**Robot barista**



Cátedra: Robótica 1

Profesora responsable de cátedra: CAROLINA DIAZ BACA

Jefe de trabajos prácticos:ERIC SANCHEZ FERREYRA

Cursado 2023

Integrantes del grupo y legajo:

* Casarotto Mauricio 12341
* Tassara Renzo 12299

**ÍNDICE**

[**Resumen 2**](#_dvhz1gpp4vmm)

[Introducción 2](#_u4l3oxc81bm1)

[Presentación del robot y la tarea 2](#_pv2fhlqv4eks)

[Tarea a realizar 2](#_9ywkdtvb1ux5)

[Robot seleccionado 3](#_r9ylgz5hfzm)

[Especificaciones técnicas del IRB 1100-4/0.475 4](#_cd4sv27osdu)

[Espacio de trabajo 5](#_dqpohm8jzht6)

[Robot 6](#_7xc53uhk0fix)

[Cinemática directa 6](#_axwwriee7nqd)

[Cinemática inversa 9](#_s4iy18iyw27k)

[Planificación de trayectoria 10](#_kmxsapiqjcs9)

[Relación de velocidades 13](#_78moise17sdm)

[Herramienta 15](#_l90fd9y9um9d)

[Sensores 16](#_x98fcs6xsn1v)

[Sensores externos 16](#_cw0bn8piq00r)

[Sensores internos 17](#_vevdkww7e8dm)

# Resumen

En este informe se explicará el proyecto final de la cátedra “Robótica 1”, el cual hemos desarrollado en el transcurso del semestre. Este proyecto involucra muchas temáticas de dicha cátedra, tales como la parametrización del robot, analizar cinemática directa e inversa, sus puntos singulares, planificación de trayectorias, etc.

# Problemática a resolver

El objetivo de este informe es abordar la cinemática del movimiento de un brazo robótico para lograr la realización efectiva de diseños de latte art. El problema implica resolver las trayectorias requeridas, junto con sus posiciones, velocidades y aceleraciones correspondientes, para garantizar una ejecución precisa y consistente del latte art.

La complejidad radica en la necesidad de coordinar múltiples grados de libertad del brazo robótico mientras se cumplen los requisitos de precisión en el dibujo del latte art. Se deben calcular y controlar con precision las trayectorias para evitar colisiones, optimizar la eficiencia del movimiento y garantizar que la herramienta de dibujo del brazo robótico realice trazos suaves y precisos.

# Introducción

El presente informe aborda la implementación de un brazo robótico IRB 1100-4/0.475 de la marca ABB para la realización de Latte Art, que implica la creación de dibujos con leche emulsionada en una taza que ya contiene espresso.

Para la elaboración de este proyecto optamos por usar el IRB 1100-4/0.475, el cual es un brazo robótico versátil que ofrece 6 grados de libertad y una muñeca esférica, lo que lo hace ideal para la manipulación de utensilios y la creación de movimientos precisos. En este informe, se describirán las tareas específicas que el robot llevará a cabo, así como los desafíos y consideraciones técnicas relacionadas con la automatización del Latte Art.

# Presentación del robot y la tarea

## Tarea a realizar:

A continuación se detallaran todas las acciones que deberá hacer el brazo robótico:

1. *Buscar jarra (ya servida de leche con un peso aproximado de 0.5Kg) y agarrarla*
2. *Llevarla bajo la lanceta de la máquina espresso y esperar a que termine de emulsionar la leche\*1*
3. *Dar tres vueltas a la jarra para texturizar la leche\*2*
4. *Dar dos golpes a la mesa para sacar las burbujas*
5. *Estar en un posición arriba de la taza y empezar a dar tres vueltas para mezclar la leche con el cafe*
6. *Acercarse a la taza para que se expanda la crema sobre el cafe*
7. *Levantar jarra y cortar el dibujo para generar un “corazón”*
8. *Guardar la jarra donde estaba situada inicialmente*

En el siguiente link se podrá acceder a un video el cual un robot hace una tarea similar:

<https://www.youtube.com/shorts/R1NDNDZpneA>

Las tareas finales de nuestro brazo robótico se asemejan a las del brazo robótico del video realizando de esta forma el corazon en la taza de café (figura 1).



Figura 1 - Latte Art

## Robot seleccionado

Después de definir las acciones necesarias para la realización de Latte Art, el siguiente paso crucial en el proceso fue la selección del robot que mejor se adapte a los requisitos del proyecto. Para esta elección, se consideraron tres propuestas de robots fabricados por la marca ABB (figura 2). Cada uno de estos robots posee 6 grados de libertad, lo que permite una amplia gama de movimientos y tareas. Sin embargo, presentan diferencias significativas en términos de capacidad de carga y alcance de movimiento.

Figura 2.1 - **IRB 1010** Figura 2.2 - **IRB 1100** Figura 2.3 - **IRB 120**

En la siguiente tabla se puede ver las principales diferencias que nos ayudaron a tomar una decisión final:

| **Características generales** | | **IRB 1010** | **IRB 1100** | **IRB 120** |
| --- | --- | --- | --- | --- |
| **Carga [Kg]** | | 1.5 | 4 | 3 |
| **Alcance [m]** | | 0.37 | 0.58 - 0.475 | 0.58 |

Concluimos que la mejor opción era el **IRB 1100-4/0.475** debido a su alcance y su capacidad de carga máxima. Además, este robot cumple con la norma DIN EN ISO 14644-1,-14, siendo clasificado como ISO 4, y aprobado por IPA. Esta es una certificación fundamental a tener en cuenta debido a que el robot estará rodeado de alimentos.

Estás características junto a sus capacidades de control y de carga lo hacen óptimo para aplicaciones industriales tales como pintura y soldadura, empaquetado, atornillado, carga y descarga, sellado, etiquetado y otras.

## Especificaciones técnicas del IRB 1100-4/0.475

Es importante señalar que, aunque también consideramos el modelo 0.58 del IRB 1100, la diferencia más notable entre ambos radica en su alcance. En el caso de nuestra aplicación, el alcance proporcionado por el modelo 0.475 fue considerado adecuado y suficiente para llevar a cabo de manera efectiva y eficiente las tareas de preparación de bebidas de Latte Art.

A continuación se mostrarán dos tablas donde muestran las especificaciones técnicas que se tuvieron que tener en cuenta para el desarrollo del proyecto:

| Robot IRB 1100-4/0.475 | | |
| --- | --- | --- |
| Alcance [m] | | 0,475 |
| Carga máxima [Kg] | | 4 |

| Movimiento | | | | |
| --- | --- | --- | --- | --- |
| Articulación | Límite de posición articular | | Límite de velocidad articular | |
| q1 | +230º a -230º | | 460 º/s | |
| q2 | +113º a -115º | | 380 º/s | |
| q3 | +55º a -205º | | 280 º/s | |
| q4 | +230º a -230º | | 560 º/s | |
| q5 | +120º a -125º | | 420 º/s | |
| q6 | +400º a -400º | | 750 º/s | |

Como se puede observar en la tabla de movimientos, todos los grados de libertad tienen movimientos articulares rotativos, ninguno es traslacional.

## Espacio de trabajo

En la “figura 3” se muestra el espacio de trabajo del robot, el cual ha sido calculado en Matlab:

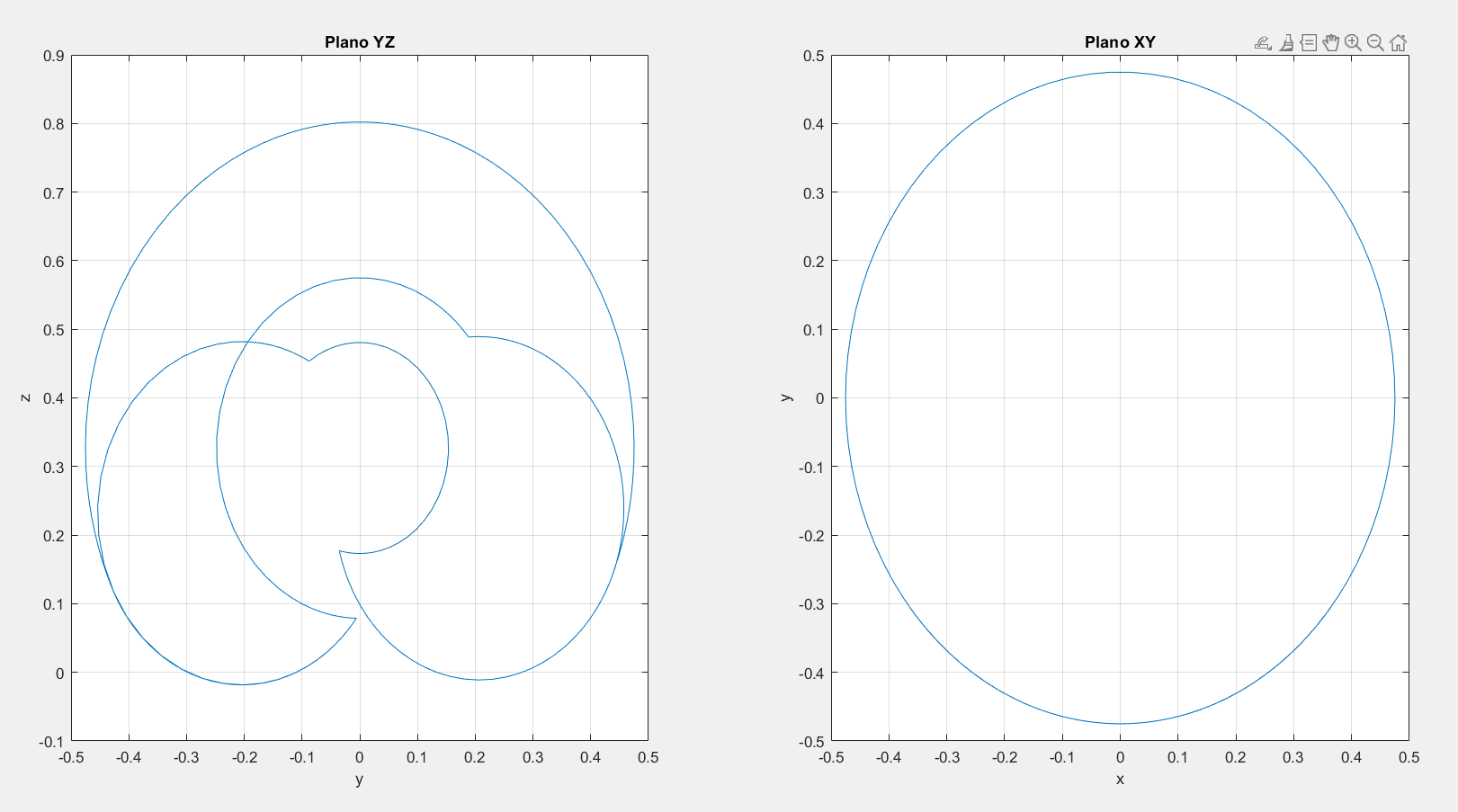


Figura 3 - Espacio de trabajo calculada

Se ha comprobado este espacio con el proporcionado en el datasheet que se muestra en la figura 4:

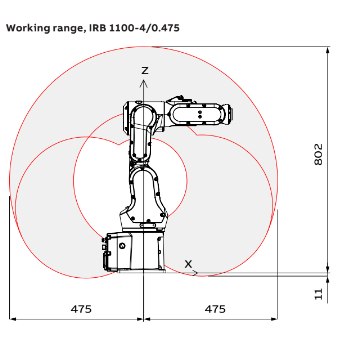


Figura 4 - Espacio de trabajo del fabricante

El robot puede ser equipado con algún controlador de la serie OmniCore, específicamente, el C30, C90 o el E10. Para nuestra aplicación es preferible el E10(figura 5), ya que los otros controladores están orientados a un control más industrial, incluyendo incluso resistencia a los ambientes agresivos. Sin embargo, no es necesaria tanta robustez, por lo que el E10 brinda un excelente compromiso entre costo y ahorro. este controlador debe ser alimentado por una tensión de alterna de 100V a 230V , y no tiene fusibles incorporados, por lo que deben agregarse externos. El fusible correspondiente para trabajar con el IRB 1100 debe ser de 10A y de clase K. Además, el controlador también es sensible a descargas electroestáticas, por lo que deben tomarse las precauciones correspondientes (por ejemplo, alfombra aislante alrededor o conexión del chasis a tierra).

Figura 5

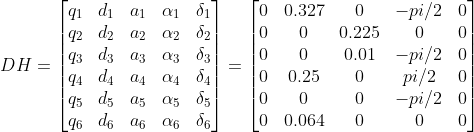
Este robot se puede calibrar mediante 3 metódos distintos: calibración de ejes, CalibWare u optimización de la muñeca. Dentro de estos tipos, es más apropiada la calibración de ejes detallada por el FlexPendant. El equipo necesario para realizar esta calibración se incluye como kit de herramientas.

# Robot

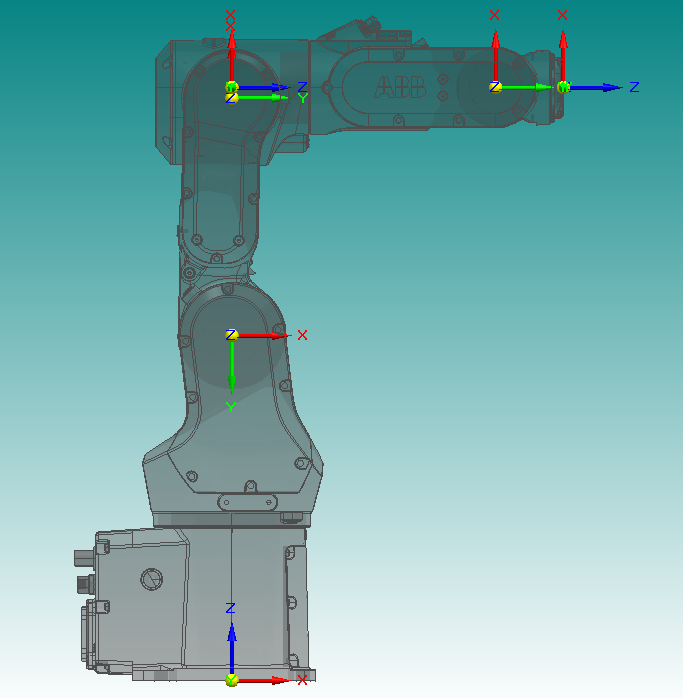
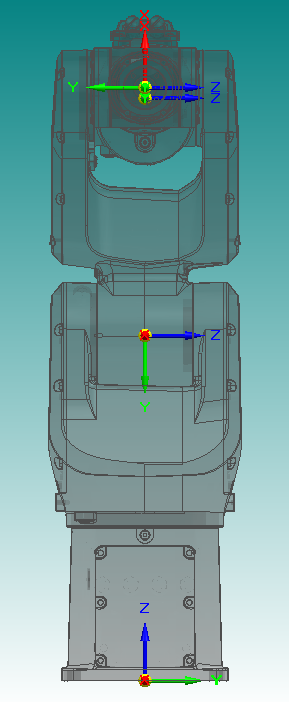
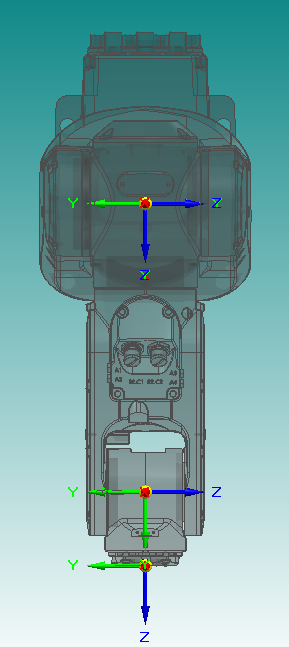
## Cinemática directa

Para crear la matriz DH, se recopilaron todos los datos necesarios a partir del datasheet del robot, lo que incluyó parámetros como las longitudes de los eslabones, los ángulos de rotación y las distancias entre los sistemas de referencia. Una vez que se obtuvieron estos datos, se pudo definir la matriz DH, que establece las transformaciones geométricas entre los sistemas de referencia a medida que se mueve a lo largo de las articulaciones del robot.

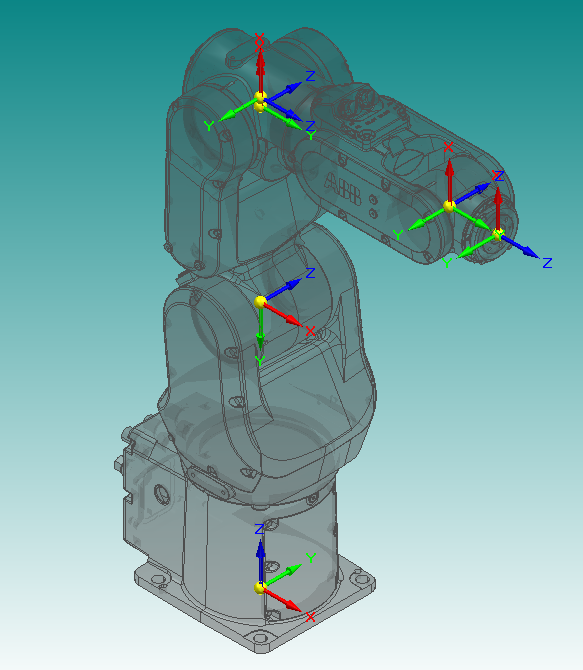
La matriz DH se configuró de la siguiente manera:



Para una comprensión visual más clara de cómo se ubican los sistemas de referencia en el robot, se empleó una representación gráfica que mostraba la disposición de los sistemas de referencia en las diferentes articulaciones del robot (figura 6). Esta visualización se logró exportando los archivos en formato **.step** a la herramienta de diseño **SolidEdge**. De esta manera, se ubican los ejes de manera eficiente en las posiciones y orientaciones elegidas, lo que facilitó la creación de la matriz DH y la comprensión de la cinemática del robot.

**Vista lateral Vista frontal Vista superior**



**Vista Isométrica**

Figura 6 - Robot y sistemas de coordenadas

A continuación, en la “figura 7”, se muestran los sistemas de referencia graficados en Matlab:

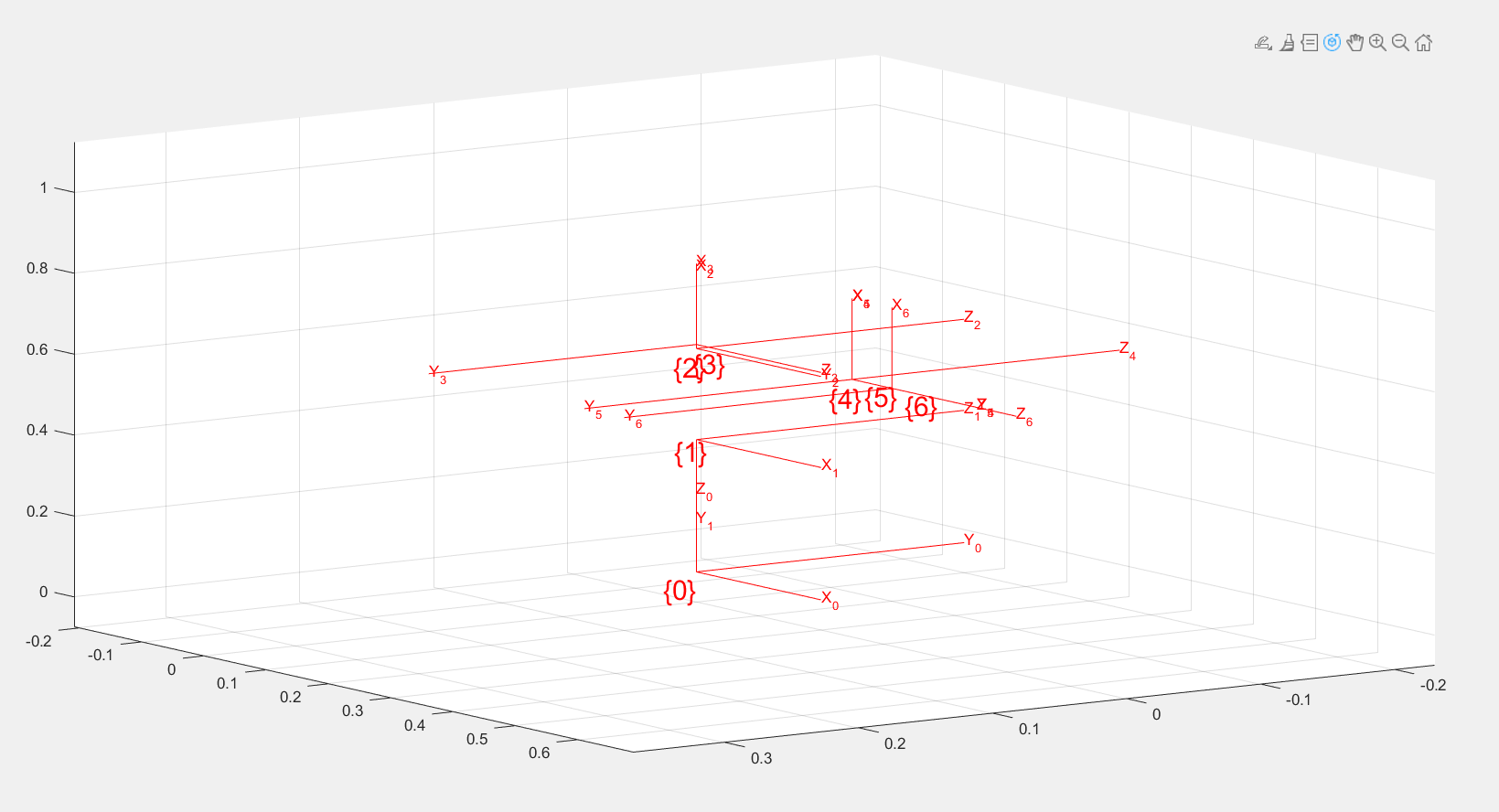


Figura 7 - Sistemas de referencia calculados

Debido a que se utilizó la convención éstandar de Denavit Hartenberg, el problema de transformar coordenadas del espacio articular (q1,...,q6) al espacio cartesiano (x,...,𝜸), quedó reducido a una serie de multiplicaciones matriciales usando matrices de transformación lineal homogénea.

## Cinemática inversa

El problema de cinemática inversa consiste en transformar las coordenadas del espacio cartesiano, (x;y;z;α;𝛽;𝛾), o en otras palabras, una matriz de transformación lineal homogénea 0T6, al espacio articular (q1;q2;q3;q4;q5;q6).

Para abordar la cinemática inversa del robot, se probó de dos formas diferentes. La primera fue creando una función personalizada, y la segunda fue utilizando la función **ikine** del toolbox. Finalmente optamos por la primera forma nombrada, usando la función personalizada (**invkine\_IRB1100.m**).

Para abordar la cinemática inversa se eligió el método de Pieper, que se destaca por su capacidad para desacoplar el robot en dos partes: una parte relativa a la posición del efector final y otra parte que se centra en la orientación. Fue posible implementar está solución ya que el robot cuenta con 6 ejes y una muñeca esférica. Esto simplificó los cálculos al permitir el uso de soluciones geométricas para cada aspecto. Esencialmente, las coordenadas articulares de una parte del robot se utilizan para determinar la posición del efector final, mientras que la otra parte se encarga de la orientación.

Este enfoque divide el desafío de la cinemática inversa en dos problemas más manejables, lo que facilita la obtención de soluciones precisas. La función personalizada desarrollada para este propósito se basa en este método, lo que garantiza que el robot pueda alcanzar correctamente la posición y orientación deseada en sus tareas de Latte Art.

Para el desarrollo de este método se tuvieron dos consideraciones:

* La distancia entre la articulación 6 y 4 se mantiene siempre constante debido a que es muñeca esférica, por lo tanto se calculó el origen de la muñeca en función de la matriz T y los parámetros de DH del robot

𝑝̅𝑐 = 𝑑̅6 - d6.𝒂̅𝟔

Donde:

* 𝑝̅𝑐: vector de posición de la muñeca respecto del origen
* 𝑑̅6: vector de posición del extremo final respecto del origen (dato de la matriz de transformación homogénea del extremo)
* 𝑑6: parámetro de DH que indica la distancia entre el extremo y la muñeca.
* 𝒂̅𝟔: versor que indica la dirección del extremo, y por lo tanto, la dirección entre el extremo y la muñeca (dato de la matriz de transformación homogénea del extremo).
* En el IRB1100 hay un codo fijo el cual tuvimos que tener en cuenta para el cálculo de la cinemática inversa como se puede apreciar en la “figura 8”.

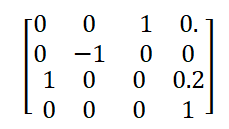


Figura 8

### Validación

La validación de la función utilizada se puede realizar utilizando un vector de prueba, verificando que todas las matrices obtenidas por cinemática directa de los vectores solución resultan iguales, y comprobando que el vector de prueba figura entre los vectores solución.

Por ejemplo, para cuando el robot se posiciona enfrente de la lanceta, el vector de coordenadas resulta qL = [0.0000, 1.0310, 0.3335, 0.0000, -1.3645, -0.0000], para el cual la matriz 0T6 es:



Si se pasa está matriz como argumento a la función “**invkine\_IRB1100.m**”, se obtienen los siguientes 8 vectores solución:

|  | | Sol1 | Sol2 | Sol3 | Sol4 | Sol5 | Sol6 | Sol7 | Sol8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| q1 | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | -3.1416 | -3.1416 | -3.1416 | -3.1416 |
| q2 | | 1.0310 | 1.0310 | 3.0379 | 3.0379 | 3.2452 | 3.2452 | 5.2522 | 5.2522 |
| q3 | | 0.3335 | 0.3335 | -3.3952 | -3.3952 | 0.3335 | 0.3335 | -3.3952 | -3.3952 |
| q4 | | 0.0000 | -3.1416 | 3.1416 | 0 | -0.0000 | 3.1416 | -3.1416 | 0 |
| q5 | | -1.3645 | 1.3645 | -0.3572 | 0.3572 | -0.4372 | 0.4372 | -1.2845 | 1.2845 |
| q6 | | -0.0000 | 3.1416 | -3.1416 | -0 | -3.1416 | 0.0000 | 0.0000 | -3.1516 |

Estos vectores se probaron en la función ikine del toolbok de Peter Corke, y arrojaron todos la misma matriz 0T6, la cual coincidió con la matriz utilizada como dato. Así también, se ve que en este caso, el vector Sol1 coincide con el vector qL esperado. En el desarrollo realizado en las distintas funciones utilizadas, se utilizó la función “**invkine\_IRB1100.m**” en conjunto con vectores semillas en cada llamado (un vector por llamado), los cuales definian la posición anterior. De este modo, se obtuvo solo un vector solución en cada caso, en específico, el más próximo a la posición anterior.

# Planificación de trayectoria

Para la planificación de las trayectorias se tuvieron en cuenta las acciones tales como tomar la jarra, emulsionar y texturizar leche, golpear jarra para sacar burbujas, hacer un dibujo con forma de corazón en la taza de café y luego guardar la jarra donde estaba inicialmente.

Para el desarrollo se utilizaron diferentes interpolaciones usando las funciones ctraj, jtraj y msjtraj. El método ctraj se utilizó solamente en algunas trayectorias (específicamente en las funciones buscarJarra.m y emulsionar.m) para comparar con los resultados de jtraj y mstraj, sin embargo se ha llegado a la conclusión que para nuestra aplicación buscaremos evitar el uso de ctraj debido a que puede ocasionar sobreaceleraciones y es más propenso a pasar por singularidades. Además ctraj ocasiona movimientos lineales, los cuales para nuestra aplicación no son necesarios. Dichas comparaciones se puede apreciar en los gráficos que se mostraran a continuación (figura 9):

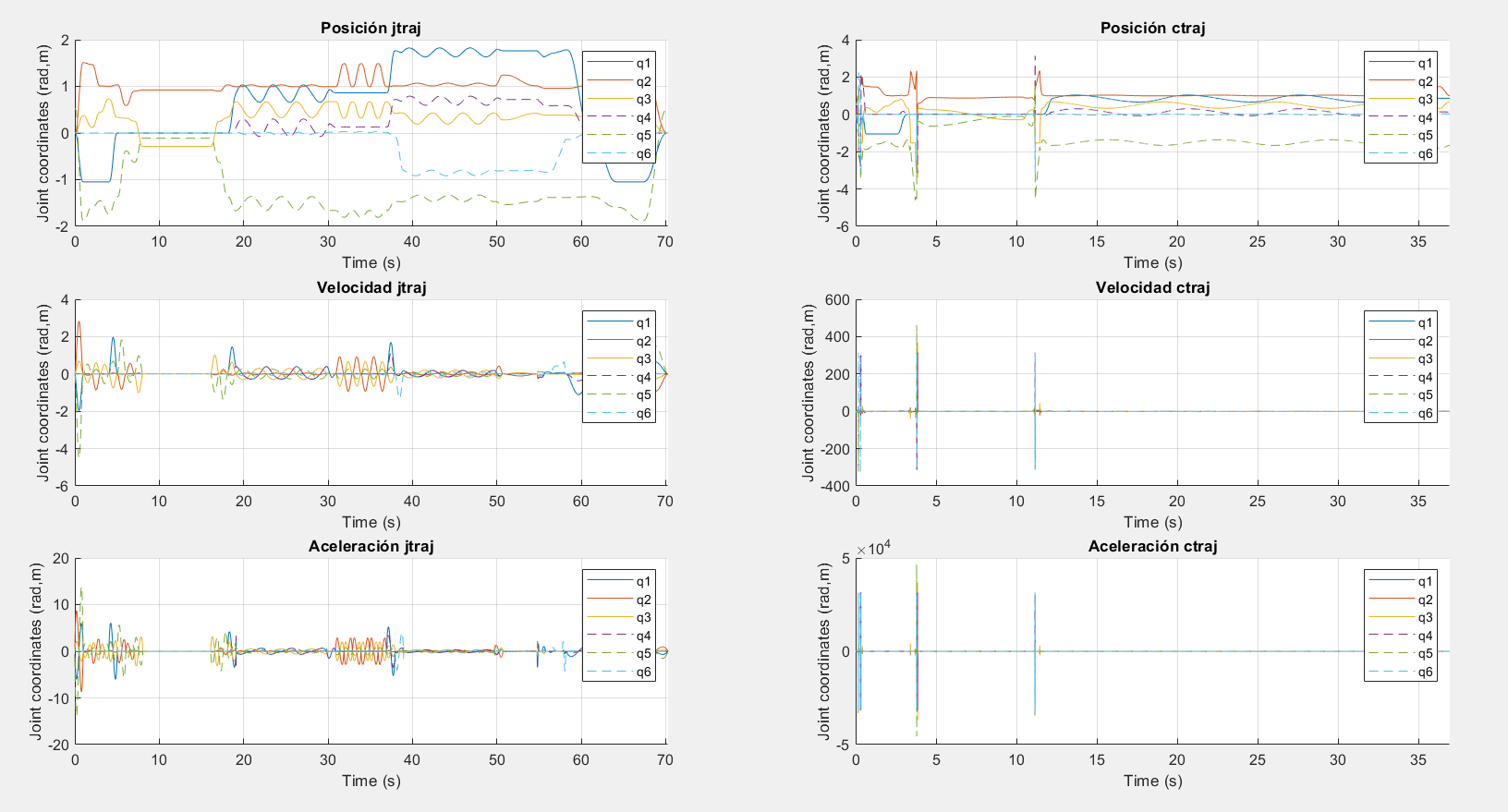


Figura 9 - Variables articulares

Como se puede ver, tiene cambios más abruptos de posición en ctraj debido a que pasa por por puntos singulares y esto produce sobreaceleraciones.

Al graficar las respuestas de la zona donde hace las circunferencias para texturizar la leche, se puede apreciar una gran diferencia entre el método mstraj respecto del ctraj, ya que el mstraj funciona con aproximaciones y nunca termina de frenar en cada punto, sino que pasará relativamente cerca. Para la función ctraj pasará por cada punto interpolado y en dichos puntos frenará y se producirán sobreaceleracions al pasar al otro punto (figura 10):

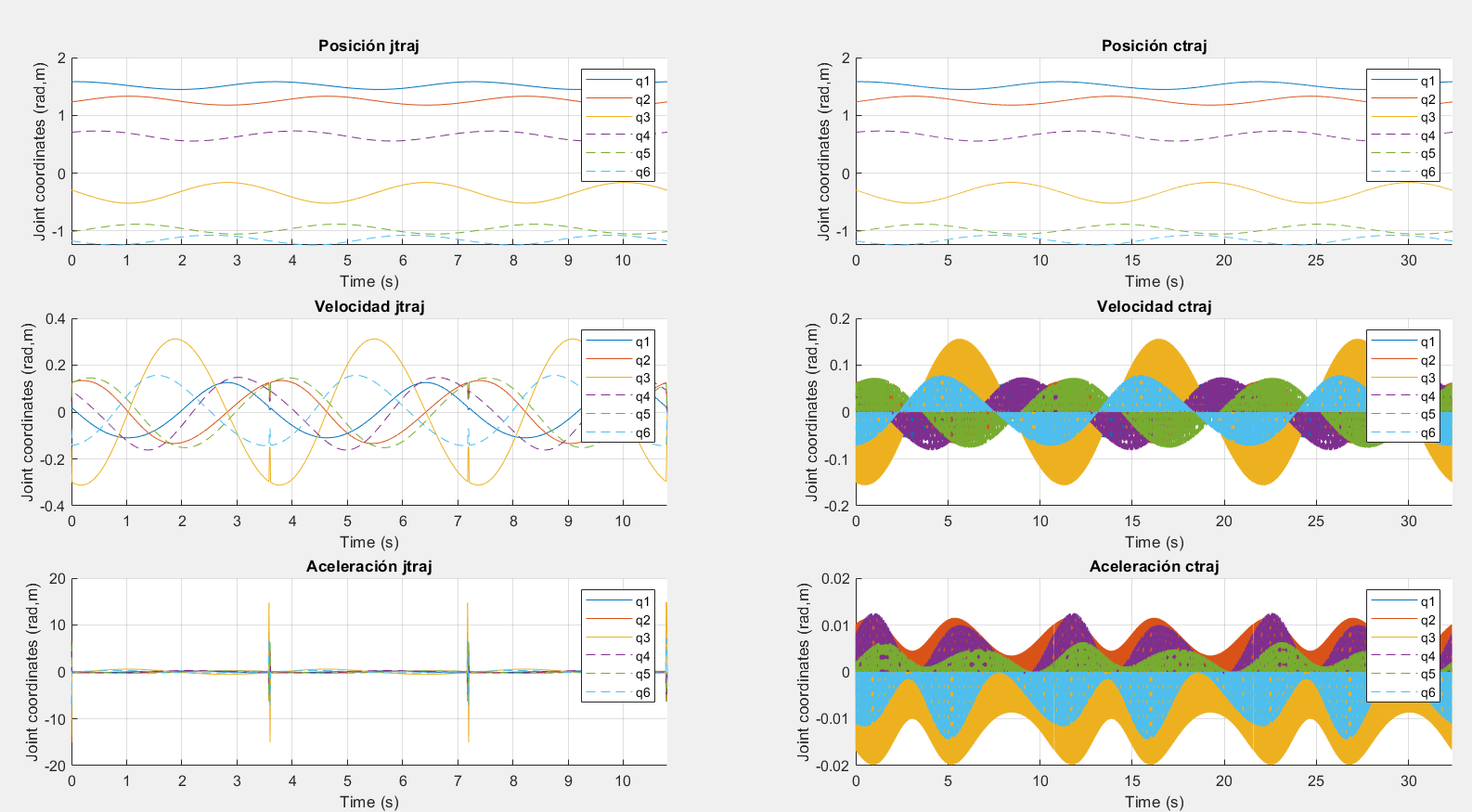


Figura 10 - Variables articulares en detalle

En el siguiente gráfico (figura 11) se muestran las posiciones, velocidades y aceleraciones cartesianas absolutas con los diferentes métodos para **buscarJarra.m** y **emulsionar.m**

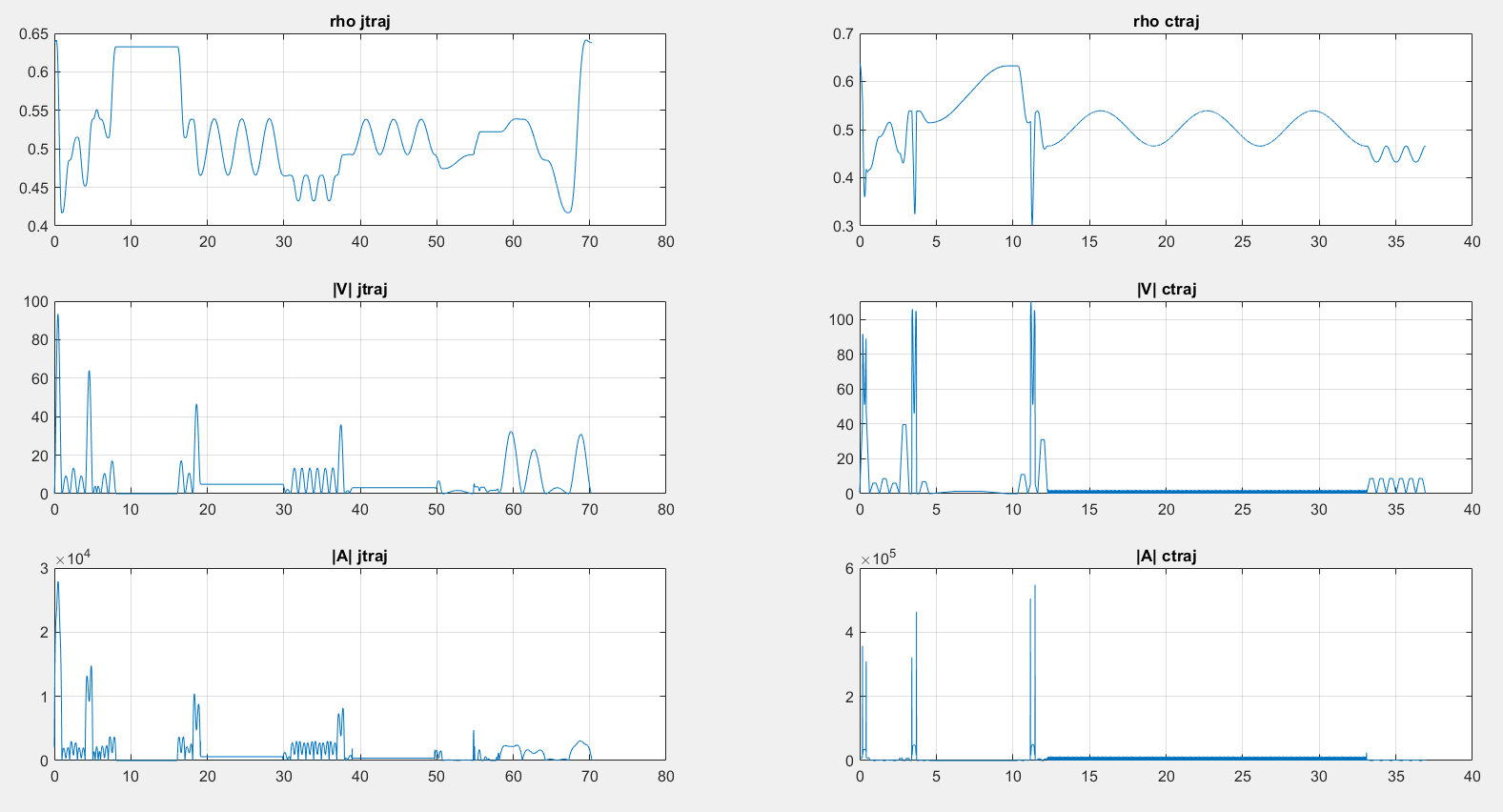


Figura 11 - Variables cartesianas

En la “figura 12” se puede apreciar la trayectoria del robot:

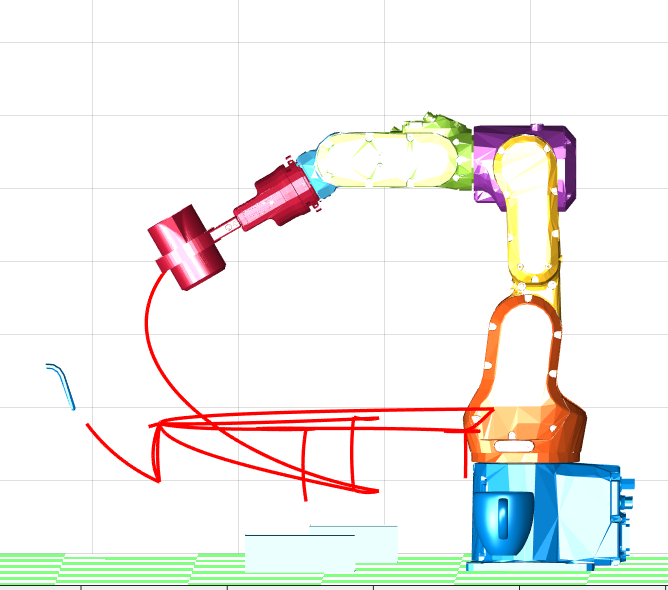
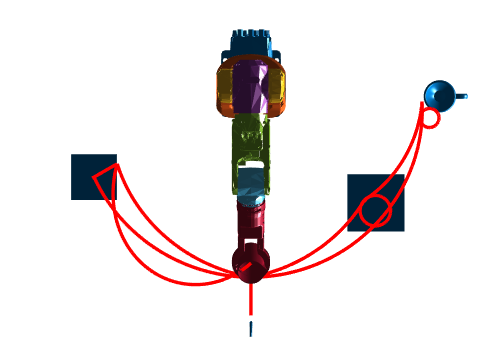


Figura 12 - Trayectorias en MatLab

Como se comentó anteriormente, se omite el uso de ctraj debido a las sobreaceleraciones que ocasiona, por lo tanto todas las trayectorias se trabajaron con jtraj y mstraj. Específicamente se utilizó msjtraj para las dos circunferencias generadas ya que no necesitábamos precisión de movimiento y no queriamos que se frenara en cada punto, y jtraj para el resto de los movimientos. La función mstraj solamente se ve localizada en la función “**Texturizado\_Giro.m**”.

## Relación de velocidades

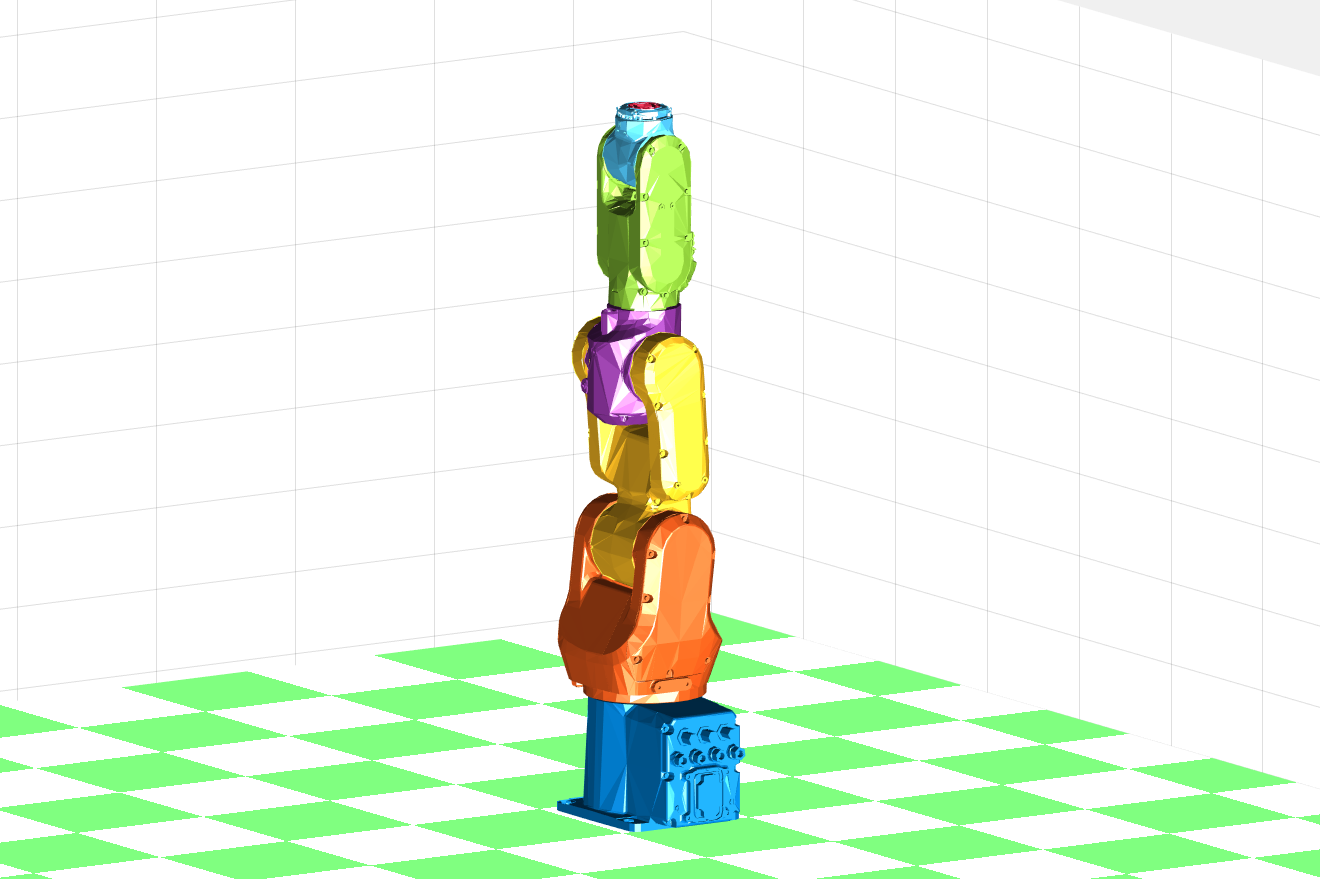
Se verificó en los gráficos de velocidades obtenidos en las figuras anteriores, que las velocidades de cada articulación no superaran los límites del fabricante.

También se intentó utilizar el Jacobiano para encontrar posibles singularidades, sin embargo su obtención de manera simbólica es demasiado laboriosa incluso para MatLab, de manera que no se pudo utilizar como método para encontrar posibles soluciones.

Nos vimos entonces obligados a explorar un poco más en detalles las posiciones y posibles movimientos del robot para intentar determinar aquellos puntos problemáticos.

El método implementado para probar los puntos consistió en una prueba de “fuerza bruta”, se realizaron bucles de 10000 iteraciones para probar los puntos, intentando de esta manera aumentar la robustez de cada prueba. En cada iteración, se actualizaba la semilla para números aleatorios, y se configuraba una vector de coordenadas articulares con todos sus parámetros de manera aleatoria, excepto aquel o aquellos que querían probarse.

La primera prueba consistió en forzar q5 a un valor 0 y permitiendo variar todas las demás articulaciones (figura 13), debido a que es conocido que esta configuración resulta problemática en robots con muñecas esféricas al alinear los ejes 4 y 6. En las 10000 iteraciones, el determinante del Jacobiano se anuló las 10000 veces, permitiendo asegurar que cualquier configuración con q5=0 resulta en un punto singular.

Figura 13

Luego, se intentó determinar mediante examinación, cuales configuraciones podrían presentar una problemática similar. Se interpretó que el problema se presentaba debido a la alineación de de dos ejes, lo que causaba que ciertos movimiento presentarán dualidad, al ser posibles de obtener mediante giro de un eje, el otro, o combinación de ambos de manera indefinida.

Se llegó a la conclusión de que una combinación muy específica de q2 y q3 podría resultar en una alineación de ejes 1 y 4, en particular, aquella combinación en donde la componente horizontal del vínculo 2 fuera anulada por la componente horizontal del vínculo 3. Se determinó que esta combinación era :

siendo L2 y L3 la longitud de los vínculos 2 y 3 respectivamente. Se repitió el procedimiento explicado para q5 pero forzando ahora la combinación mencionada anteriormente y permitiendo todas las demás coordenadas (incluida q5), y se obtuvo nuevamente que el determinante del Jacobiano se anulaba en todos los casos. Esto confirmó el análisis.

Cabe aclarar, que la singularidad q2-q3 limita virtualmente el espacio de trabajo, ya que si bien todos los puntos del espacio de trabajo son puntos singulares, esta combinación de coordenadas resulta en un punto en específico muy cercano al espacio de trabajo pero no coincidente con el límite, que es también en un punto singular.

Estas fueron las consideraciones analíticas que se tuvieron en cuenta para diseñar la trayectoria de la tarea, a sabiendas de que era posible encontrar otros puntos singulares durante el desarrollo de la misma.

# Herramienta

El brazo robótico debe tener una herramienta para lograr la acción de “pick and place” de la jarra. Para esto hemos seleccionado el gripper **LWR50L-03-00003-A** (figura 14) el cual se puede ver en la figura siguiente:



Figura 14 - Gripper

Link:

<https://www.zimmer-group.com/es/tecnologias-y-componentes/componentes/tecnologia-de-manipulacion/match-cambiador-de-herramientas-con-efector-final/lote-de-efectores-finales/hrc/pinzas-paralelas/lwr-hrc-03/productos/lwr50l-03-00003-a>

Es posible acoplar a esta herramienta una impresión 3D en sus extremos para que se adapte a la forma de la jarra (Figura 15):

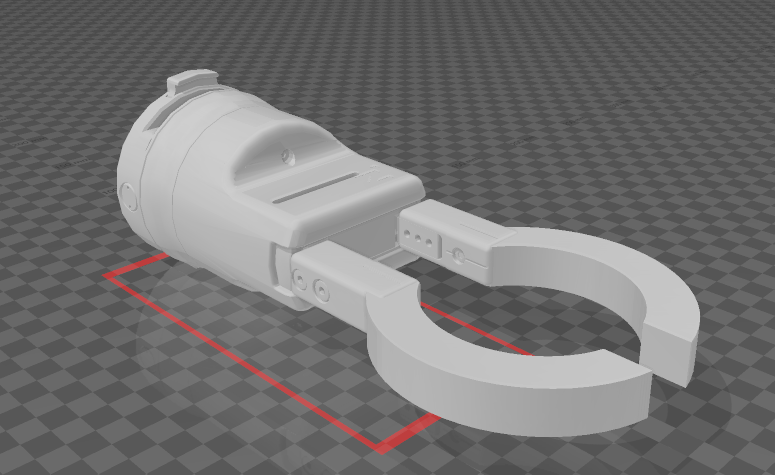


Figura 15 - Gripper con extensión

Para los cálculos en el código se tuvo en cuenta el punto de trabajo de la herramienta, por lo tanto se modificó el vector R.tool, teniendo un vector igual a [0,0,0.182].

También se tuvo en cuenta que la jarra no posee manija (figura 16), ya que esto facilita el agarre y es relativamente sencillo sacar la manija a una jarra (muchas cafeterías lo implementan para mayor facilidad del barista)

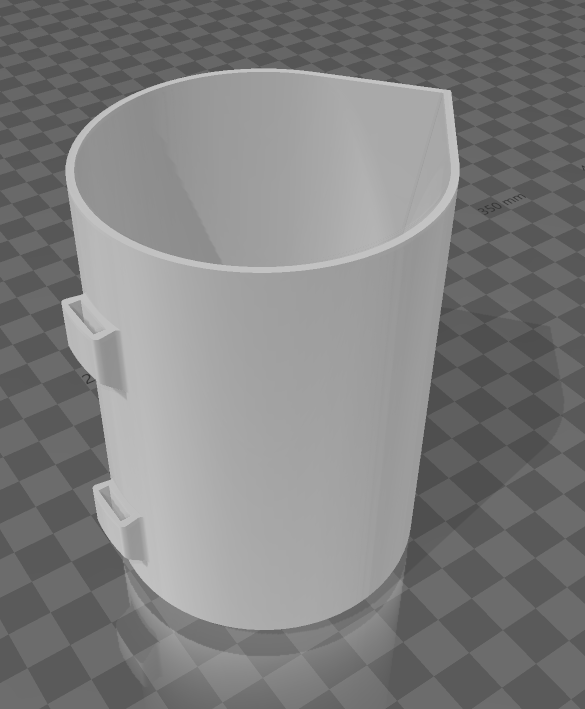


Figura 16 - Jarra modelada

# Sensores

## Sensores externos

Se considera fundamental la incorporación de un sensor de fuerza piezoresistivo en el gripper del robot. Esto se debe a varias razones que hacen que esta elección sea particularmente ventajosa.

* Control Preciso de la Fuerza de Agarre:
  + Un sensor de fuerza piezoresistivo permitirá medir la fuerza con la que la pinza del robot agarra la jarra de leche. Esto es esencial para garantizar que la presión ejercida sobre la jarra sea la adecuada. Además, se puede programar el robot para detener la aplicación de fuerza una vez que se alcance un umbral predeterminado, evitando así la deformación de la jarra. Esto es de suma importancia para evitar deformaciones de la jarra
* Adaptación a Cambios en la Carga:
  + En el proceso de verter leche en el café, el peso de la jarra disminuirá gradualmente a medida que se reduce el volumen de leche. Un sensor de fuerza piezoresistivo permitirá al robot ajustar la fuerza de agarre en función de esta reducción de carga. De esta manera, se puede mantener un control preciso en todo momento y evitar aplicar una fuerza excesiva a medida que disminuye la carga.

## Sensores internos

Además del sensor de fuerza en el gripper, es recomendable aprovechar los sensores internos del propio robot. Por ejemplo, el detector de colisiones del robot es esencial para garantizar la seguridad en el entorno de trabajo. Cuando el robot entra en colisión, se producen picos de corriente en los bobinados, que son detectados y leídos por el software del robot como señales de colisión. Esto es crucial para evitar daños al robot y, lo que es más importante, garantizar la seguridad de las personas que puedan ingresar al área de trabajo.

# 

# Glosario

\*1) En este proceso se mezcla la leche con el vapor de agua, produciendo así la cremosidad necesaria que necesita la leche y la temperatura correcta

\*2) En este proceso se suele mezclar la leche para homogeneizar su textura y sacar todas las burbujas

Referencias

1. Robot modelling and control - Spong
2. IRB 1100, datasheet - ABB
3. Product Manual IRB 1100 - ABB
4. IRB 1010, datasheet - ABB
5. Product Manual IRB 1010 - ABB
6. IRB 120, datasheet - ABB
7. Product Manual IRB 120 - ABB
8. Product Manual OmniCore E10 - ABB
9. OmniCore E10 controller datasheet - ABB
10. Product Manual, OmniCore 30 - ABB
11. LWR50L-03-00003-A datasheet - Zimmer Group