

Planificación y Generación de Trayectorias

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

Ejercicio 1: Generación de trayectoria entre 2 puntos articulares.

1. Trabajando en Matlab con el toolbox RTB interpole las posiciones articulares q_0 y q_1 mediante la función "jtraj". Use 3 segundos y una décima de paso para la discretización.

$$q_0 = [0, -\pi/2, 0, 0, 0, 0]$$

$$q_1 = [-\pi/3, \pi/10, -\pi/5, \pi/2, \pi/4, 0]$$

2. Adopte el robot "FANUC Paint Mate 200iA" y realice una animación cinemática entre los puntos q_0 y q_1 interpolados en 1) con la función "SerialLink/plot". Use la definición del robot dada en el TP4.
3. Grafique las variables articulares con respecto al tiempo usando la función "qplot".

Ejercicio 2: Generación de trayectorias entre 2 puntos cartesianos, interpolación articular.

1. Interpole las posiciones articulares correspondientes al extremo del robot en los puntos siguientes:

$$P1 = [0, 0, 0.95]$$

$$P2 = [0.4, 0, 0.95]$$

Note que es un desplazamiento de 0.4 unidades en dirección X, y que las coordenadas son cartesianas, se debe interpolar en el espacio articular. Utilice la orientación del extremo dada por la siguiente posición articular:

$$qq = [0, -\pi/2, -\pi/4, 0, \pi/4, 0]$$

Tenga en cuenta que necesitará resolver la cinemática inversa. Use el método "ikine" para lo cual es recomendable usar "qq" como vector semilla ya que se encuentra cerca de las posiciones que deberá tomar el robot en la trayectoria. En lugar de un vector de tiempo, elija 100 puntos de discretización.

2. Realice una animación cinemática del robot del ejercicio 1 y grafique las variables articulares.

Ejercicio 3: Generación de trayectorias entre 2 puntos cartesianos, interpolación cartesiana.

1. Interpole los mismos puntos del ejercicio anterior, pero en el espacio de tarea y en una línea recta. Haga uso de la función "ctrj". Analice los parámetros de entrada y de salida de CTRAJ y use 100 puntos de discretización.
2. Realice una animación cinemática y grafique las variables articulares. Considere que para esto es necesario conocer el valor de las variables articulares. Haga un correcto uso de la función "ikine". Tenga en cuenta que "qq" es un vector cercano a la

trayectoria en cuestión, y que la función “ikine” acepta un arreglo de matrices de transformación homogénea.

Ejercicio 4 (**obligatorio**): Otras gráficas de una trayectoria interpolada.

1. Use la función “jtraj” para obtener la posición, velocidad y aceleración articulares de la trayectoria del ejercicio 2, y grafique las 3 interpolaciones respecto del tiempo en figuras separadas.
2. Aplique alguna derivada numérica y grafique la posición, velocidad y aceleración articulares del ejercicio 3, en las figuras correspondientes al punto anterior, para poder compararlas. Verifique la posición, velocidad y aceleración iniciales (debe coincidir) y explique brevemente lo que puede apreciar al comparar las gráficas. Tenga en cuenta el escalado de las gráficas.
3. Grafique las variables X, Y, Z del extremo del robot, respecto del tiempo, para ambos casos. Compare y extraiga conclusiones.
4. Grafique la variable Z respecto de X para ambos casos y verifique sus conclusiones anteriores. Tenga en cuenta una escala apropiada para los ejes.

Ejercicio 5: Generación de trayectorias entre varios puntos, frenando.

1. Proponga 4 puntos (articulares o cartesianos) y realice una animación de la trayectoria interpoladora mediante usos de la función “jtraj”.
2. Grafique posición, velocidad y aceleración articulares con respecto al tiempo.

Ejercicio 6: Generación de trayectorias entre varios puntos, puntos de paso (sin frenar).

1. Explore cómo resolver el problema anterior con la función “mstraj”.
2. Compare gráficas de resultados.
3. Saque conclusiones.

Ejercicio TF (**obligatorio**): Generación de trayectorias para el robot elegido como Proyecto Final.

1. Analice el problema de planificación y generación de trayectoria en el robot seleccionado para el Proyecto Final. Tenga en cuenta la aplicación.
2. Establezca requisitos generales y requisitos específicos.
3. Implemente **al menos dos soluciones diferentes**:
 - a. Establecer puntos en el espacio cartesiano, transformar al espacio articular, interpolar adecuadamente.
 - b. Establecer puntos en el espacio cartesiano, interpolar en espacio cartesiano, transformar al espacio articular.
4. Grafique los perfiles de posición, velocidad y aceleración, tanto en el espacio articular como en el de tarea, para ambas soluciones, de forma tal que se pueda apreciar las diferencias.
5. Seleccione la más adecuada para la aplicación. Justifique.

Algunas pruebas con curvas genéricas

Primero se probaron algunas trayectorias en el espacio como rectas a partir de 2 o 3 puntos, y espirales cuyas proyecciones de las circunferencias se situaban en el plano XY o XZ. En los scripts “Espiral.m”, “EspirallInterpolacionCartesiana.m”, “EspirallXZ.m”, “EspirallXZInterpolacionCartesiana.m” y “SimulacionRectasSolucionB.m” hay algunas trayectorias que se han generado a través de los dos caminos: el primero, discretizar las curvas en el espacio 3D en una cierta cantidad de puntos N, luego calcular la cinemática inversa para cada uno de esos puntos, y finalmente interpolar en el espacio articular. También se ha interpolado en el espacio cartesiano, para luego calcular la cinemática inversa de todos los puntos de la trayectoria y simular.

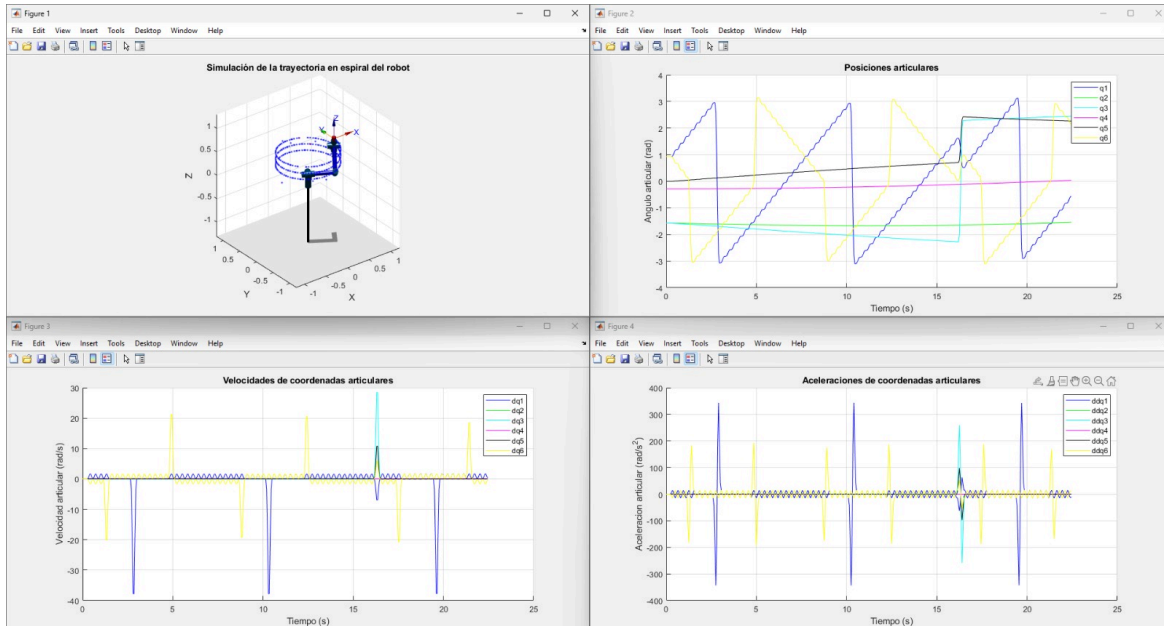
La diferencia clave entre los dos métodos de simulación de trayectorias, es que si se interpola en el espacio articular, estos vectores de coordenadas articulares pueden llevar a puntos y orientaciones del efector final que son incorrectas para la trayectoria. Esto se debe a los cambios grandes entre dos vectores de coordenadas articulares correspondiente a puntos sucesivos de la trayectoria y calculados con la función cinInversa.m, generada con los métodos geométricos planteados en el práctico 6B. Los vectores de coordenadas articulares intermedios a esos puntos sucesivos son calculados mediante interpolación y por lo tanto no se está colocando ninguna restricción o condición para que la matriz de transformación homogénea resultante de la cinemática directa de esos vectores satisfagan la trayectoria.

En cambio, cuando se interpola en el espacio cartesiano y luego se calcula la cinemática inversa de todos esos puntos obtenidos para la trayectoria en coordenadas cartesianas, el efector final del robot siempre sigue a la trayectoria en el espacio. El problema que vimos en los scripts donde se implementó la interpolación cartesiana, es que puede que en algún punto de la trayectoria, el codo cambie de posición. Para la mayoría de casos de posiciones y orientaciones finales del efector final del robot, y teniendo en cuenta límites articulares de $[-180,180]$, se observó que hay 8 combinaciones de coordenadas articulares que llevan al efector final a la posición y orientación correcta. 4 soluciones corresponden al codo posicionado en un punto particular, y otras 4 corresponden a otra posición del codo en el espacio. Estos dos puntos pertenecen a la circunferencia intersección de las dos esferas generadas en el planteo del trabajo práctico 6B: una con centro en el punto del hombro y radio igual a d_3 y la otra con centro en la muñeca y radio igual a a_4 .

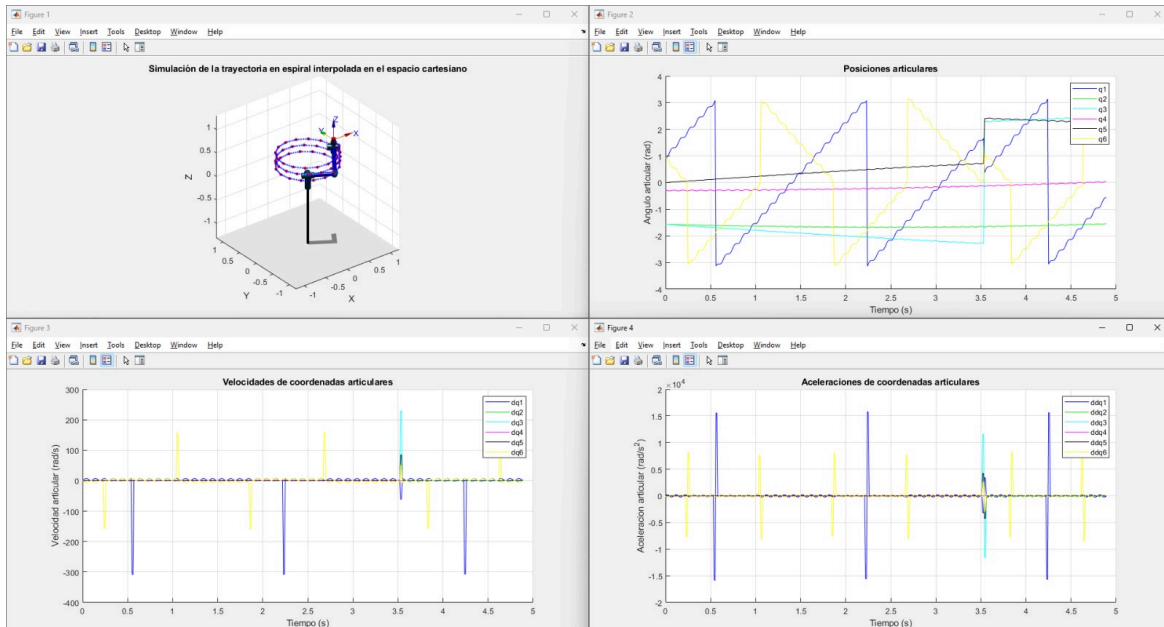
También se calcularon y graficaron las velocidades y aceleraciones articulares para cada caso particular.

Espirales

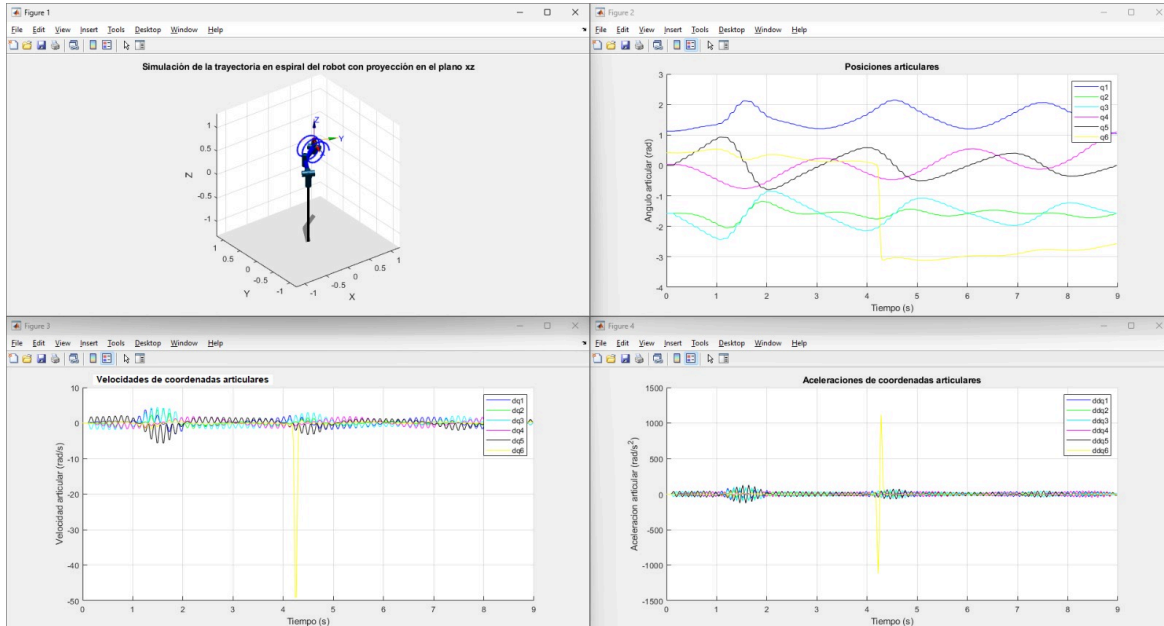
Interpolación en el espacio articular (Espiral.m)



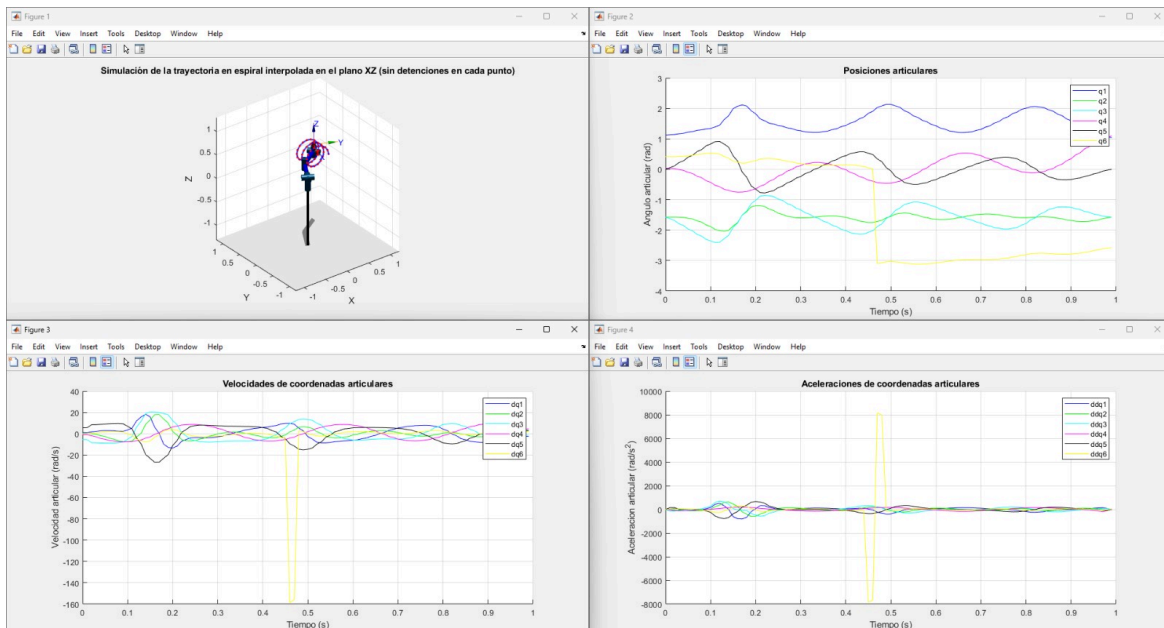
Interpolación en el espacio cartesiano (EspiralInterpolacionCartesiana.m)



Interpolación en el espacio articular (EspiralXZ.m)



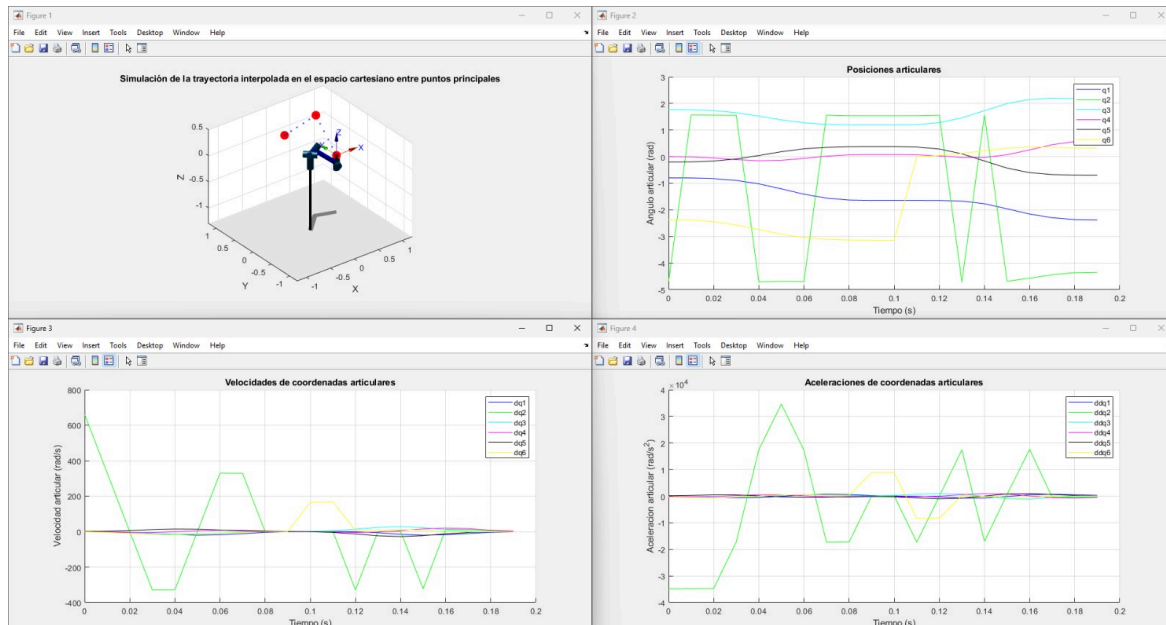
Interpolación en el espacio cartesiano (EspiralXZInterpolacionCartesiana.m)



Tanto en las espirales con circunferencia proyección en el plano X-Y como en las que tienen la circunferencia proyección en el plano X-Z, en el script se puede variar el punto medio, radio, cantidad de puntos de discretización, puntos de interpolación y cotas iniciales y finales en Z o Y según corresponda el tipo de espiral.

Rectas

Interpolación en el espacio cartesiano (SimulacionRectasSolucionB.m)



Implementación de Scripts para la Generación y Simulación de Trayectorias en un Brazo Robótico Servidor de Helado

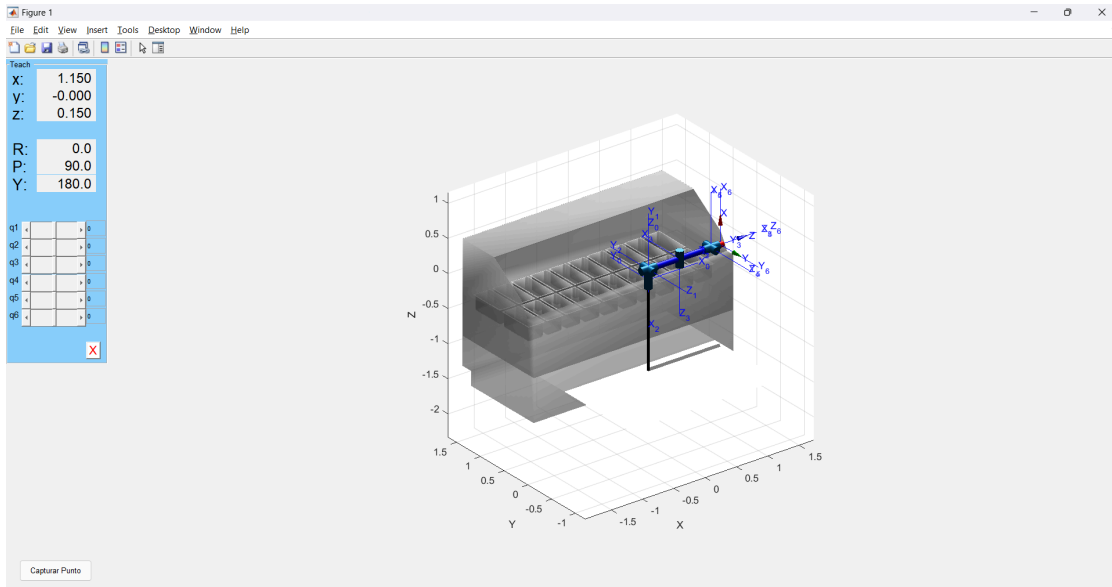
Para el control y simulación de un brazo robótico diseñado para servir helado, se han desarrollado dos scripts en MATLAB que permiten generar trayectorias y simular el movimiento del robot en el espacio 3D. Estos scripts utilizan interpolación cartesiana y cálculo de cinemática inversa para definir y ejecutar trayectorias en el espacio 3D del efector final del robot. A continuación, se describen los objetivos y funcionalidades principales de cada script: GenerarTrayectorias.m y SimulacionTrayectoriasArchivos.m.

Script GenerarTrayectorias.m

El objetivo de este script es permitir la generación y captura de puntos de una trayectoria definida por el usuario en un espacio de trabajo interactivo.

- **Carga y Configuración del modelo 3D:** Se carga el modelo de la vitrina de helados 3D para utilizarlo en la misma figura que la del brazo robótico y se carga también la configuración del robot que se encuentra en el script robot2.m. Luego de incluir los scripts cargarModelo.m y robot2.m se incluye la visualización de los sistemas de referencia de cada articulación y del efector final.
- **Interfaz de Usuario y Control Manual:** Utilizando la función R.teach(), se habilita una interfaz interactiva que permite al usuario manipular el robot manualmente. El usuario puede mover el brazo en tiempo real, mientras observa la configuración de cada articulación y los sistemas de coordenadas en pantalla.
- **Captura de Puntos:** En la interfaz, se incluye un botón "Capturar Punto" que, al ser presionado, guarda la matriz de transformación homogénea TTT del efector final en el momento de la captura. Esta matriz TTT representa la posición y orientación del efector en el espacio y se almacena en una lista de matrices.

- **Almacenamiento de Datos:** Al finalizar la sesión de captura, las matrices almacenadas se escriben en un archivo de texto, permitiendo su uso en simulaciones posteriores.



Script SimulacionTrayectoriasArchivos.m

Este segundo script se utiliza para leer los archivos de texto con las matrices capturadas, interpolar entre ellas y simular la trayectoria del robot mediante cálculos de cinemática inversa.

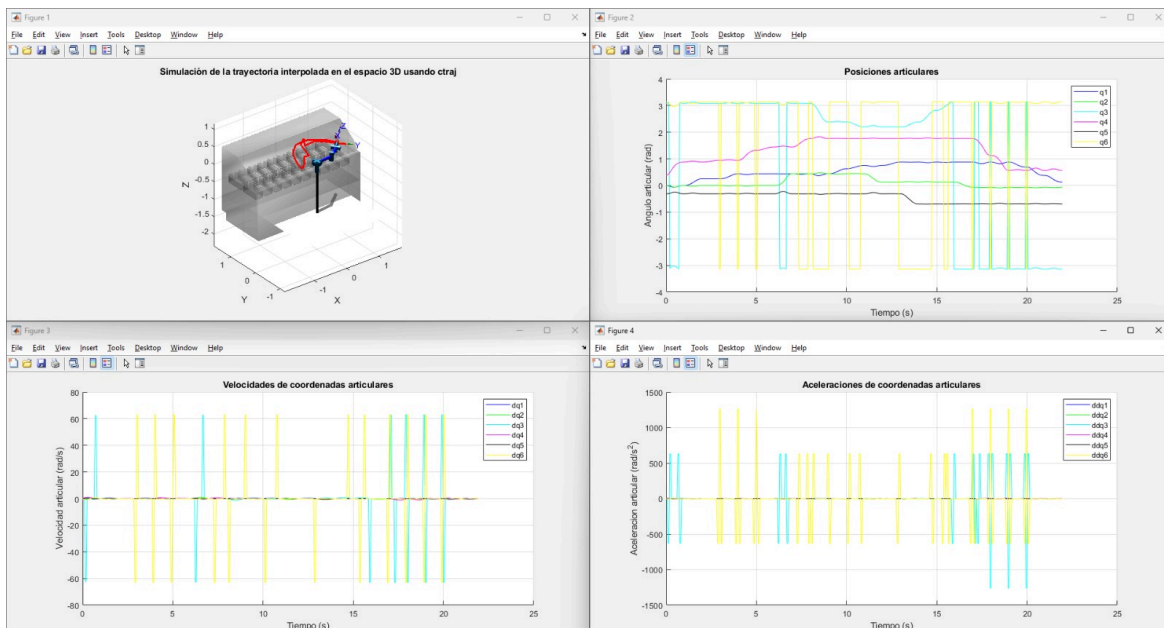
- **Selección de Trayectoria:** Al iniciar, el script muestra un menú que permite al usuario seleccionar entre cuatro trayectorias diferentes. Cada opción carga un archivo de texto específico con las matrices de transformación de la trayectoria correspondiente.
- **Interpolación de Matrices:** Las matrices de transformación homogénea leídas del archivo se interpolan usando la función `ctrj`, generando una trayectoria continua en el espacio 3D. Este proceso asegura una transición suave entre cada posición de la trayectoria.
- **Cálculo de Cinemática Inversa:** Para cada matriz interpolada, se calcula la configuración articular que permite al robot alcanzar la posición y orientación deseada del efector final. Esto se realiza usando la función `ikine` de MATLAB, aunque se probó también con la función `cinInversa`. La diferencia principal entre una función u otra se observa en los cambios repentinos de posición que toma el codo, o un recorrido suave pero con el codo mal posicionado a lo largo de toda la trayectoria con la función `cinInversa`.
- **Simulación de la Trayectoria:** La trayectoria calculada se simula gráficamente en MATLAB, mostrando el movimiento del robot en el espacio 3D. Además, se incluye una visualización de las posiciones articulares a lo largo del tiempo.

Visualización de Variables Articulares

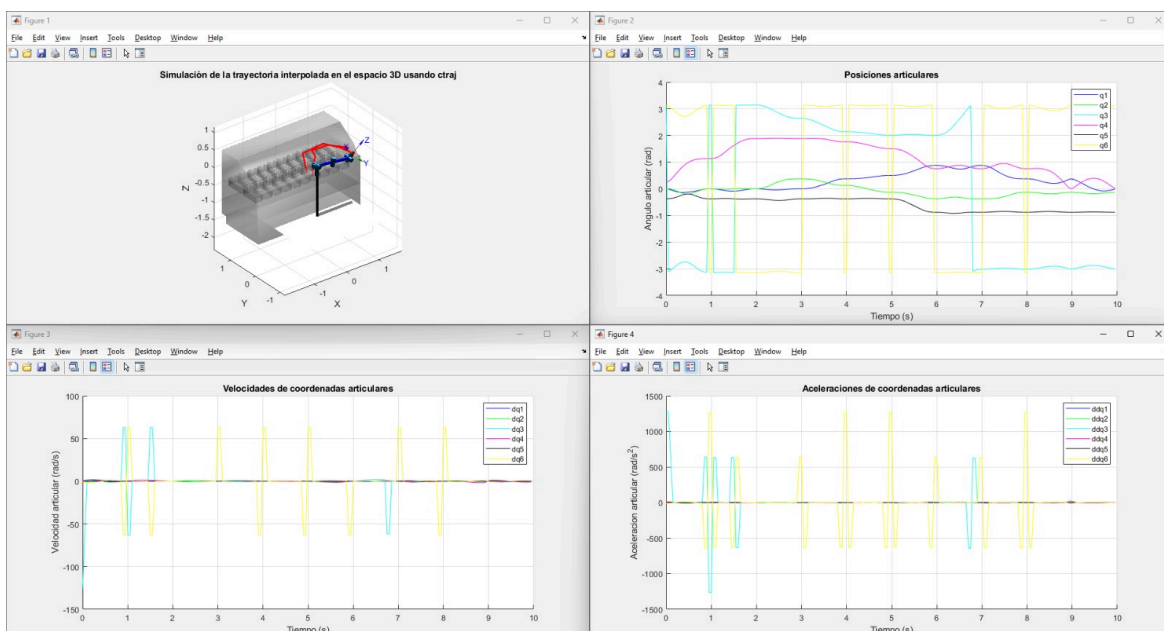
Para analizar en detalle el comportamiento del robot, se han agregado gráficos adicionales:

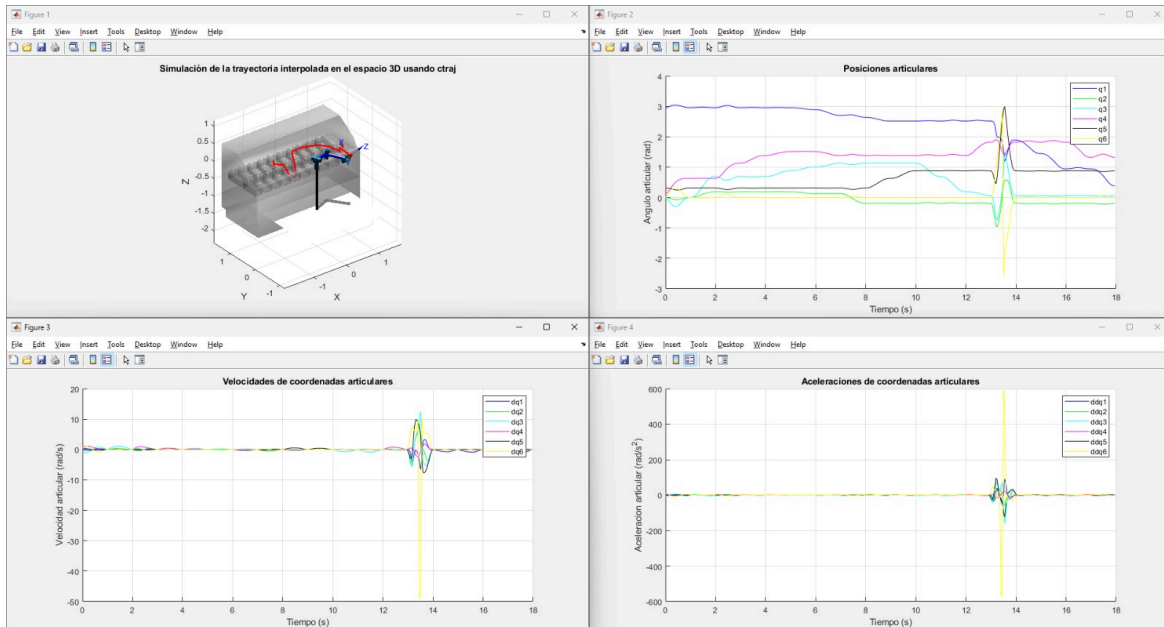
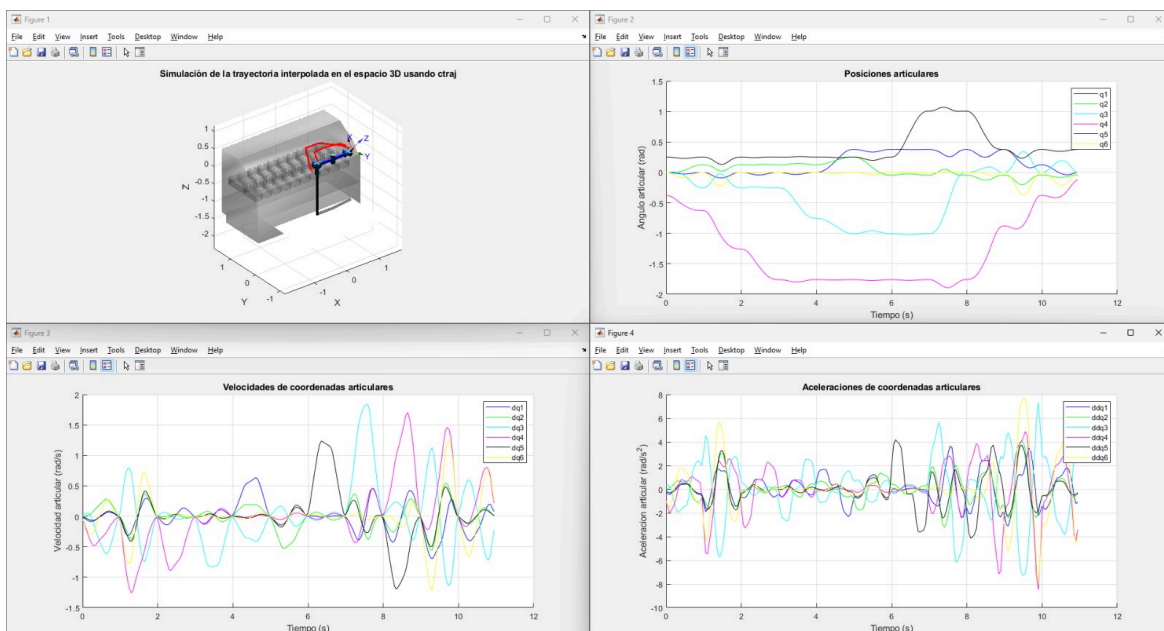
- **Posiciones Articulares:** Un gráfico muestra cómo varían los ángulos de cada articulación a lo largo de la trayectoria.
- **Velocidades Articulares:** Utilizando una derivada numérica propia (basada en la primera diferencia central), se calculan y grafican las velocidades de cada articulación.
- **Aceleraciones Articulares:** Similarmente, se calculan y grafican las aceleraciones de cada articulación, proporcionando información valiosa sobre el esfuerzo dinámico requerido.

Trayectoria 1:



Trayectoria 2:



Trajectory 3:**Trajectory 4:****Conclusión**

Con estos dos scripts, es posible generar trayectorias personalizadas y simular el movimiento de un brazo robótico en un entorno controlado, aplicando interpolación cartesiana y cinemática inversa para alcanzar posiciones y orientaciones específicas en el espacio 3D. Este proceso es crucial para validar y optimizar trayectorias en aplicaciones de robótica, como es el caso de nuestro robot servidor de helado.