- Retroalimentación al Usuario: Actuadores hápticos (vibración) o indicadores visuales en la muñeca. Alertan al operario sobre eventos clave: confirmación de un punto táctil, baja calidad de escaneo (si se usa LLP) o si se está alcanzando un límite articular.

9.3 Variante cooperativa actuada (enfoque académico)

Para justificar cinemática inversa y planeamiento, se plantea una opción actuada en simulación:

- Motores backdrivable o con control de admitancia/impedancia para que el humano guíe y el robot asista, con perfiles LSPB/quintic y límites de junta.

- Sensado de par (o estimado) para contacto suave y virtual fixtures que “peguen” la punta a una superficie virtual durante el barrido.
- Seguridad: paradas controladas, zonas de exclusión y validación previa del Jacobiano para evitar regiones mal condicionadas.

9.4 Selección concreta para el proyecto

Para los fines de este informe y sus simulaciones en MATLAB, se adopta la siguiente configuración: * Sensores Internos (CD/CI): Se asume el uso de encoders absolutos en cada junta, como “mejor práctica” por su robustez ante cortes de energía. Son la fuente de q para todos los cálculos.

- Sensores Externos (Tarea): Se utiliza el palpador táctil como baseline para la validación de la Cinemática Directa. Se describe el escenario de escáner láser (LLP) para justificar la necesidad de control de orientación y el análisis de trayectorias cartesianas.
- Actuación (Simulación): El análisis de CI, Jacobiano y planificación de movimiento se realiza sobre la variante cooperativa actuada. Las simulaciones en MATLAB ejecutarán trayectorias P2P (LSPB/tpoly) y barridos cartesianos, respetando los límites articulares definidos y demostrando la viabilidad de los modelos cinemáticos desarrollados.

10 Conclusiones

El proyecto estableció una base cinemática sólida para un brazo tipo ScanArm de 6 GDL parametrizado con DH estándar (unidades en metros), alineada con los objetivos de la cátedra. Se modeló la cadena geométrica, se implementó la cinemática directa y se incorporó una solución analítica de cinemática inversa orientada a planificación, aun considerando que el uso industrial habitual de estos equipos es pasivo. Este enfoque dual permitió estudiar alcanzabilidad y orientar la herramienta con criterios reproducibles, sin depender de un hardware actuado real.

El análisis del jacobiano y de las singularidades brindó un mapa claro de zonas mal condicionadas, útil para definir restricciones de operación y para justificar decisiones de trayectoria. La planificación en espacio articular con perfiles LSPB y polinomios de quinto orden, junto con trayectorias cartesianas en $SE(3)$, mostró transiciones suaves que respetan límites de posición, velocidad y aceleración, condición necesaria para preservar la calidad metrológica. Las demostraciones se realizaron deliberadamente con trayectorias simples (punto a punto, arco con normal controlada y raster sobre plano), suficientes para validar el modelo, ilustrar la relación con el jacobiano y documentar resultados de forma clara.

Además, el estudio diferencial del jacobiano permitió cuantificar la relación entre velocidades en el espacio de tarea y en el espacio articular. A partir de $\dot{x} = J(q)\dot{q}$ se utilizó $\dot{q} = J^{-1}(q)\dot{x}_{ref}$ para imponer una **velocidad constante del TCP** a lo largo de las pasadas y dimensionar en qué medida deben modificarse las **coordenadas articulares y sus derivadas** para sostener esa consigna. Este enfoque facilitó detectar con anticipación saturaciones de velocidad/aceleración por junta y ajustar la temporalización de los perfiles antes de ejecutar la trayectoria.

En cuanto a instrumentación, se describió el rol central de los encoders de junta y se caracterizaron dos modalidades de adquisición en el efector (palpador y escáner óptico), subrayando cómo la calidad de dato depende de la calibración geométrica, la rigidez del conjunto y los perfiles cinemáticos. Para el contexto académico, se presentó una variante cooperativa actuada a

nivel de simulación que habilita la ejecución automática de vías mediante cinemática inversa, sin alterar el modelo geométrico de base ni la trazabilidad del análisis.

Por otra parte, el hecho de contar con una **muñeca esférica** permitió abordar la cinemática inversa mediante la descomposición clásica de **posición** (primeros tres grados de libertad hacia el centro de muñeca) y **orientación** (tres últimos grados de libertad), habilitando soluciones analíticas robustas y selección de ramas que evitan configuraciones desfavorables. En contraste, para un 6R **sin muñeca esférica** la CI debe resolverse con métodos numéricos y/o geométricos acoplados; en nuestras pruebas conceptuales, lograr simultáneamente una **velocidad de avance constante** y una **orientación estrictamente normal** a la superficie incrementó de forma notable la complejidad y la sensibilidad del planificador.

El trabajo es reproducible: se integró en scripts de MATLAB que generan las figuras de workspace, los perfiles cinemáticos y las animaciones empleadas en el informe. Esto facilita repetir experimentos, ajustar límites articulares y explorar nuevas trayectorias de forma controlada.

Como limitaciones, no se realizó calibración geométrica con artefactos físicos ni validación de incertidumbre de medida en banco real, y no se integró control a nivel de par ni sensado de fuerza para contacto. Quedan abiertas líneas de trabajo que agregan valor directo: calibración y compensación geométrica, incorporación de un modelo simple de error para estimar incertidumbre en el TCP, extensión a 7 GDL para mejorar la orientabilidad en escaneo, y validación experimental de perfiles temporales frente a restricciones de sensor (frecuencia de muestreo, reflectividad y ángulo de incidencia). Estas extensiones no modifican los resultados centrales del informe: el modelo cinemático es consistente, las trayectorias cumplen criterios de suavidad y límites, y el marco propuesto es adecuado para inspección y escaneo en una célula didáctica o de baja complejidad.

11 Referencias

- 1 FARO Technologies, "Quantum X FaroArm® Series | Brochure," Oct. 2024. [Online]. Available: https://.../Brochure_Quantum-X_3DM_ENG_Web.pdf. (letoltes.sidex.hu)
- 2 FARO Technologies, "Quantum X FaroArm® Series (product page)," 2025. [Online]. Available: <https://www.faro.com/en/Products/Hardware/Quantum-FaroArms>. (FARO.com)
- 3 Hexagon Manufacturing Intelligence, "The Absolute Arm (A4 brochure)," 2025. [Online]. Available: <https://.../Hexagon-MI-Arm-Absolute-Arm-Brochure-A4-en.pdf>. (shop.hexagonmi.com)
- 4 Hexagon Manufacturing Intelligence, "Absolute Arm systems and accessories (catalog)," 2023. [Online]. Available: https://hexagon.com/.../hexagon_mi_arm_absolute_arm_catalog_23_210x210_en.pdf. (hexagon.com)
- 5 Nikon Metrology, "MCAx S portable CMM arm with ModelMaker H120 (datasheet)," 2021. [Online]. Available: <https://industry.nikon.com/.../mcaxs-modelmaker-h120-en.pdf>. (Nikon Industrial Metrology - MASTER)
- 6 Kreon, "ACE measuring arm & Skyline scanners (brochure)," 2021. [Online]. Available: <https://www.crossco.com/.../Kreon-Ace-Brochure.pdf>. (CrossCo)
- 7 Kreon, "Measuring arm – Ace (product page)," 2025. [Online]. Available: <https://www.kreon3d.com/products/ace-measuring-arm>. (kreon3d.com)

- 8 Creaform, “HandyPROBE technical specifications,” 2025. [Online]. Available: <https://www.creaform3d.com/en/products/coordinate-measuring-machines-handyprobe/technical-specifications>. (creaform3d.com)
- 9 Creaform, “Coordinate measuring machines – HandyPROBE (product page),” 2025. [Online]. Available: <https://www.creaform3d.com/en/products/coordinate-measuring-machines-handyprobe>. (creaform3d.com)
- 10 ISO, ISO 10360-12:2016 — *Geometrical product specifications (GPS) — Acceptance and reverification tests for coordinate measuring machines (CMM) — Part 12: Articulated arm CMM*. Geneva, Switzerland: International Organization for Standardization, 2016. [Online]. Available: <https://.../ISO-10360-12-2016.pdf>. (Iteh)
- 11 ISO, ISO 10360-8:2013 — *Geometrical product specifications (GPS) — Acceptance and reverification tests for coordinate measuring machines (CMM) — Part 8: CMMs with optical distance sensors*. Geneva, Switzerland: International Organization for Standardization, 2013. [Online]. Available: <https://www.iso.org/standard/54522.html>. (iso.org)
- 12 Kreon, “ISO 10360-12 standard for measuring arms (explainer article),” 2023. [Online]. Available: <https://www.kreon3d.com/article/iso-10360-12-standard-for-measuring-arms>. (kreon3d.com)
- 13 P. Corke, *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*, 2nd ed. Cham, Switzerland: Springer, 2017. [Online]. Available: <https://petercorke.com/books/robotics-vision-control-all-versions/>. (Peter Corke)
- 14 P. Corke, “Robotics Toolbox for MATLAB (official site),” 2025. [Online]. Available: <https://petercorke.com/toolboxes/robotics-toolbox/>. (Peter Corke)
- 15 P. Corke, “Robotics Toolbox for MATLAB (GitHub repository),” 2025. [Online]. Available: <https://github.com/petercorke/robotics-toolbox-matlab>. (GitHub)
- 16 J. Haviland and P. Corke, “A Systematic Approach to Computing the Manipulator Jacobian and Hessian using the Elementary Transform Sequence,” *arXiv:2010.08696*, 2020. [Online]. Available: <https://arxiv.org/abs/2010.08696>. (arXiv)
- 17 J. Haviland and P. Corke, “Manipulator Differential Kinematics: Part 1: Kinematics, Velocity, and Applications,” *arXiv:2207.01796*, 2022. [Online]. Available: <https://arxiv.org/abs/2207.01796>. (arXiv)
- 18 FARO Technologies, “Quantum X FaroArm® Series | Brochure (web page),” 2025. [Online]. Available: <https://www.faro.com/en/Resource-Library/Brochure/Quantum-FaroArm>. (FARO.com)

12 Anexos