



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

# **ROBÓTICA I**

## **Proyecto Final**

**Análisis Cinemático del Robot Franka Panda Emika**

**Integrantes:**

Juan Stella - Legajo: 12552  
Floencia Prado - Legajo: 12160  
Miled Felicito - Legajo: 11901  
Gabriel Medina - Legajo: 14086

**Universidad Nacional de Cuyo**  
Facultad de Ingeniería  
Cátedra: Robótica I

**Año 2025**

## Resumen

Este documento presenta un análisis cinemático del robot Franka Panda Emika. El estudio aborda el modelado cinemático completo del manipulador de 7 grados de libertad, aplicando la convención de Denavit-Hartenberg modificado para su parametrización.

Se utilizan distintos métodos para la cinemática directa e inversa, incluyendo métodos de solución y validación mediante simulaciones. El análisis del Jacobiano permite identificar singularidades y estudiar la posibilidad de obtener velocidades articulares. Adicionalmente, se implementa la generación de trayectorias mediante distintos métodos, para una representación aproximada del problema.

Por último se presentan gráficas para el análisis de posiciones, velocidades y aceleraciones, tanto cartesianas como articulares.

## Índice

### 1. Introducción

#### 1.1. Descripción del Robot Franka Panda Emika

#### 1.2. Justificación de la Selección

### 2. Presentación Técnica del Robot

#### 2.1. Especificaciones Técnicas

#### 2.2. Configuración Cinemática

#### 2.3. Tipos de Articulaciones y Límites

#### 2.4. Espacio de Trabajo

#### 2.5. Layout de la celda

### 3. Análisis Cinemático del Robot

#### 3.1. Parametrización Denavit-Hartenberg

#### 3.2. Cinemática Directa

#### 3.3. Cinemática Inversa

#### 3.4. Jacobiano y Relación de Velocidades

#### 3.5. Generación de Trayectorias

#### 3.6. Simulaciones y Gráficas Finales

### 4. Sensores y Actuadores del Sistema

#### 4.1. Actuadores

#### 4.2. Sensores Internos

#### 4.3. Sensores Externos

### 5. Conclusión

### 6. Referencias

### 7. Anexos



## 1. Introducción

### 1.1. Descripción del Robot Franka Panda Emika

El Franka Emika Panda es un manipulador colaborativo de 7 grados de libertad, diseñado para operar de forma segura junto a humanos gracias a sus sensores de par integrados en cada articulación.

En el contexto particular del proyecto, se pensó al Franka Panda Emika para aplicaciones de pulido de piezas, aunque puede ser utilizado para otras tareas como manipulación fina, ensamblaje de precisión, etc.

### 1.2. Justificación de la Selección

Su redundancia de 7 GDL permite mantener la orientación y presión constante de la herramienta durante el pulido, incluso sobre superficies curvas.

## 2. Presentación Técnica del Robot

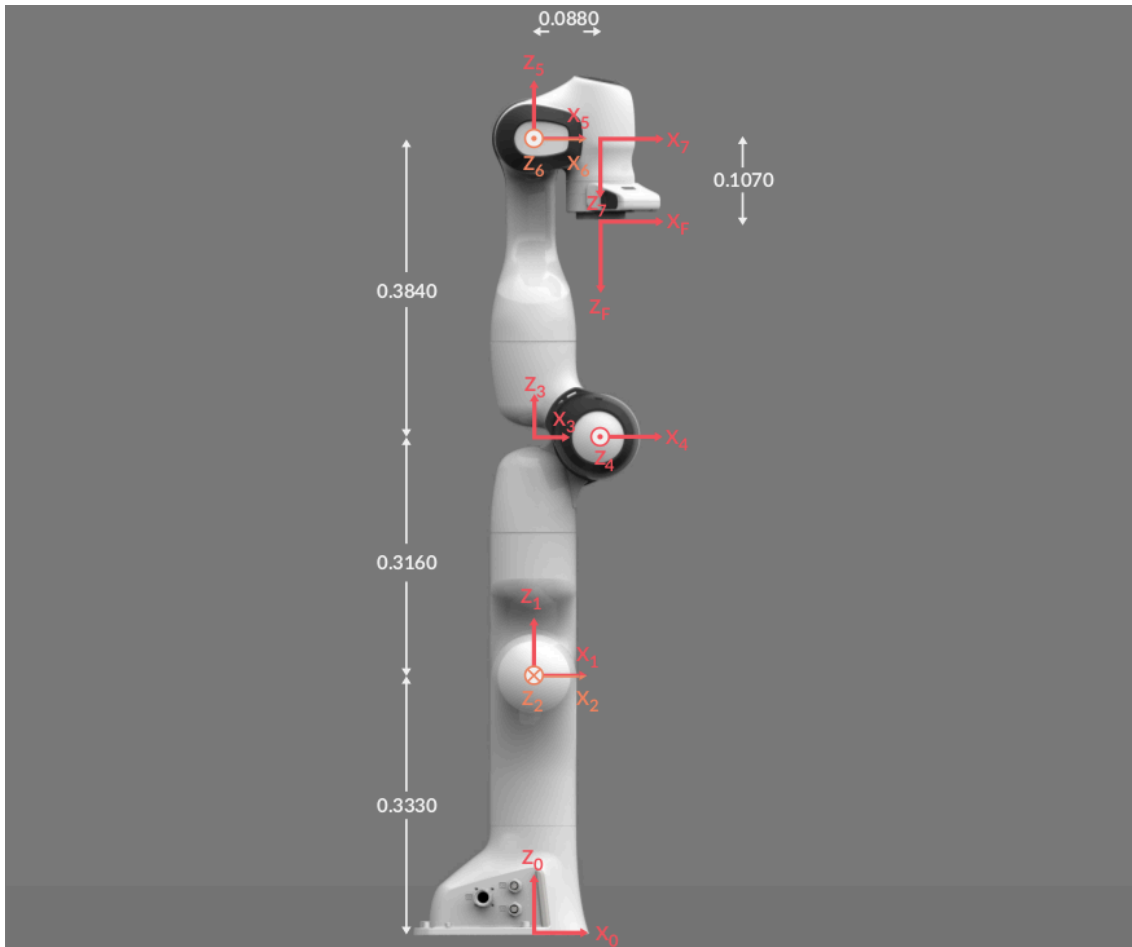
### 2.1. Especificaciones Técnicas

Parámetro	Valor
Grados de libertad	7 DOF
Carga útil máxima	3 kg
Alcance máximo	855 mm
Repetibilidad	$\pm 0.1$ mm
Peso del robot	18 kg
Velocidad máxima articular	2.175 rad/s (todas las articulaciones)

Velocidad Articular	Límite
$q_1, q_2, q_3, q_4$	150°/s
$q_5, q_6, q_7$	180°/s

## 2.2. Configuración Cinemática

El robot presenta una configuración de cadena abierta con siete articulaciones rotacionales, distribuidas de forma similar a un brazo humano (hombro–codo–muñeca). Esta estructura redundante permite mantener la orientación del efector mientras se optimiza la postura de las articulaciones



**Figura 1:** Esquema del Franka Panda Emika mostrando los 7 grados de libertad y sistemas de coordenadas.

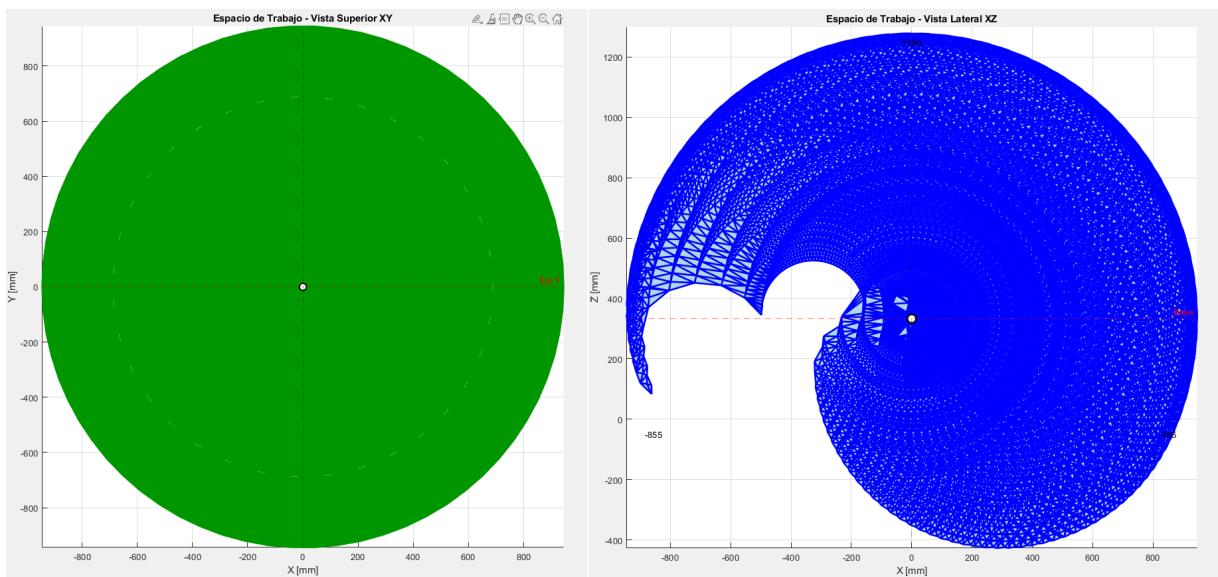
### 2.3. Tipos de Articulaciones y Límites

Todas las articulaciones del Franka Panda son de tipo rotacionales, con los siguientes límites angulares:

Articulación	Límite Inferior	Límite Superior
$q_1, q_3, q_5, q_7$	$-166^\circ$	$+166^\circ$
$q_2$	$-101^\circ$	$+101^\circ$
$q_4$	$-176^\circ$	$-4^\circ$
$q_6$	$-1^\circ$	$+215^\circ$

### 2.4. Espacio de Trabajo

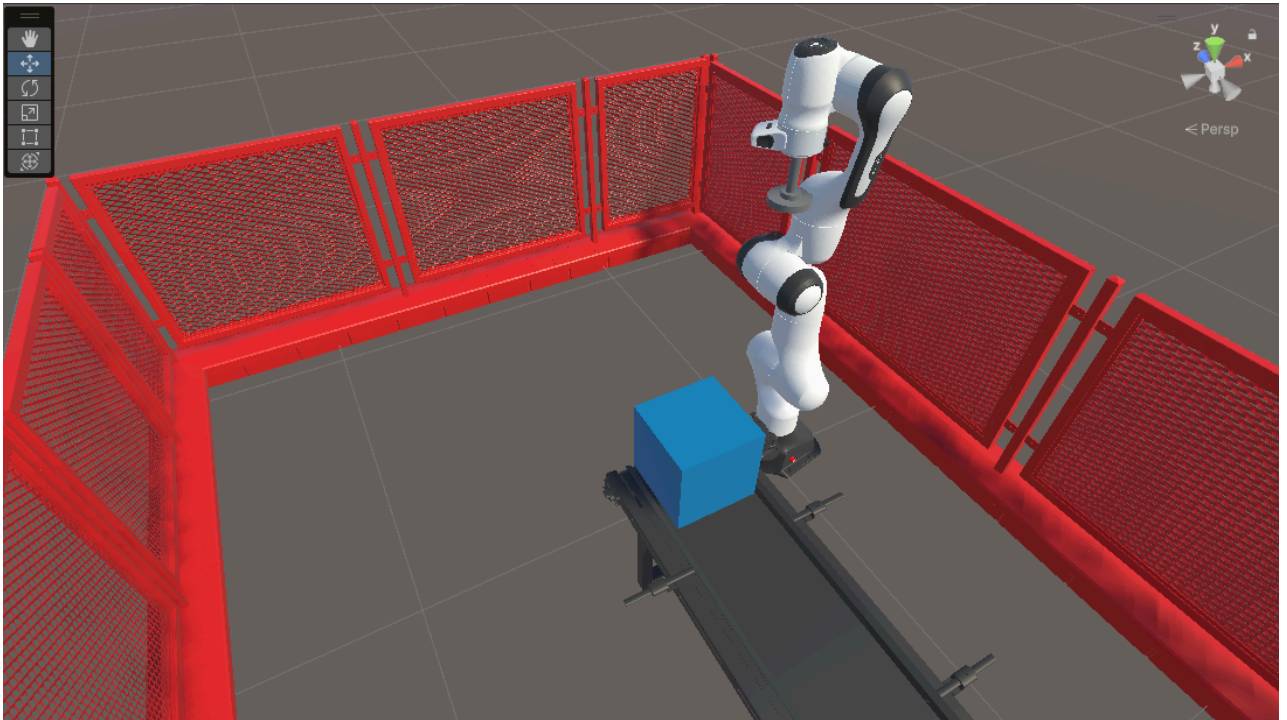
El espacio de trabajo del Franka Panda es aproximadamente esférico con un radio de 855 mm desde la base.



**Figura 2:** Espacio de trabajo del Franka Emika Panda.

## 2.5. Layout de la celda

Para el layout de la celda se planteó un entorno simple: Un compartimiento donde está posicionado el robot y una cinta transportadora que aproxima la caja a ser pulida hasta una posición donde quedará fija. Luego, el robot realiza la tarea de pulido sobre dos caras de la misma (superior y lateral derecha).



**Figura 3:** Layout de la celda.

## 3. Análisis Cinemático del Robot

### 3.1. Parametrización Denavit-Hartenberg

#### 3.1.1. Convención Utilizada

Para la parametrización del Franka Panda se utiliza la convención modificada de Denavit-Hartenberg (DH modificado). Esta convención permite representar la relación entre sistemas de coordenadas consecutivos mediante cuatro parámetros de cierta manera explicada.

En comparación con los parámetros DH clásicos, las coordenadas del marco  $O_{i-1}$  se ponen sobre el eje  $i-1$ , no sobre el eje  $i$  en la convención clásica de DH.

Las coordenadas de  $O_i$  se colocan en el eje  $i$  (no en el eje  $i+1$ ) en la convención DH clásica.



Según la convención DH modificada, la matriz de transformación se aplica en este orden, a diferencia a la no modificada:

$${}^{n-1}T_n = \text{Rot}_{x_{n-1}}(\alpha_{n-1}) \cdot \text{Trans}_{x_{n-1}}(a_{n-1}) \cdot \text{Rot}_{z_n}(\theta_n) \cdot \text{Trans}_{z_n}(d_n)$$

### 3.1.2. Tabla de Parámetros DH

Los parámetros de Denavit-Hartenberg para el Franka Panda son:

i	$\theta_i$	$d_i$ (mm)	$a_i$ (mm)	$\alpha_i$ (rad)
1	$q_1$	333	0	0
2	$q_2$	0	0	$-\pi/2$
3	$q_3$	316	0	$\pi/2$
4	$q_4$	0	82.5	$\pi/2$
5	$q_5$	384	-82.5	$-\pi/2$
6	$q_6$	0	0	$\pi/2$
7	$q_7$	0	88	$\pi/2$

## 3.2. Cinemática Directa

### 3.2.1. Introducción

Para nuestro robot, Franka Emika Panda con siete articulaciones rotacionales, se busca obtener la transformación homogénea total ( ${}^0T_7$ ) que relaciona el sistema de coordenadas de la base con el del efector final.

### 3.2.2. Método de Solución

Se utiliza la convención DH modificada para mantener coherencia con la cinemática inversa, ya que tanto el modelo oficial del robot como los trabajos de referencia adoptan esta formulación.

Esto asegura que las matrices obtenidas en ambos métodos sean equivalentes y compatibles.

La solución se implementa mediante la función Fkine del Robotics Toolbox de Peter Corke.

Esta función recibe como entrada los valores articulares y el modelo del robot (definido como un objeto SerialLink con parámetros DH modificados), devolviendo la matriz de transformación homogénea  ${}^0T_7$ .

En este caso, el robot se modeló sin aplicar transformaciones adicionales en la base ni offsets articulares, manteniendo la misma configuración encontrada en los artículos tomados como referencia.

Para obtener el espacio de trabajo, se variaron secuencialmente las articulaciones que afectan cada plano: se llevó una  $q_i$  a su máximo valor y se barrió el rango de otra  $q_j$ , repitiendo el proceso para distintos valores de  $q_i$ . Las trayectorias del efector final se graficaron y suavizaron, generando las curvas que se muestran en la Figura 2.

### 3.2.3. Validación y Resultados

Los resultados obtenidos muestran coincidencia con los gráficos de espacio de trabajo proporcionados por el fabricante, tanto en forma como en alcance.

Además, la matriz DH (modificada) resultante se contrastó con modelos previos del robot, verificando su coherencia y validez para futuras aplicaciones de cinemática inversa mediante el método adoptado.

## 3.3. Cinemática Inversa

### 3.3.1. Planteamiento del Problema

El objetivo de la cinemática inversa (CI) es encontrar los valores articulares  $q_i$  que permiten alcanzar una pose cartesiana deseada del efector final TEE. Para el Franka Panda, siendo un robot redundante (7-GDL para alcanzar poses de 6-GDL), existen infinitas soluciones matemáticas posibles. Por este motivo optamos por seguir el desarrollo propuesto en el artículo mencionado en [1], logrando resolver el problema utilizando un método geométrico analítico.

En este se hacen las siguientes aclaraciones:

- **Parámetro de Redundancia:** El último ángulo articular,  $q_7$ , se fija como un parámetro de redundancia y se considera conocido en los pasos restantes. Esto reduce el problema a 6 dimensiones.
- **Pose de Entrada:** La pose cartesiana del efector final (TEE) se especifica como la entrada del problema de CI.
- **Punto Fijo:** La posición de O2 está fija debido al diseño del robot.

El método se basa en una formulación geométrica propuesta específicamente para el robot Franka Emika Panda, donde los tres últimos ejes (5, 6 y 7) no se intersecan en un mismo punto, ya que el eslabón 6 presenta un pequeño desplazamiento ("offset") entre los ejes 6 y 7. Esto implica que al girar la articulación  $q_7$ , la posición del efector final también se modifica levemente, por lo que el conjunto de las últimas tres articulaciones no forma una muñeca esférica. Por esta razón, el ángulo  $q_7$  se fija previamente, y los restantes ( $q_1$  a  $q_6$ ) se determinan analíticamente a partir de dicho valor. Esto reduce el problema de 7 a 6 grados de libertad.

Primero se determina la posición del sistema asociado al eslabón 7, retrocediendo una distancia fija desde el efector final en la dirección de su eje  $z$ . A partir de allí, se calcula la posición del sistema 6 considerando el desplazamiento y orientación definidos por el fabricante.

Luego se plantea un triángulo formado por los eslabones intermedios, donde las longitudes son conocidas y el ángulo del “codo” constituye la variable principal. Aplicando relaciones trigonométricas simples (Ley de Cosenos y Senos) se obtienen los valores de los ángulos internos y, con ellos, las posibles configuraciones geométricas del brazo. Estas soluciones generan dos configuraciones simétricas conocidas como “codo arriba” y “codo abajo”; sin embargo, debido a restricciones mecánicas, solo la segunda es físicamente realizable en el Panda.

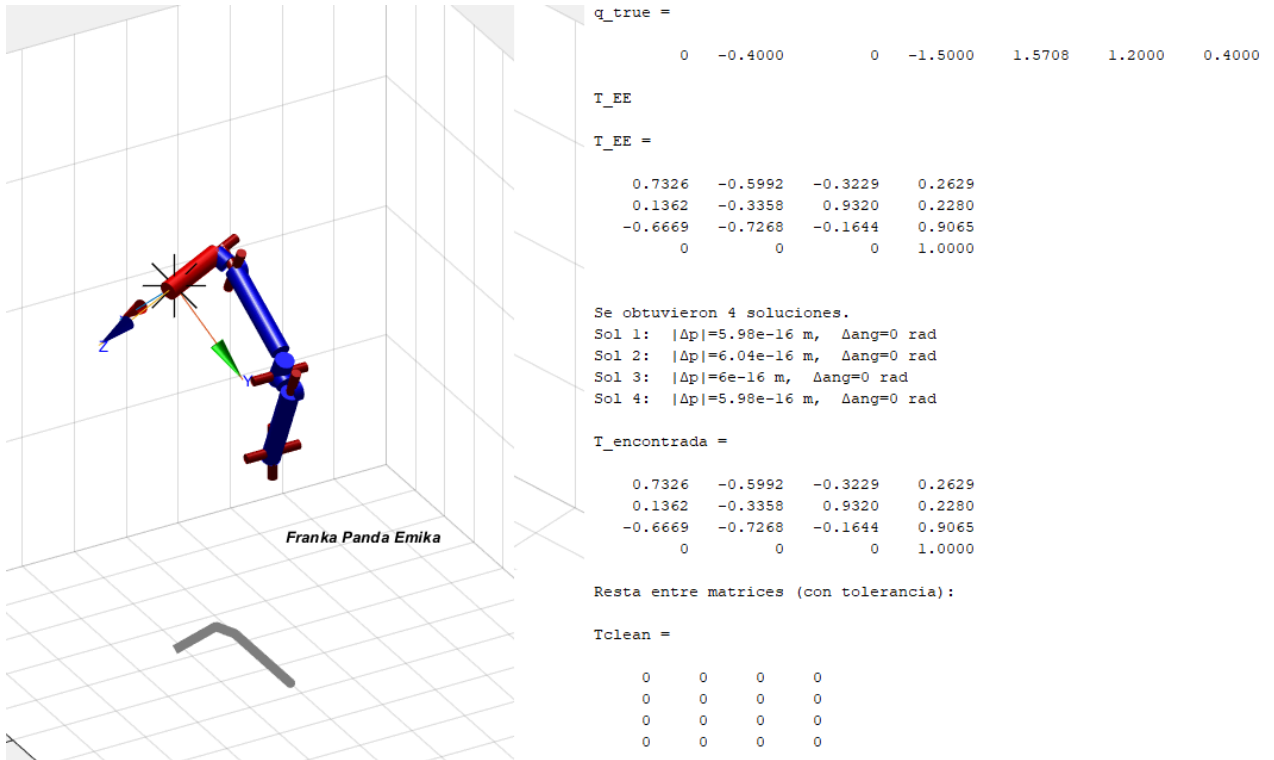
A continuación, se calculan las orientaciones de los sistemas sucesivos (del 6 al 1) asegurando la coherencia geométrica del modelo:

- El eje  $z_6$  se obtiene a partir de la orientación del efector final.
- El eje  $y_6$  se define paralelo al eje  $z$  del efector.
- El eje  $x_6$  surge del producto vectorial entre ambos, formando un sistema dextrógiro.

Finalmente, se resuelven los ángulos de las articulaciones del hombro y muñeca ( $q_2$  a  $q_6$ ) imponiendo condiciones de coplanaridad entre puntos clave del mecanismo y respetando la geometría impuesta por el fabricante. De esta manera, habiendo despreciado la solución A1 para  $q_4$ , quedamos con 4 soluciones prácticas posibles.

El procedimiento completo permite obtener una solución analítica cerrada para el conjunto de articulaciones, dependiente del valor de  $q_7$ . Esta formulación es válida para cualquier pose alcanzable dentro del espacio de trabajo del robot y fue validada en base al modelo cinemático descrito en el artículo de referencia.

A continuación se muestra un ejemplo básico de funcionamiento, donde  $T_{EE}$  es la matriz del efector final propuesta inicialmente (se encuentra a través de un  $q$  dato). Luego, se realiza cinemática inversa, se encuentra una solución y se verifica que la matriz que se encuentre sea exactamente igual a la matriz inicial dato. Adicionalmente, se grafica el punto y al robot en la posición encontrada con una de las soluciones, para verificar también visualmente la correcta solución.



**Figura 4:** Ejemplo de aplicación de CI.

### 3.4. Jacobiano y Relación de Velocidades

#### 3.4.1. Matriz Jacobiana y Análisis de Singularidades

Como tenemos un robot redundante en el espacio (7 GDL), la expresión del Jacobiano es muy extensa (pero es obtenible), pero para el determinante deberíamos utilizar distintos métodos para analizar singularidades (como por ejemplo dejar el 7ºGDL fijo, para que sean 6 GDL efectivos y tener un Jacobiano 6x6).

Pero el problema persiste al no tener muñeca esférica y no poder dividir el problema en 2 matrices de 3x3, por lo que los tiempos de cálculo del determinante simbólico sería muy grande.

Las singularidades ocurren cuando el Jacobiano pierde rango, indicando configuraciones donde el robot pierde uno o más grados de libertad. Se identifican calculando el determinante del Jacobiano  $J$ .

Por eso, pasamos al análisis visual del robot, donde podemos concluir que tendremos singularidades cuando las articulaciones 1, 3 y 5 se alinean (posición totalmente vertical, donde habrían infinitas posibilidades de movimiento para las articulaciones). Pero sabemos que como la articulación 4 tiene un límite de movimiento desde  $-4^\circ$  desde la vertical, entonces nunca se podrá llegar a esa posición en la práctica. Por lo tanto, tenemos únicamente una singularidad cuando  $q_1$  y  $q_3$  se alinean (haciéndolas linealmente dependientes), eliminando la singularidad con  $q_5$ .

Para demostrarlo se muestra a continuación un ejemplo donde se le da un vector de posiciones articulares  $q = [0 \ 0 \ 0 \ q_4 \ 0 \ 0 \ 0]$ , donde  $q_4$  se encuentra dentro de los límites articulares. Luego, se encuentra el jacobiano y su determinante para éste caso, viendo que  $\det(J)$  es aproximadamente igual a 0.

Recordemos que el resultado de  $\det(J) = 0$  nos indica la presencia de un punto singular, es decir, estamos frente a un caso de singularidad (solo nos pone en evidencia el punto singular, no nos dice cuál es la singularidad, pero en este caso solo contamos con una singularidad).

```
Jacobiano
0.0000    0.5836    0.0000   -0.2676    0.0000    0.0622         0
0.3579    0.0000    0.3579    0.0000    0.0880    0.0000         0
0         -0.3579         0    0.2754   -0.0000    0.0622         0
0.0000    0.0000    0.0000         0    0.7071         0   -0.7071
0.0000    1.0000         0   -1.0000   -0.0000   -1.0000   -0.0000
1.0000    0.0000    1.0000    0.0000    0.7071    0.0000   -0.7071

Jr =

0.0000    0.5836    0.0000   -0.2676    0.0000    0.0622
0.3579    0.0000    0.3579    0.0000    0.0880    0.0000
0         -0.3579         0    0.2754   -0.0000    0.0622
0.0000    0.0000    0.0000         0    0.7071         0
0.0000    1.0000         0   -1.0000   -0.0000   -1.0000
1.0000    0.0000    1.0000    0.0000    0.7071    0.0000

det(J) =    7.6703e-51

Rank:
6

Singu
1 linearly dependent joints:
q3 depends on: q1
```

**Figura 5:** Ejemplo de punto singular mediante determinante del Jacobiano.

Por otro lado tenemos otra singularidad presente cuando el eje de rotación de  $q_7$  se alinea de forma precisa con el centro cinemático del hombro. Esta configuración, se logra con el brazo flexionado hacia abajo. El problema se manifiesta porque el movimiento de  $q_7$  se vuelve linealmente dependiente de los ejes de la base.

## 3.5. Generación de Trayectorias

### 3.5.1. Criterios de Planificación

La planificación de trayectorias debe considerar múltiples criterios relacionados con la aplicación:

- Suavidad: minimizar frenados repetidos, es decir, intentar realizar la trayectoria sin parar en una cara del cuerpo a pulir.
- Tiempo: buscar reducir el tiempo, pero sin perder calidad.
- Precisión: seguimiento aproximado del camino en el espacio cartesiano. No es necesaria una exactitud en el paso por ciertos puntos, debido al diámetro de la herramienta.
- Intentar evitar singularidades y/o posiciones no óptimas: mantener manipulabilidad adecuada donde no se vea comprometido el movimiento del propio robot.

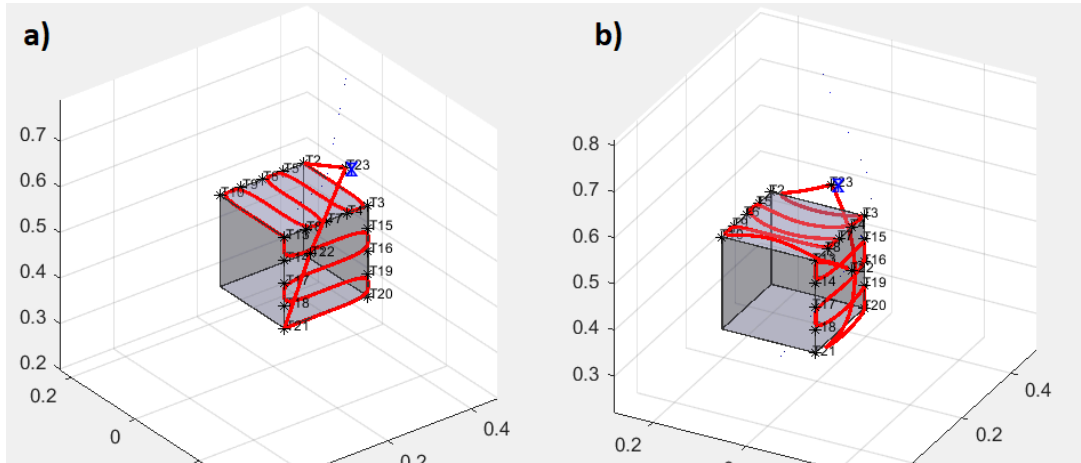
### 3.5.2. Trayectorias en Espacio Cartesiano

Como se mencionó anteriormente, debemos tener un control sobre la trayectoria del robot, en términos cartesianos. Por lo tanto, se implementaron dos distintas alternativas:

- Interpolación en el espacio cartesiano: En esta alternativa se resuelve la enteridad de la trayectoria mediante la función “CTRAJ”, la cuál interpola en el espacio cartesiano, dándonos la posibilidad de generar trayectorias en línea recta, pero con la desventaja de tener que parar entre puntos deseados, lo cuál podría ser un problema respecto al tiempo y quizás la necesidad de parar la herramienta en esos puntos (ya que se corre el riesgo de pulir de más).
- Aproximación en el espacio cartesiano: En este caso se resuelve mediante la función “MSTRAJ”, la cuál nos permite seguir una trayectoria aproximada sobre la cara del robot, pero manteniendo el control sobre el espacio cartesiano. Ahora, tenemos la ventaja de no frenar cuando se está realizando el pulido en la cara del cuerpo, si no que se sigue una trayectoria con cierta curvatura sobre la cara del mismo. Además, debemos aclarar que sí se utilizó “CTRAJ” cuando no se está puliendo, es decir, en los movimientos de posicionamiento. Ésta es la opción que se eligió como la más útil para la tarea.

También se realizó el análisis de la interpolación y/o aproximación en el espacio articular, pero se descartó automáticamente ya que debemos mantener siempre una cierta distancia a la cara del cuerpo que se está puliendo. Es decir, interpolando en el espacio articular no tenemos control sobre la trayectoria en sí (cartesiana), si no que sólo sobre los puntos iniciales y finales de la misma.

Para realizar pruebas de funcionamiento, se utiliza como cuerpo de ejemplo un cubo, simulando una caja a ser trabajada. Sobre la misma, se realiza una trayectoria “en serpentina”, ya que debido al diámetro o barrido de la herramienta, no sería necesario pasar por puntos demasiado próximos entre sí.



**Figura 6:** a) Trayectoria aproximada con “mstraj” en el espacio cartesiano. b) Trayectoria aproximada con “mstraj” en el espacio articular.

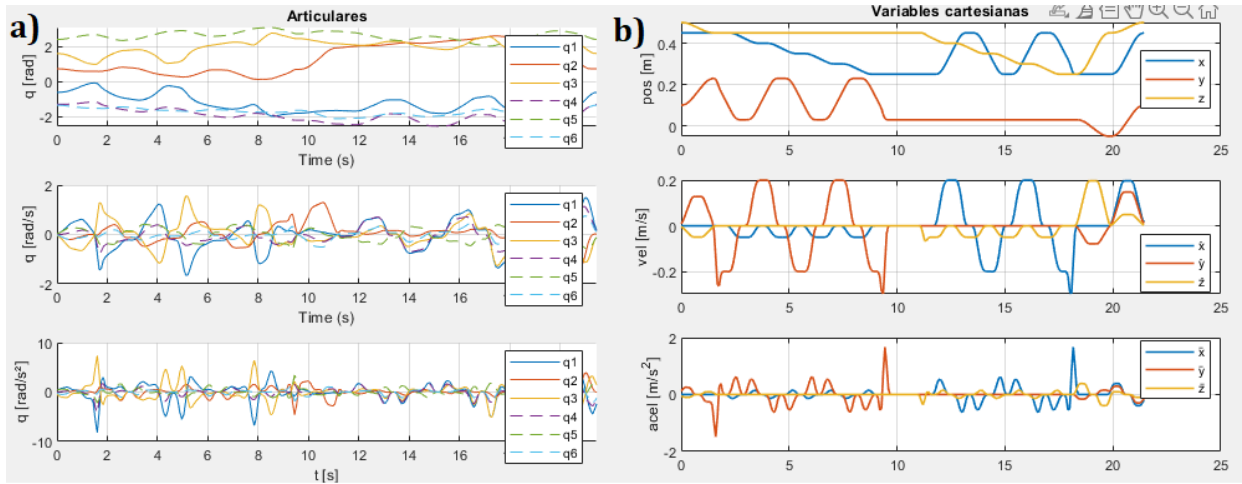
En la **Figura 6 b)** podemos observar que se pierde el control sobre los puntos intermedios entre segmentos, por lo que el robot podría alejarse o introducirse dentro del objeto, mostrando gráficamente por qué se descarta la opción de trabajar en el espacio articular.

En la **Figura 6 a)** podemos observar que se utilizó “ctrj” para los movimientos de aproximación o alejamiento del objeto, debido a que de otra manera se corría el riesgo de colisionar debido a las trayectorias curvas.

### 3.5.3. Análisis de posición, velocidad y aceleración

En cuanto al control de tiempos y velocidades de la trayectoria, se puede decir que:

- No se realizó un control sobre las velocidades y aceleraciones articulares, ya que aproximamos en el espacio cartesiano.
- Se impuso una cierta cantidad de tiempo en el que cada tramo debe ser realizado, es decir, un tiempo por cada segmento que es aproximado por “MSTRAJ”.



0

**Figura 7:** a) Posición, velocidad y aceleración en espacio articular. b) Posición, velocidad y aceleración en espacio cartesiano

Analizando los gráficos, podemos ver que aunque el control se está realizando sobre variables cartesianas, en las variables articulares tenemos un cambio suave, es decir,  $q1$  a  $q6$  no tienen un cambio brusco en ninguna parte del trayecto.

Si vemos la posición en coordenadas cartesianas, vemos que se corresponde correctamente con el movimiento descrito en la **Figura 7**, ya que por ejemplo, en el primer tramo se hace una aproximación bajando en Z, que luego se mantiene constante, X decrece en todo el tramo e Y tiene un movimiento “oscilatorio”.

## 3.6. Simulaciones y Gráficas Finales

### 3.6.1. Entorno de Simulación

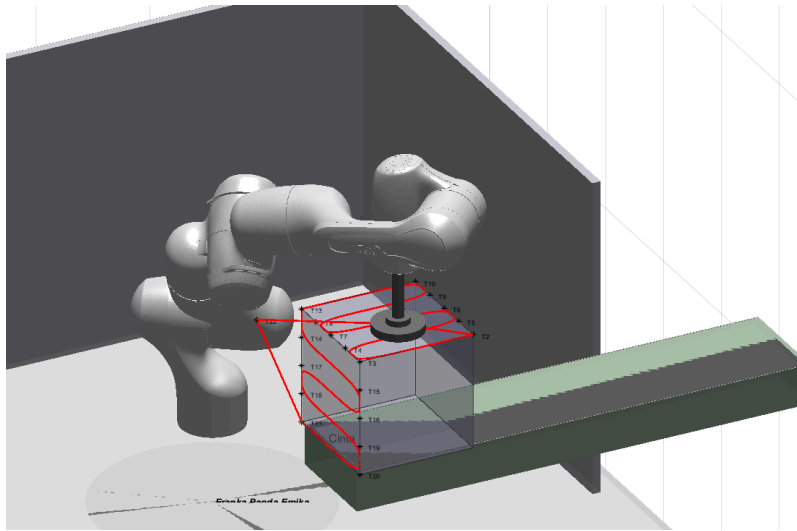
Las simulaciones se desarrollaron utilizando:

- MATLAB Robotics Toolbox para cálculos cinemáticos.
- Archivos .STL, .DAE y .URDF para la una visualización realista del robot.
- Entorno Unity para representación gráfica, con archivos encontrados en [5].

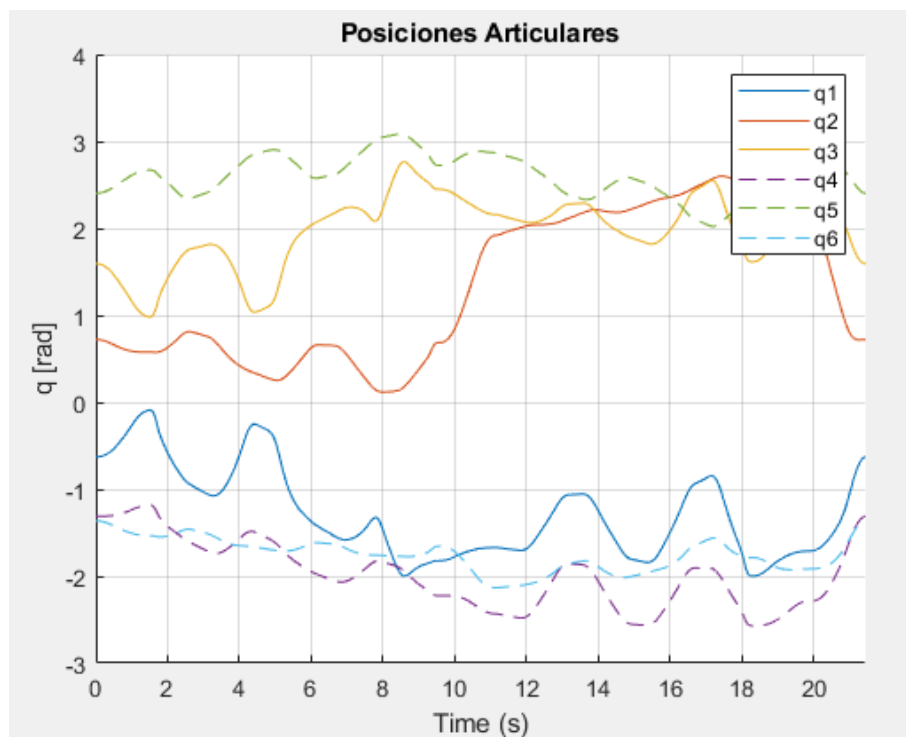
La simulación final podrá encontrarse en el apartado de anexos.

### 3.6.2. Gráficos finales

Los gráficos correspondientes a posiciones, velocidades y aceleraciones, tanto articulares como cartesianas, han sido expuestas a lo largo del desarrollo, por lo que solo se desarrollará más detenidamente los gráficos de posiciones articulares y una imagen ilustrativa de la trayectoria graficada en Matlab.



**Figura 8:** Animación en MATLAB del Franka Panda Emika a lo largo de la trayectoria propuesta.



**Figura 9:** Detalle de posiciones articulares.

Como podemos observar en la **Figura 9**, encontramos que todas las variables articulares siguen una forma suave, indicando que no hay presencia de discontinuidades o saltos en la trayectoria.

Además, es necesario mencionar que para el desarrollo de la simulación y de la trayectoria, se asumen todos los límites articulares en el rango  $[-180^\circ, 180^\circ]$ , lo cuál coincide correctamente con la figura anterior, ya que ninguna variable articular supera dichos límites (si no, estaríamos ante la presencia de una discontinuidad).

Dicha decisión fue tomada debido a la falta de información confiable sobre la “posición cero” definida por el fabricante para determinar los límites articulares. Distintos desarrollos muestran distintas posiciones, pero no mencionan específicamente que sea la posición cero definida por el fabricante.

Esto fue detectado principalmente debido a la articulación 6, donde tenemos un límite de  $[-1^\circ, 215^\circ]$ , y en ningún momento de la trayectoria propuesta nuestro robot cumpliría con tener a dicha articulación dentro de esos límites. Entonces, se infirió que probablemente dichos límites están definidos a partir de una posición cero desconocida.

Por último debemos mencionar que, gracias a estos inconvenientes en la etapa del desarrollo, se detectó que se debía normalizar  $q_6$  dentro de los límites articulares, según se nombra en [1], ya que por lo contrario puede dar valores muy elevados. En nuestro caso en particular, se normaliza en el rango  $[-180^\circ, 180^\circ]$ , por las razones mencionadas anteriormente.

## **4. Sensores y Actuadores del Sistema**

### **4.1. Actuadores**

#### **4.1.1. Motores de las Articulaciones**

Cada articulación del Franka Panda está accionada por un motor brushless de alta eficiencia con reductor armónico integrado.

- Alto par de salida con tamaño compacto
- Baja inercia para movimientos rápidos y precisos
- Bajo backlash para mayor precisión de posicionamiento
- Control de corriente preciso para regulación de par



## **4.2. Sensores Internos**

### **4.2.1. Sensores de Par en Articulaciones**

Una de las características del Franka Panda es la integración de sensores de par de alta resolución en cada una de las 7 articulaciones. Estos sensores permiten:

- Detección de colisiones en tiempo real
- Control de impedancia para interacción segura
- Estimación de fuerzas externas
- Compensación de gravedad y fricción

### **4.2.2. Encoders de Posición**

Cada articulación incorpora encoders absolutos de alta resolución que proporcionan:

- Medición de posición absoluta (sin necesidad de homing)
- Frecuencia de actualización: 1 kHz

## **4.3. Sensores Externos**

### **4.3.1. Cámara de Visión**

Para aplicaciones de visión artificial, se puede integrar una cámara RGB-D en el efector final o como cámara fija.

### **4.3.2. Sensores de Fuerza-Par Externos**

Para tareas que requieren medición precisa de fuerzas de contacto, se puede montar un sensor:

- Medición simultánea de fuerzas ( $F_x$ ,  $F_y$ ,  $F_z$ ) y momentos ( $M_x$ ,  $M_y$ ,  $M_z$ )
- Resolución típica: 0.05 N en fuerza, 0.01 Nm en par
- Aplicaciones: ensamblaje con control de fuerza, pulido, inspección táctil

## 5. Conclusión

### 5.2. Trabajos Futuros

Como trabajos futuros podemos considerar distintos casos:

- Cinemática directa: Generación de gráficos más precisos sobre el espacio de trabajo, con posibilidad de agregar puntos que puedan ser posibles puntos singulares. Analizar el espacio de trabajo tanto en posición como en orientación. Generar una interfaz interactiva para probar funcionamientos de una forma más visual (Mejora de R.teach)
- Cinemática inversa: Considerar la posibilidad de agregar manejo de errores y que las soluciones que estén dentro de límites articulares. Considerar el caso descartado de  $q_4$ , de manera de tener ocho soluciones en total. Además, existen análisis donde no se elimina un grado de libertad (llamado grado de libertad redundante), si no que se proponen métodos y soluciones que nos permitan el control de los 7 GDL.
- Jacobiano: Realizar un análisis y control más profundo de las velocidades articulares, de manera de poder mejorar la calidad de la tarea.
- Generación de trayectorias: Considerar un entorno más complejo y realista, donde se encuentren más objetos en el espacio. Optimizar la trayectoria investigando más a profundidad cómo es el proceso real de pulido mediante robots en industrias existentes. Incorporar algún método de detección de colisiones.

## 6. Referencias

[1] Y. He and S. Liu, “Analytical Inverse Kinematics for Franka Emika Panda – a Geometrical Solver for 7-DOF Manipulators with Unconventional Design”

[2] [Cinemática inversa](#)

[3] [Análisis de singularidades - Peter Corke](#)

[4] [Archivo URDF](#)

[5] [Animación en Unity](#)

[6] [Artículo GEOFIK, para análisis de singularidades](#)

## 7. Anexos

### Anexo A: Códigos

#### [Códigos Proyecto](#)

En el mismo podemos encontrar:

- **Cinemática directa:** “Franka\_Panda\_CD.m”
- **Espacios de trabajo:** “Franka\_Panda\_ESPTRAB\_XY.m” y “Franka\_Panda\_ESPTRAB\_XZ.m”
- **Jacobiano:** “Franka\_Panda\_Jacobiano.m”
- **Cinemática inversa:** “cin\_inv\_panda.m”
- **Trayectoria en espacio cartesiano:** “Franka\_Panda\_MSTRAJ\_Cartesiano.m”
- **Trayectoria en espacio articular:** “Franka\_Panda\_MSTRAJ\_Articular.m”
- **Archivos para la visualización con URDF y STL:** Carpeta “franka\_description”
- **Archivo de pulidora simple y representativa:** “Pulidora.stl”

Para la correcta ejecución del código simplemente se debe descargar el repositorio completo y ejecutar alguna de las trayectorias. Las mismas realizan el llamado a la cinemática inversa para la resolución, y la cinemática inversa llama a la definición del robot.

### Anexo F: Hojas de Datos

#### [Franka Emika Panda Datasheet](#)