



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD  
DE INGENIERÍA

## ROBÓTICA I - PROYECTO FINAL 2025

# ROBOT DE EXTRACCIÓN Y COLOCACIÓN DE NEUMÁTICOS EN VEHÍCULOS DE COMPETICIÓN (F1)

Grupo N°10:

- **Bergamin** Mikael
- **García** Tomás
- **Gantus** Gonzalo
- **Squartini** Jerónimo

Profesores:

- **Díaz** Carolina
- **Sanchez** Eric



## Resumen

El presente proyecto tiene como objetivo el diseño, modelado y simulación de una celda robótica destinada al cambio automatizado de neumáticos en vehículos de competición de Fórmula 1. El sistema propuesto contempla el uso de ocho robots industriales KUKA KR 70 R2100, organizados en pares para trabajar de forma simultánea sobre cada rueda del vehículo. En este trabajo se desarrolla el modelado y planificación de trayectorias correspondientes a uno de estos pares de robots, encargados de ejecutar las maniobras de extracción e instalación del neumático.

Para el desarrollo del proyecto se aplicaron los fundamentos de la cinemática directa e inversa utilizando la parametrización Denavit–Hartenberg, el análisis del Jacobiano y la detección de singularidades, así como la generación de trayectorias mediante interpolación en los espacios articular y cartesiano. Las trayectorias fueron diseñadas con el software MATLAB, empleando funciones como jtraj y ctraj para garantizar movimientos continuos y libres de colisiones.

Como aporte principal, se logró la coordinación de dos manipuladores KUKA trabajando en secuencia sincronizada dentro de una celda de trabajo compartida, junto con el diseño conceptual de un efecto final híbrido que integra un gripper y una llave de impacto central.

(Palabras clave: Robótica industrial, planificación de trayectorias, KUKA KR 70 R2100, cinemática, Fórmula 1)

## Índice

<b>Resumen</b>	<b>1</b>
<b>Introducción</b>	<b>3</b>
<b>Especificaciones técnicas</b>	<b>3</b>
1. Robot Seleccionado: KUKA KR 70 R2100	3
2. Tarea y disposición de la celda	4
3. Especificaciones del Manipulador (KR 70 R2100)	4
4. Especificaciones del Objeto (Rueda de F1)	5
<b>Diseño del efecto final</b>	<b>6</b>
<b>Robot</b>	<b>7</b>
Denavit - Hartenberg	7
Cinemática Directa	9
Cinemática Inversa	10
Relación de velocidades	10
Planificación y generación de trayectorias	12
Simulaciones y gráficas	19
<b>Sensores y actuadores</b>	<b>20</b>
<b>Aporte</b>	<b>22</b>
<b>Conclusiones</b>	<b>22</b>
<b>Referencias</b>	<b>24</b>
<b>Anexos</b>	<b>25</b>
Código Matlab	25
Datasheet	26

## Introducción

La aplicación que se ha ideado es la del cambio de neumáticos en vehículos de competición, para el caso específico de la Fórmula 1.

En la aplicación elegida se hará uso de dos robots de 6 grados de libertad trabajando de manera simultánea y combinada para realizar el cambio de rueda. El efecto final, por otra parte, consiste en un elemento de agarre para manipular los neumáticos y una llave de impacto central que permitirá ajustar y desajustar la tuerca que une el neumático al auto.

Una vez el vehículo ingrese a realizar su parada, mediante un sistema de cámaras, se detectará la posición específica del neumático, permitiendo realizar una corrección fina de la posición. Conocida esta, el primer robot se colocará frente a la rueda y quitará el neumático gastado, para posteriormente darle paso al segundo robot, el cual se colocará frente al vehículo, colocará el neumático nuevo, ajustará la tuerca y se apartará del monoplaza para permitirle a éste la salida a pista sin riesgos de colisión con los robots.

Para esta implementación se ha decidido hacer uso de un robot KUKA KR 70 R2100, que debido a su alcance de 2100 mm y su capacidad de carga de 70 kg, resulta ser una opción con características más que adecuadas para la manipulación de ruedas de fórmula 1, las cuales cuentan con una masa de entre 18 a 21 kg. Esta implementación contará además, con un efecto final diseñado personalizado.

## Especificaciones técnicas

En esta sección se detallan los componentes de la celda de trabajo, las características del manipulador seleccionado y las especificaciones del objeto a manipular, definiendo así el entorno y las restricciones del problema.

### 1. Robot Seleccionado: KUKA KR 70 R2100

Para este proyecto se optó por el KUKA KR 70 R2100, un robot industrial de seis grados de libertad (tipo articulado) perteneciente también a la familia KR IONTEC. Este modelo ofrece una buena relación entre alcance, capacidad de carga y precisión, por lo que resulta muy adecuado para manipular las ruedas de Fórmula 1 (KUKA AG, s.f.).

La elección se basó principalmente en dos aspectos:

- **Capacidad de carga:** el robot puede manejar hasta 70 kg, lo cual deja un margen importante considerando que cada rueda completa (llanta + neumático) pesa entre 18 y 21 kg (Pirelli & C. S.p.A., 2022). A esto se suma el peso del efecto final, que incluye la llave de impacto y el sistema de agarre. Con esta capacidad el robot trabaja sin exigirse y mantiene una buena precisión dinámica incluso durante movimientos rápidos.



- **Alcance:** con un radio de trabajo de 2101 mm, el KR 70 R2100 puede llegar sin problema hasta la posición de la rueda y luego retirarse a una zona segura, lo que fue clave al definir la disposición de la celda (KUKA AG, s.f.).

El robot tiene una repetibilidad de  $\pm 0,06$  mm, estructura robusta y movimientos suaves. Además, puede instalarse en suelo, pared o techo; en nuestro caso se eligió montaje en piso para asegurar mayor rigidez y simplicidad en la instalación.

## 2. Tarea y disposición de la celda

En la aplicación propuesta se requiere alta precisión y velocidad, por lo que se decidió usar dos robots KR 70 R2100 que trabajen en conjunto, donde uno quita la rueda usada y otro coloca la rueda nueva.

En el diseño de la celda, los robots se ubican simétricamente respecto al eje central del auto:

- **Robot A:** colocado a +0,75 m sobre el eje Y, encargado de retirar la rueda gastada.
- **Robot B:** colocado a -0,75 m sobre el eje Y, responsable de colocar la nueva rueda.

Ambos están montados sobre el suelo y comparten una zona de trabajo central frente al vehículo. De esta forma se evita que interfieran entre sí y se logra reducir el tiempo total de la operación. El área útil de trabajo tiene aproximadamente 3 m  $\times$  2 m, con una altura efectiva cercana a 1,5 m, suficiente para los movimientos de ambos efectores.

Como el vehículo puede detenerse con pequeñas desviaciones respecto al punto nominal, se contempla el uso de cámaras de visión para realizar una corrección fina en las trayectorias. Este sistema detecta la posición real de la tuerca central y permite al controlador ajustar los movimientos antes del acople para asegurar una alineación precisa.

## 3. Especificaciones del Manipulador (KR 70 R2100)

El KR 70 R2100 está compuesto por seis articulaciones rotacionales, lo que le da una configuración cinemática 6R. Los ejes se denominan A1 a A6 y cada uno tiene un rango de movimiento que define el volumen total de trabajo del robot.(KUKA AG, s.f.).

Eje	Rango aproximado
A1	$\pm 185^\circ$
A2	$-135^\circ$ a $+65^\circ$
A3	$-120^\circ$ a $+158^\circ$



A4	$\pm 350^\circ$
A5	$\pm 122.5^\circ$
A6	$\pm 350^\circ$

Tabla 1. Límites articulares del robot KUKA KR 70 K2100.

El espacio de trabajo generado por estos límites, considerando el alcance de 2,1 m, cubre cómodamente toda la zona donde se ubican las ruedas del monoplaza y permite definir posiciones de espera seguras para ambos manipuladores. [Anexo 2]

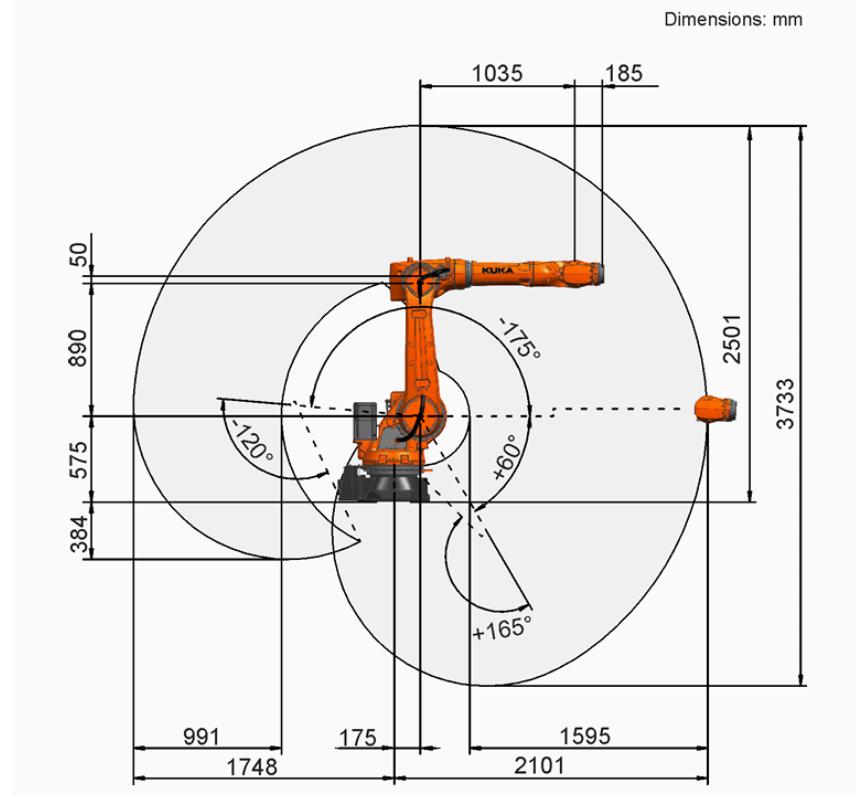


Gráfico 1. Espacio de trabajo

#### 4. Especificaciones del Objeto (Rueda de F1)

El objeto a manipular es la rueda completa del monoplaza, formada por la llanta y el neumático. A partir de la temporada 2022 se utilizan llantas de 18 pulgadas, lo que aumentó tanto el tamaño como la masa del conjunto respecto a versiones anteriores (Pirelli & C. S.p.A., 2022).

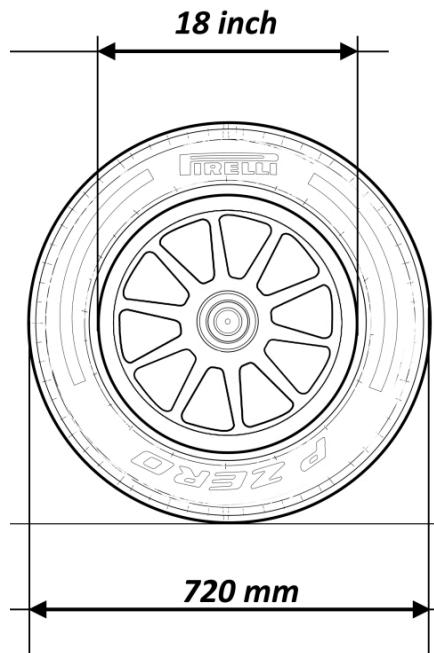


Gráfico 2. Tamaño del neumático Pirelli de Fórmula 1

Parámetro	Valor aproximado
Masa total	18 – 21 kg
Diámetro exterior	720 mm
Ancho (delantero / trasero)	305 mm / 405 mm

Tabla 2. Parámetros del neumático Pirelli utilizado en la Fórmula 1

La diferencia de ancho entre los neumáticos delanteros y traseros fue una de las principales restricciones al diseñar el efecto final, que debe ajustarse a ambas medidas manteniendo una sujeción firme y sin dañar la goma. Se realizó el diseño del gripper ajustable debido a que, aunque el análisis y simulación se haya realizado considerando solo el neumático delantero, el hecho de que el gripper funcione para ambos neumáticos permite simplificar el estudio del neumático trasero en un futuro, cambiando solo los valores de trayectoria de aproximación y el ángulo de ataque que adopta el robot en las fases de montaje y desmontaje.

## Diseño del efecto final

Para la manipulación del neumático se diseñó un efecto final personalizado, adaptado a la geometría de la rueda y las condiciones de trabajo del robot KUKA KR 70 R2100.

El diseño se realizó mediante la herramienta de Fusion considerando, compatibilidad con el neumático.

El mecanismo cuenta con cuatro brazos de sujeción montados sobre guías lineales, accionados de forma simultánea para la sujeción del neumático. Además, cuenta con un



acople central que funciona como llave de impacto que permitirá ajustar y desajustar la tuerca que une el neumático al auto (pieza en rojo).

En las siguientes figuras se observa el modelo 3D del gripper y su integración con el robot.



Gráficos 3, 4 y 5. Vista frontal y trasera del gripper con y sin sujeción del neumático.



Gráficos 6, 7 y 8. Diferentes vistas del robot KUKA con el gripper acoplado con y sin sujeción del neumático.

## Robot

### Denavit - Hartenberg

Para lograr modelar la cinemática del robot KUKA KR 70 R2100 aplicamos la convención Denavit-Hartenberg, que es un método sistemático que permite resolver el problema de la cinemática directa a través de matrices de transformaciones homogéneas, logrando describir la posición y orientación entre eslabones mediante cuatro parámetros geométricos

Aplicamos la convención clásica del toolbox utilizado, donde  $d_i$  representa el desplazamiento a lo largo del eje  $z_{i-1}$ , desde el origen del sistema  $i-1$  hasta el eje  $x_i$ ,  $a_i$  representa la distancia a lo largo del eje  $x_i$  desde el eje  $z_{i-1}$  hasta el eje  $z_i$ , y los ángulos  $\alpha$  y  $\theta$  representan el ángulo alrededor del eje  $x_i$  desde  $z_{i-1}$  hasta el eje  $z_i$  y el ángulo alrededor del eje  $z_{i-1}$  desde  $x_{i-1}$  hasta el eje  $x_i$ .



Esta representación nos permite definir con precisión la geometría del robot para los cálculos cinemáticos y una planificación de trayectoria.

<b>Sist</b>	<b><math>\theta</math></b>	<b><math>d [m]</math></b>	<b><math>a[m]</math></b>	<b><math>a[rad]</math></b>	<b><math>\sigma</math></b>
1	$\theta_1$	0.575	0.175	$-\pi/2$	0
2	$\theta_2$	0.000	0.890	0	0
3	$\theta_3$	0.000	0.050	$-\pi/2$	0
4	$\theta_4$	1.035	0.000	$\pi/2$	0
5	$\theta_5$	0.000	0.000	$-\pi/2$	0
6	$\theta_6$	0.185	0.000	0	0

**Offsets:** Respecto a los offsets, inicializamos ambos robots con un offset articular en la segunda articulación de  $-\pi/2$  rad, con el objetivo de alinear el eje del segundo eslabón con la posición de referencia física, permitiendo que el cero del sistema quede orientado hacia el monoplaza.

Para lograr la diferencia de orientación de ambos robots se modifican en la instancia de ploteo, aplicando una rotación inicial de  $+\pi/2$  en el caso del robot A y de  $-\pi/2$  en el caso del robot B, haciendo que queden orientados hacia el eje Y positivo y el eje Y negativo respectivamente cuando  $q_2 = 0$ .

De esta manera logramos mantener los mismos offsets en ambos casos y el mismo modelo geométrico, simplificando la planificación de trayectorias simultáneas.

**Matriz base:** En este proyecto utilizamos la matriz base para posicionar a los dos robots en diferentes lugares dentro del layout, de manera tal que ambos queden enfrentados, y a la misma distancia, del monoplaza.

Matriz base robot A

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.75 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz base robot B

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.75 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Matriz tool:** La matriz tool nos representa la transformación entre el último eslabón del robot y el punto de trabajo efectivo del efecto final, por lo que se definió, a partir del diseño 3D del efecto, una traslación de 0.245 m sobre el eje Z del sexto eslabón. Esto corresponde a la distancia entre el último eslabón del robot y el extremo activo de la herramienta.

Matriz tool de ambos robots

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Cinemática Directa

A partir de la parametrización que obtuvimos con el método DH, se realizó el cálculo de cinemática directa mediante el uso del toolbox de Peter Corke (Corke, 2017), haciendo uso de la función `fkine(q)`. Dicha función, recibe un vector “q” de posiciones articulares y usando la matriz `dh`, devuelve la matriz homogénea  ${}^{\text{base}}\mathbf{T}_{\text{tool}} = {}^{\text{base}}\mathbf{T}_0 \cdot {}^0\mathbf{T}_6 \cdot {}^6\mathbf{T}_{\text{tool}}$ . Esta matriz representa la posición del efecto final respecto al sistema base del robot, lo cual nos permite conocer la posición del gripper en todo momento dentro del entorno de trabajo.

En nuestro trabajo, utilizamos la cinemática directa principalmente para conocer la posición inicial del efecto final de cada uno de los dos robots que tenemos dentro de la celda de trabajo, así como para obtener la posición y orientación exactas del gripper en cada configuración articular en todo el proceso de trabajo.

Otro uso que le dimos a la cinemática directa fue el de obtener el espacio de trabajo, lo que nos permitió saber cuáles eran los límites de ambos robots, logrando saber en qué posiciones podríamos ubicarlos al momento de diagramar el layout.

Cabe mencionar que esta función nos permitió verificar la coherencia geométrica del modelo DH, y validar que las trayectorias generadas se desarrollen dentro del espacio de trabajo identificando así hasta donde podía alcanzar cada robot sin invadir la zona de trabajo del otro robot mientras este se encuentra trabajando en ella, ni invadir la zona del monoplaza, lo que finalmente nos llevó a ubicar las bases de forma simétrica y optimizar al máximo el espacio disponible, que en el contexto de una parada de pits es limitado y además se encuentra reglamentado.

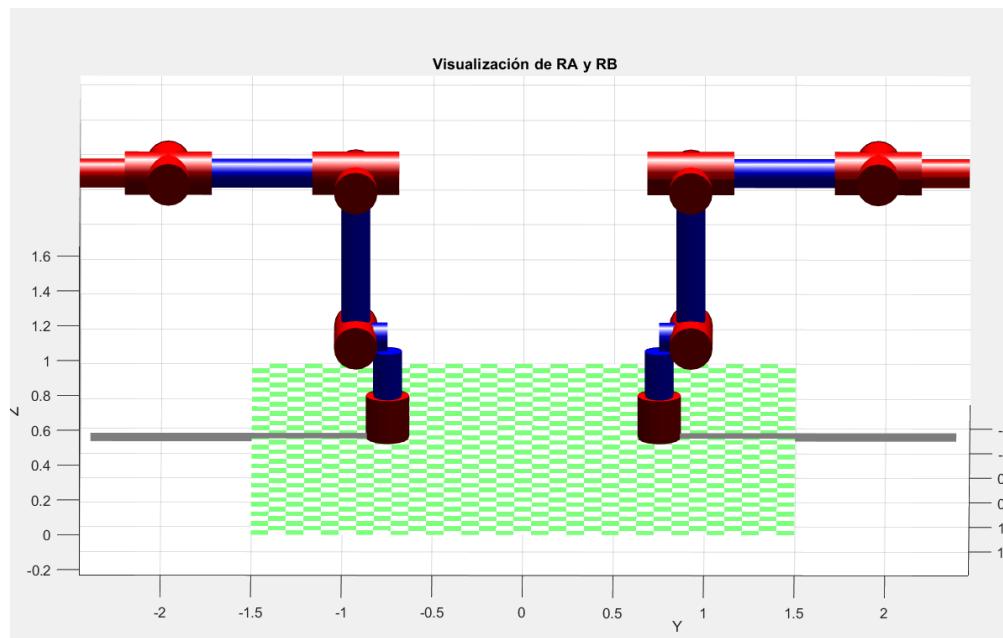


Gráfico 9. Layout de ambos robots en el espacio de trabajo

## Cinemática Inversa

Para el cálculo de la cinemática inversa en cada uno de los robots se decidió utilizar nuestra función `cinv.m`, que hemos desarrollado en un trabajo práctico previo. Esta función, que implementa un método geométrico, se aplica en todos los tipos de generación de trayectoria. En el caso de la obtención de trayectoria mediante interpolación en el espacio articular como es el caso de la función `g_trayectoria_a` y la función `g_trayectoria_ms`, la cinemática inversa se utiliza de forma limitada o una cantidad limitada de veces, ya solo es necesario calcular las configuraciones articulares de los puntos iniciales y finales de cada segmento, ya que las funciones `jtraj` y `mstraj` realizan la interpolación directamente en el espacio articular.

En cambio, para la interpolación en el espacio cartesiano que se implementa en `g_trayectoria_c`, el uso que se le da es mucho mayor, ya que la función `ctraj` genera una trayectoria densa de 100 poses intermedias entre el inicio y el fin, y nuestra función `cinv.m` es invocada en un bucle para resolver la cinemática inversa de cada una de esas 100 poses. La razón por la que se optó por esta implementación, en lugar de un método numérico iterativo externo, es la robustez que ofrece nuestro algoritmo. La función `cinv.m` está diseñada para calcular las 8 posibles soluciones geométricas y, fundamentalmente, seleccionar la "mejor" solución, definida como aquella con la menor diferencia respecto a la configuración articular anterior. Todos nuestros generadores de trayectoria utilizan esta característica, lo cual es crítico para `gTrayectoria_c`. Esto garantiza que, al calcular puntos muy próximos entre sí, el robot no sufra "saltos" en sus articulaciones, asegurando una trayectoria suave y coherente.

Un aspecto fundamental de nuestra implementación es que la función `cinv` fue diseñada para ser genérica, considerando las transformaciones de base (`R.base`), herramienta (`R.tool`) y los desfasajes (`R.offset`) de cada robot. Antes de iniciar el cálculo geométrico, la pose objetivo `T` es "desacoplada", es decir, se transforma para que represente la pose de la brida (el final del brazo) relativa al sistema de coordenadas de la base del robot. De igual manera, los offsets de las articulaciones son aplicados al final del cálculo. Esta generalización fue la que nos permitió utilizar una única función de cinemática inversa para `RA` y `RB`, a pesar de que ambos tienen transformaciones de base distintas para poder operar en la misma zona de trabajo.

## Relación de velocidades

A partir del modelo DH del robot que estamos trabajando, construimos el Jacobiano geométrico simbólico, que relaciona las velocidades articulares con velocidades lineales y angulares del efecto final.

El determinante que obtuvimos es el que representa los tres factores principales dependientes de las variables articulares  $q_2$ ,  $q_3$ , y  $q_5$ , significando la anulación de uno de ellos, un determinante nulo, y por lo tanto, una configuración singular.

$$\det J(q) = -\frac{89}{4 \cdot 10^6} s_5 \left( 7245 c_3 - 2070 c_2 - 100 s_2 + 350 s_3 + 36846 c_2 c_3 + \right. \\ \left. + 1780 c_2 s_3 + 4140 c_2 c_3^2 - 42749 c_3^2 s_2 - 42749 c_2 c_3 s_3 - 4140 c_3 s_2 s_3 \right)$$



donde  $c_i = \cos q_i$ ,  $s_i = \sin q_i$ .

Con el análisis realizado, obtuvimos 3 familias de singularidades:

- $q_5 = 0$ , donde los ejes 4 y 6 se encuentran alineados. En este caso el robot pierde un grado de libertad de orientación.

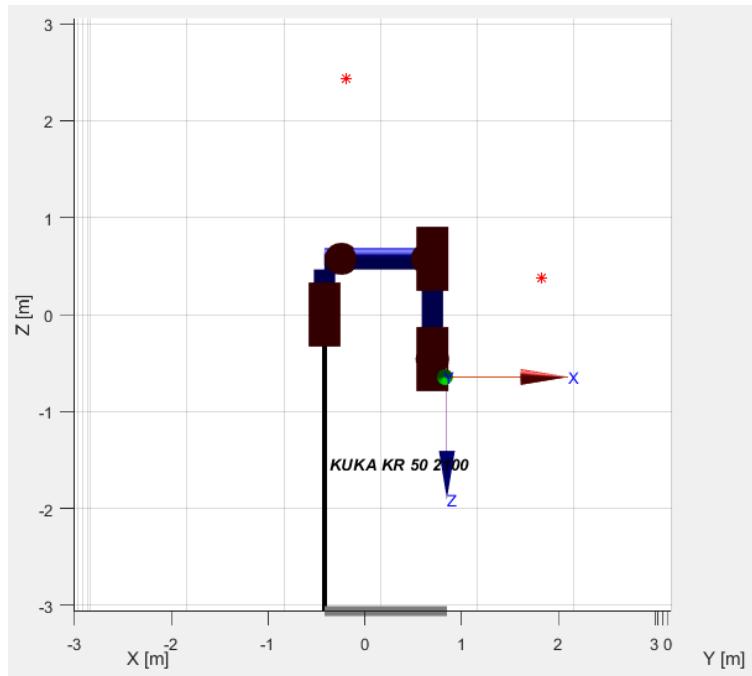


Gráfico 10. Primera singularidad (Singularidad de Muñeca)

- $q_3 = -1.5225$  y  $q_5 = \pi/2$ , posición que corresponde a la alineación del brazo y el antebrazo (singularidad de codo).

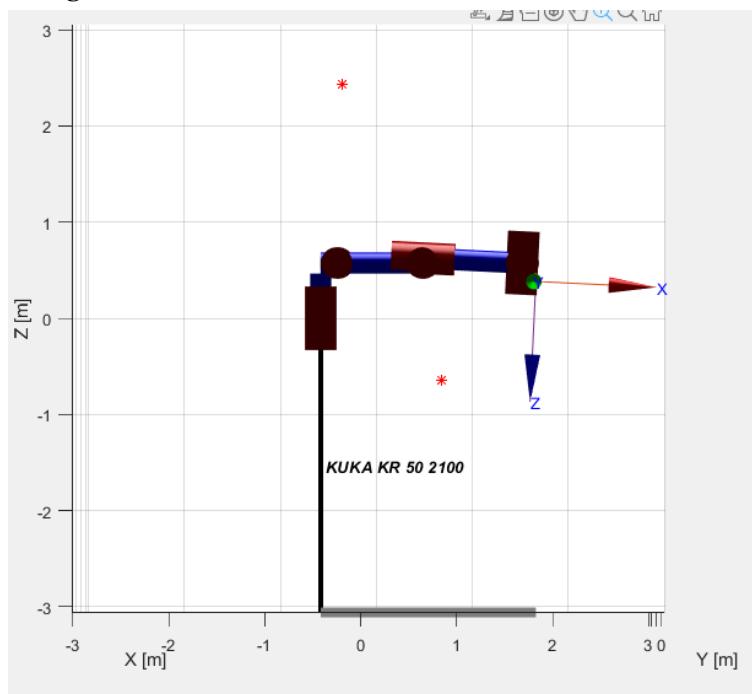




Gráfico 11. Segunda singularidad (Singularidad de Codo)

- $q_2 = -2.0944$ ,  $q_3 = -0.7353$ , y  $q_5 = \pi/2$ . Posición en donde la cadena cinemática se extiende en la misma línea de la base (singularidad de hombro).

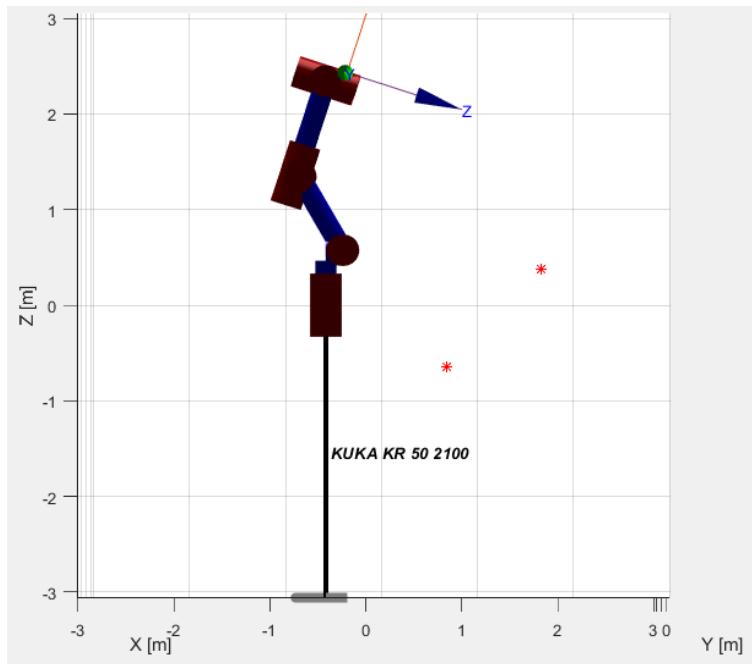


Gráfico 12. Tercera singularidad (Singularidad de Hombro)

Habiéndose comprobado estas 3 configuraciones mediante el cálculo del determinante del Jacobiano e igualando dicho determinante a cero.

Este análisis se realizó con el fin de realizar la planificación de trayectorias evitando atravesar por posiciones singulares y garantizando un comportamiento cinemático estable durante todo el proceso.

## Planificación y generación de trayectorias

En el layout nos encontramos con un robotA, que es el que está ubicado a 0.75m en positivo sobre el eje Y y el robotB, que es el que se encuentra a 0.75m negativo sobre el mismo eje.

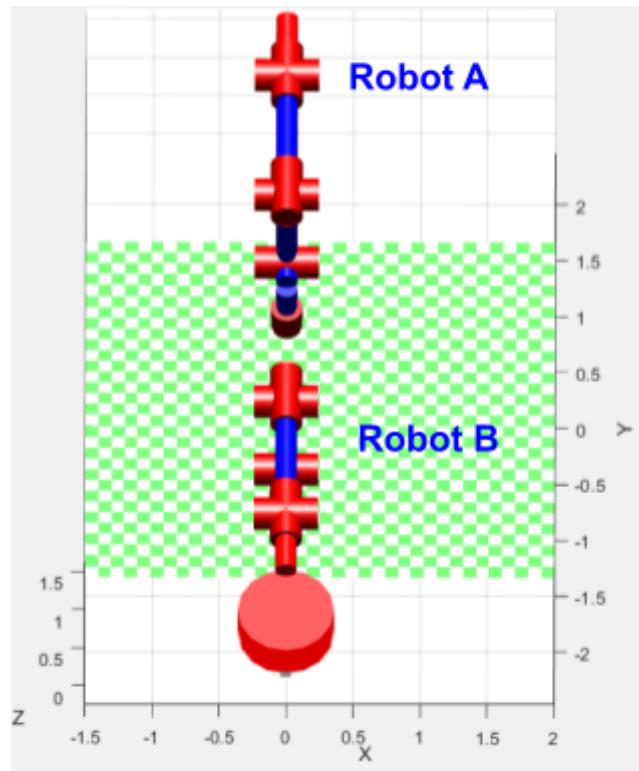


Gráfico 13. Layout de ambos robots en la posición de origen con el neumático nuevo

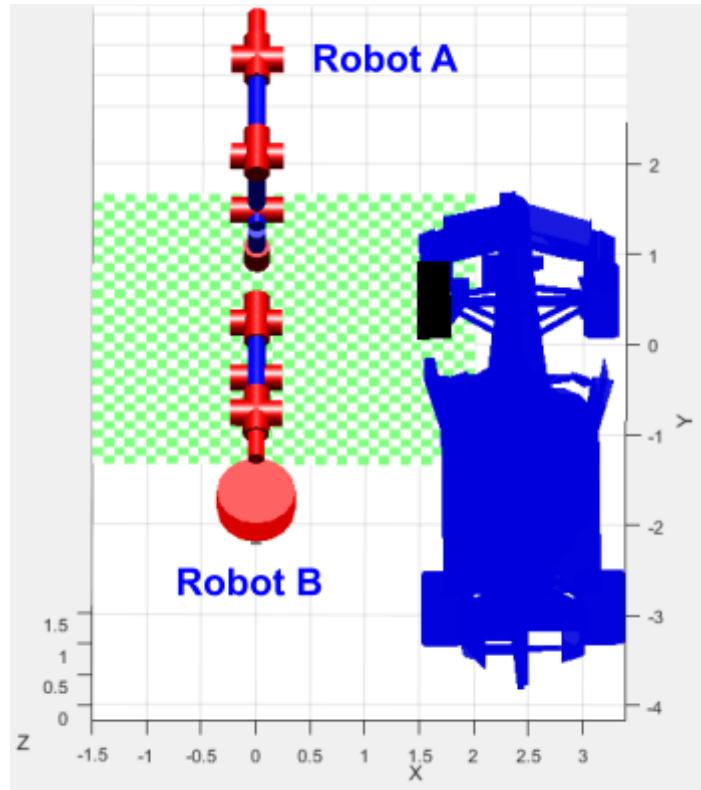


Gráfico 14. Layout de ambos robots en la posición de origen con el neumático nuevo y el monoplaza a un lado

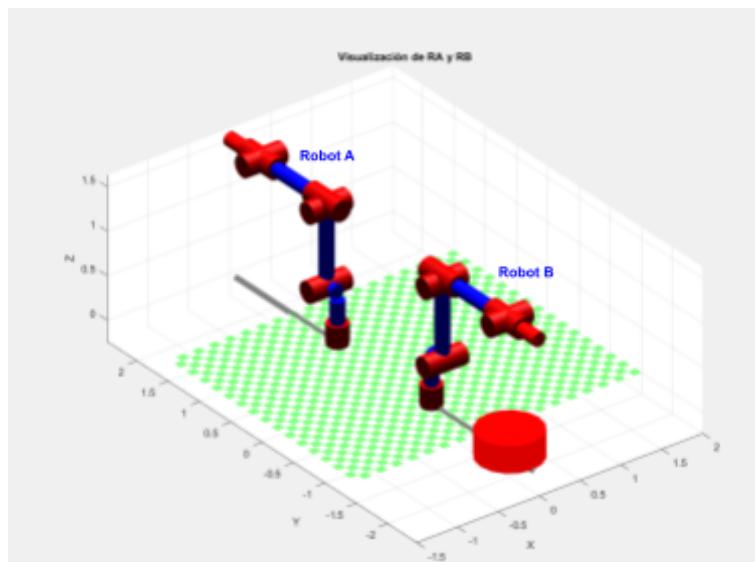


Gráfico 15. Layout de ambos robots en la posición de origen con el neumático nuevo.

El robotA se moverá desde la posición de reposo, hasta la posición de espera ubicada a +0.8m en Y respecto del eje X, y lo hará mediante interpolación en el espacio articular. Dicha posición se mantendrá hasta que el auto llegue. (Se ha supuesto que el vehículo entra desde un inicio a una altura de 10cm por encima del suelo, ya que esta altura corresponde al elevamiento que realizan los mecánicos en el momento del pit stop para luego realizar el cambio de neumáticos).

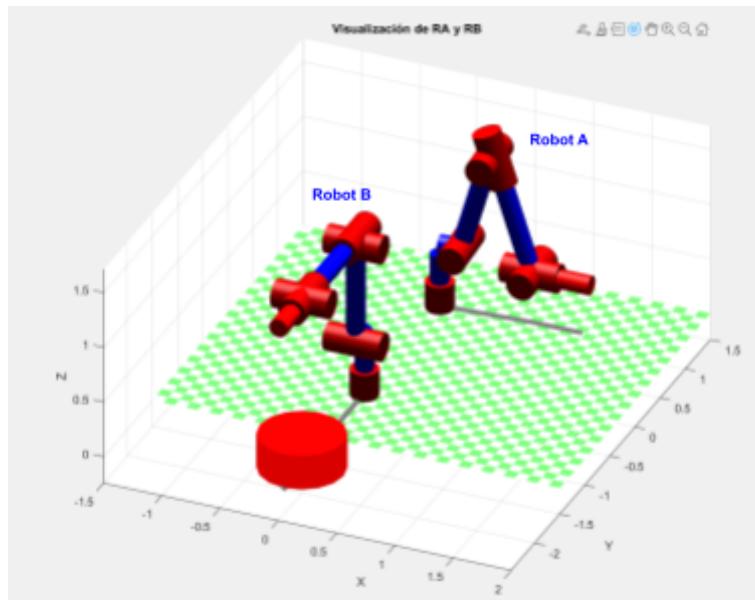


Gráfico 16. Paso 1 realizado, robot A en posición de espera.

Mientras esto ocurre, el robotB se posicionará por encima de una rueda, que se encuentra sobre el piso a -86 cm en el eje Y respecto a la base del robotB, donde este posicionamiento es logrado mediante una trayectoria generada por interpolación en el espacio articular. Posteriormente el robot se desplazará hacia abajo, sobre la dirección del eje z en sentido negativo, mediante una trayectoria generada por interpolación en el espacio cartesiano, y luego volverá a elevarse ahora con sentido positivo mediante una interpolación en el espacio articular. Finalmente la rueda se posicionará en la posición de espera a -0.8m



en Y respecto del eje X, realizando dicho posicionamiento mediante una trayectoria obtenida por una interpolación multisegmento usando la función mstraj().

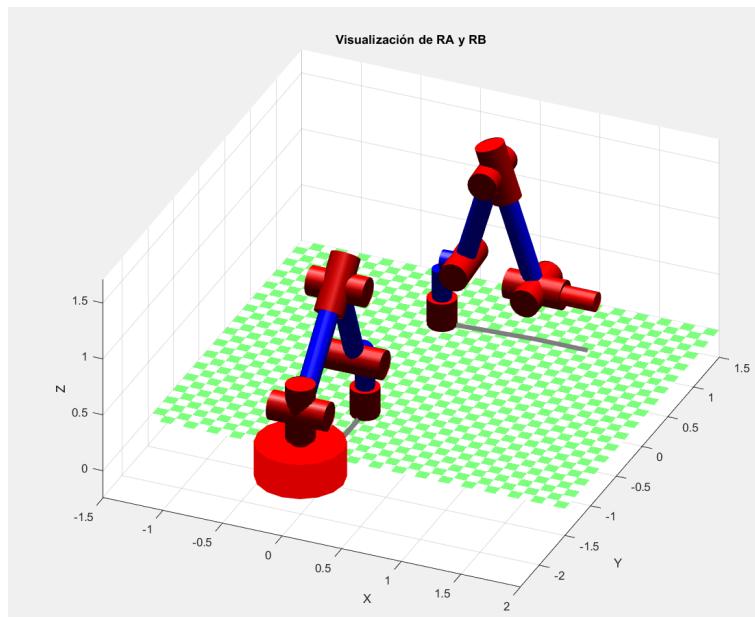


Gráfico 17. Paso 2 en proceso, Robot B tomando el neumático.

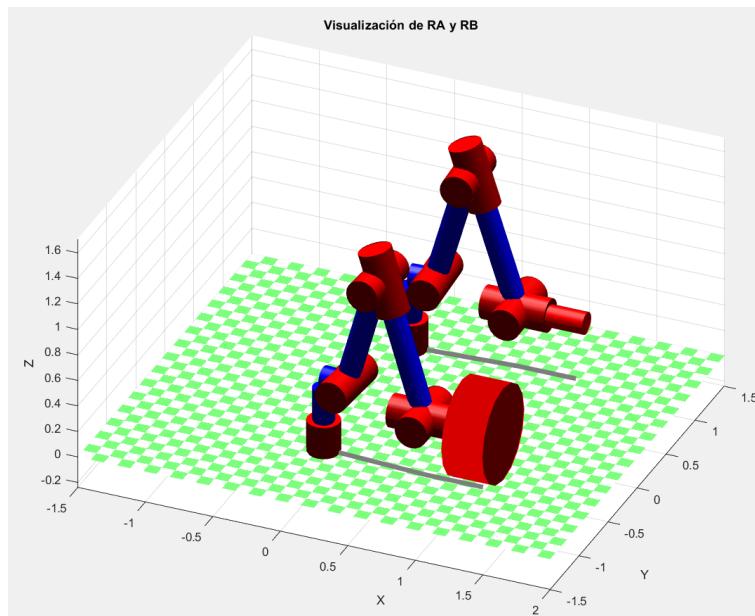


Gráfico 18. Paso 2 realizado, ambos robots en posición de espera.

Es necesario aclarar que tanto la posición de espera del robot A (+0.8m en Y) como la del robot B (+0.8m en Y) están referenciadas respecto de la posición ideal del neumático en la que teóricamente debería estar el mismo una vez ingresado el vehículo.

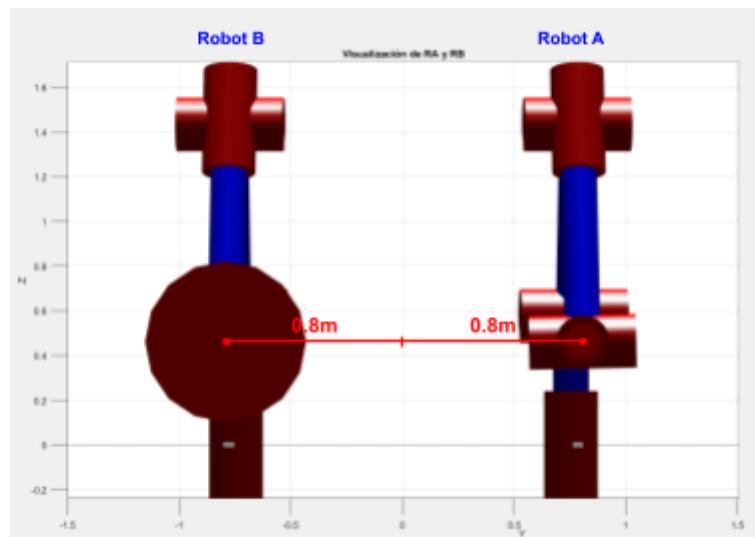


Gráfico 19. Distancia entre robots en la posición de espera.

Debido a que esto realmente no sucederá porque el vehículo no entra completamente recto a la zona de pits ni tampoco se posiciona sobre el o en el eje Y, esta posición deberá ser corregida para que el actuador se posicione de forma enfrentada al neumático (logrando dicha orientación gracias a la ayuda del sistema de cámara y QR detallado en la sección de [Sensores y actuadores](#)). Para realizar dicha corrección el robot A gira la muñeca mediante una interpolación en el espacio articular haciendo uso de la función jtraj, para luego acercarse hasta una posición enfrentada al neumático, haciendo uso de generación de trayectorias por interpolación en el espacio cartesiano.rá con una trayectoria obtenida mediante una interpolación en el espacio cartesiano. articular para girar la muñeca, se acerca con ctraj, entra sale. los dos hacen jtraj para acomodar la muñeca, se mueven los dos con ctraj, el robot a acomoda la muñeca

Una vez llegado el vehículo y con la posición del neumático ya corregida, el gripper del robot A, que se encuentra ya posicionado de frente al neumático a 60 cm del mismo, avanzará esta distancia para quitarlo y luego volverá, teniendo posesión del neumático, a la posición anterior. Todo esto es realizado mediante una trayectoria generada con interpolación en el espacio cartesiano.

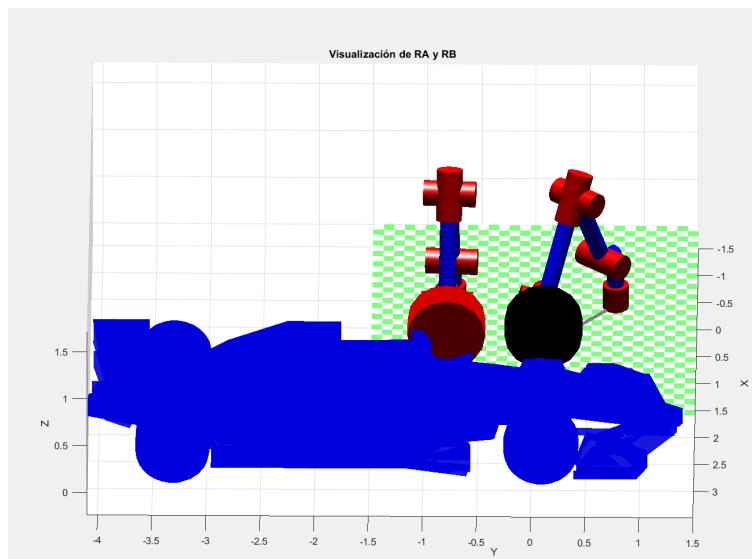




Gráfico 20. Robot A se mueve a la posición corregida y quita el neumático.

Una vez el robotA ha quitado el neumático, ambos robots acomodarán la muñeca mediante jtraj, para después moverse hacia el eje Y positivo, dando lugar en movimiento simultáneo al robotB que está ingresando el neumático, colocándose a 60cm de la posición corregida del mismo, realizando el movimiento con una trayectoria obtenida mediante una interpolación en el espacio cartesiano, realizando este tipo de generación de trayectoria con el objetivo de asegurar una distancia constante entre el efecto de la rueda y el vehículo. Luego, el robotB avanzará para colocarlo y finalmente se retirará dejando una cierta distancia de seguridad, permitiendo al monoplaza retornar a la pista. Dicho movimiento de entrada y salida se realizará mediante una trayectoria obtenida por interpolación en el espacio cartesiano.

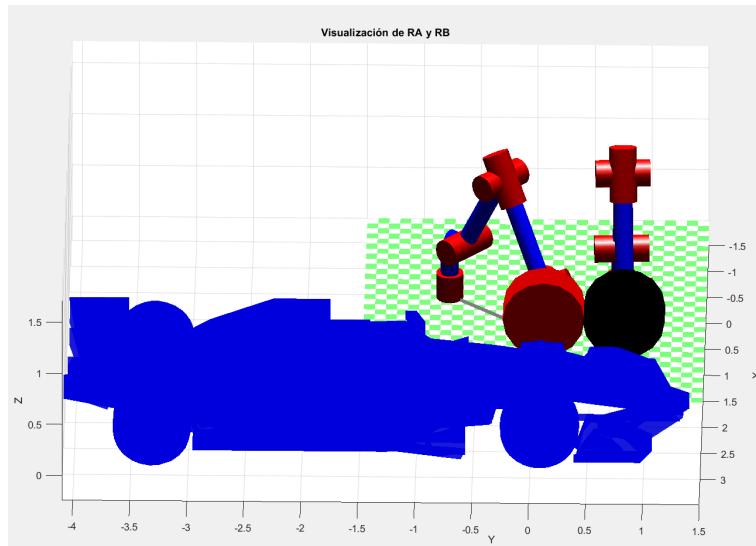


Gráfico 21. Robot A se desplaza y Robot B se coloca en la posición corregida

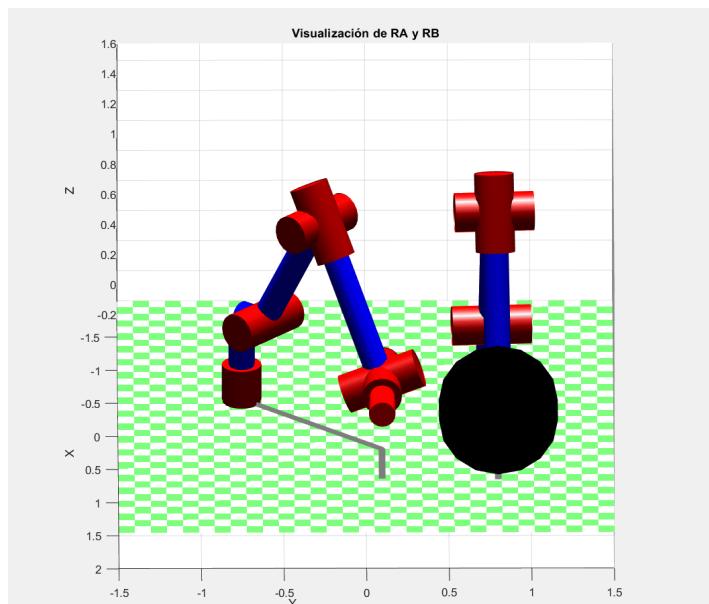


Gráfico 22. Robot B coloca el neumático y se retira el monoplaza

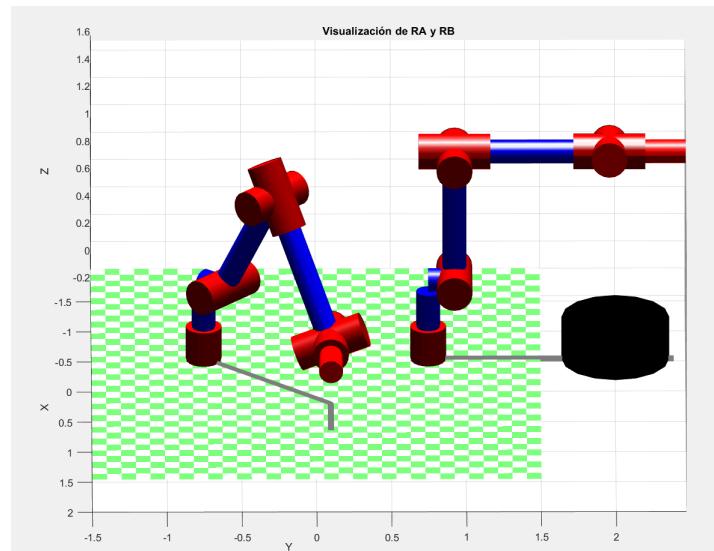


Gráfico 23. Robot A deja el neumático y vuelve a la posición de homing

Como último movimiento del robotA, éste se colocará en una posición superior a la que corresponde a la posición en donde se deposita el neumático mediante una trayectoria obtenida con el uso de la función mstraj. Posteriormente deposita la rueda descendiendo, colocando el neumático, y subiendo nuevamente con una trayectoria obtenida mediante interpolación en el espacio cartesiano, y por último, el robot volverá a la posición de homing mediante una trayectoria obtenida con interpolación en el espacio articular.

Por último, para finalizar el proceso, robotB realiza un homing haciendo uso de la función mstraj pasando por una posición intermedia para asegurar el correcto retorno al punto inicial.

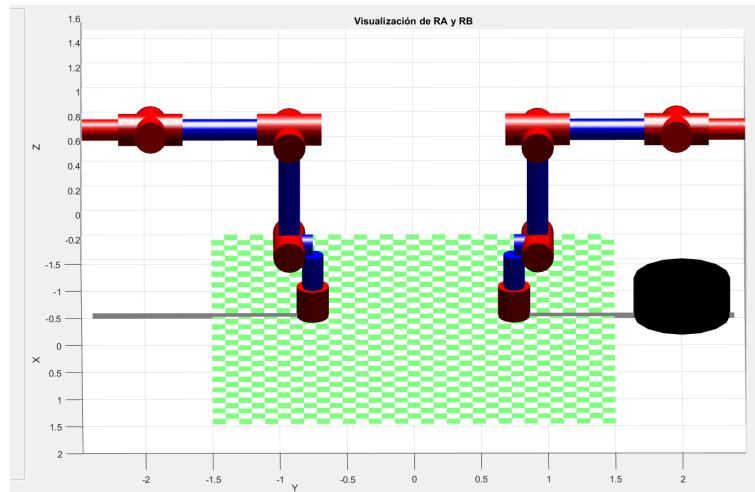


Gráfico 24. Robot B vuelve a la posición de homing

## Simulaciones y gráficas

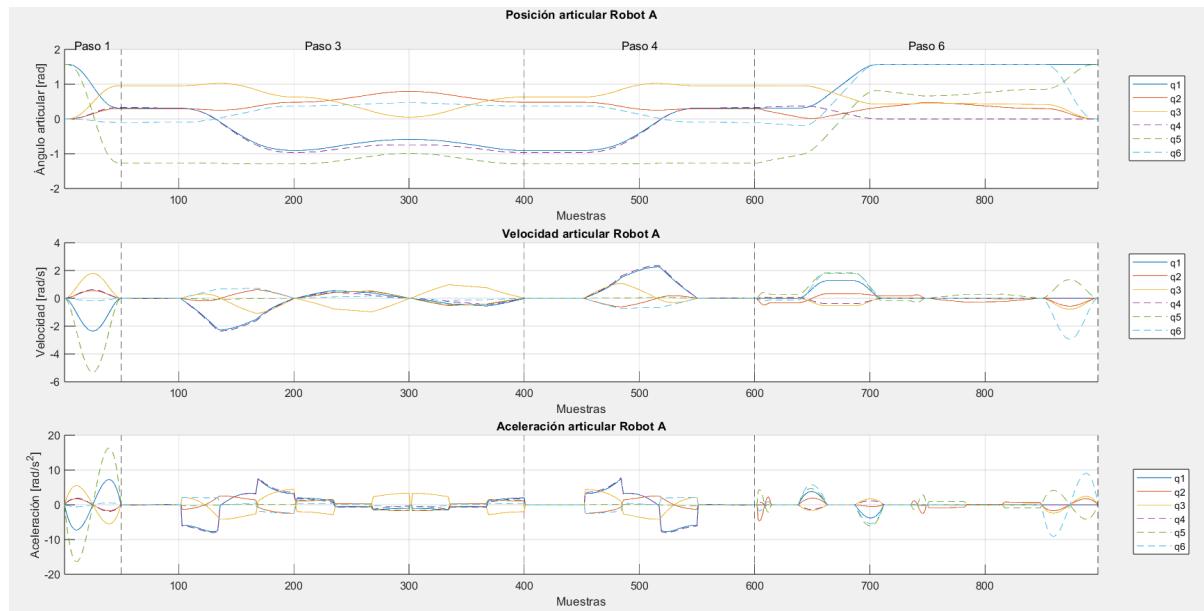


Gráfico 25. Posiciones, velocidades y aceleraciones articulares correspondientes a Robot A

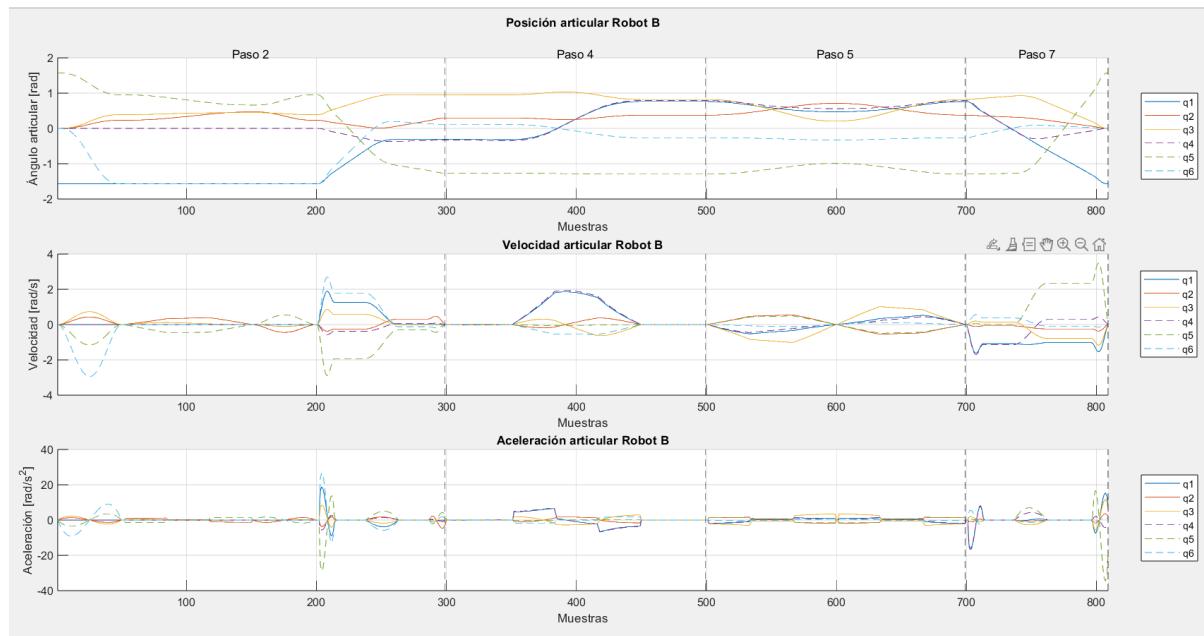


Gráfico 26. Posiciones, velocidades y aceleraciones articulares correspondientes a Robot B

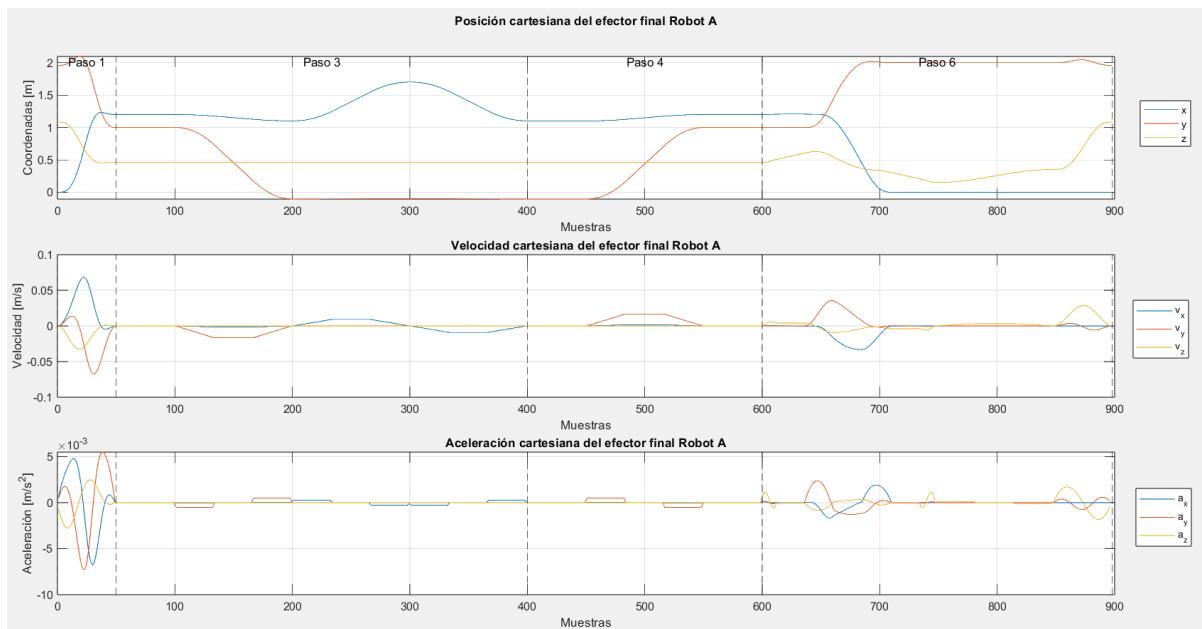


Gráfico 27. Posiciones, velocidades y aceleraciones cartesianas correspondientes a Robot A

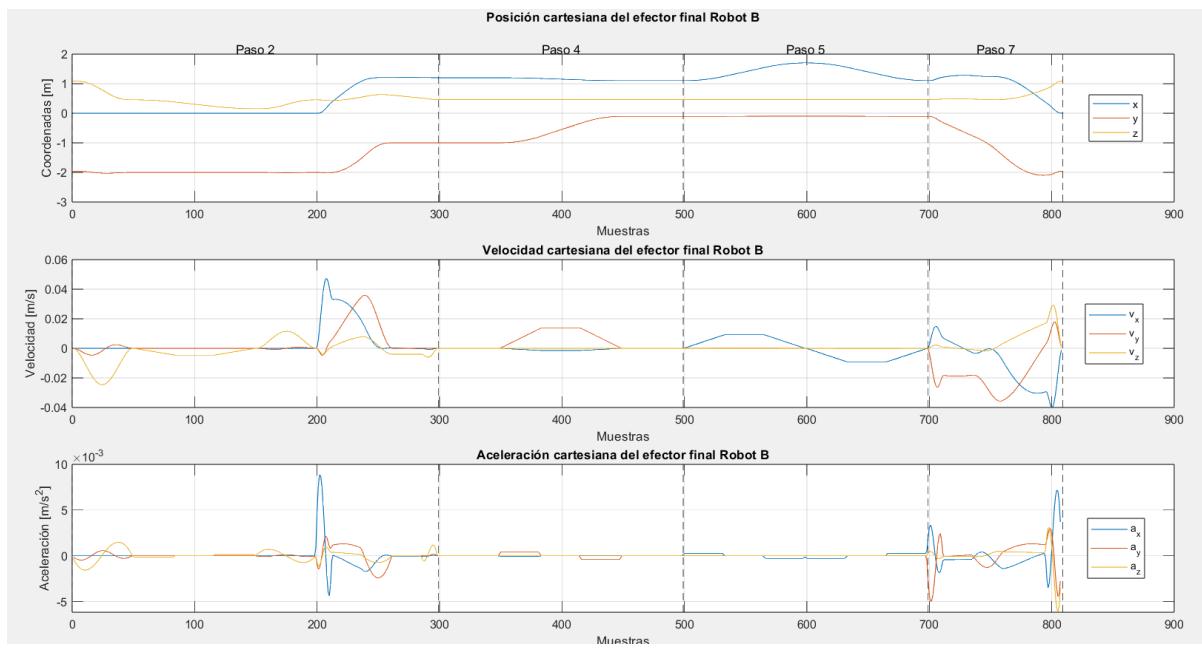


Gráfico 28. Posiciones, velocidades y aceleraciones cartesianas correspondientes a Robot B

## Sensores y actuadores

En el sistema propuesto, los sensores y actuadores cumplen un rol fundamental en para poder garantizar una gran precisión en las trayectorias repetitivas que se presentan en el proceso realizado.

En este caso contamos con una celda estructurada y controlada, con muy pocas variantes, en donde la posición de los robots y de los operarios se mantendrá fija, variando sólo levemente la posición del vehículo al cual se le realizará el cambio de neumático.



Debido a esto, la cantidad de sensores externos es baja, limitándose el sistema completo a un solo sensor de visión que permite realizar las correcciones de posición en función de la posición del neumático y a los sensores internos propios del robot.

### Sensores internos:

Cada robot KUKA posee un conjunto de sensores internos integrados los cuales permiten tener la información precisa del estado del robot en tiempo real. Entre este tipo de sensores pueden destacarse:

- **Encoders absolutos** rotativos en cada articulación. Estos encoders se encargan de medir el ángulo real de cada uno de los ejes con una resolución de milésimas de grado. Esta información la utiliza el controlador del robot para realizar el control de posición y velocidad en lazo cerrado, logrando trayectorias suaves y con una precisión que permite una repetibilidad de posición de 0.05mm.
- **Sensores de corriente y par.** Estos sensores se encuentran integrados a los servomotores, los cuales permiten estimar la carga aplicada en cada articulación. Con esta información se puede lograr una buena detección de sobreesfuerzos y una muy buena compensación dinámica del peso. Funciones muy útiles para mantener una buena seguridad y estabilidad mecánica durante el gran cambio de torque que se debe realizar debido al constante cambio entre trabajar sin peso extra a trabajar con 25 kg de carga.
- **Sensores de temperatura de los motores y reductores.** Estos tipos de sensores son muy importantes para la protección térmica del sistema de accionamiento, sobre todo en circuitos donde la temperatura no es despreciable como lo es el circuito de Singapur o Arabia Saudita. Esto toma más relevancia sabiendo que las temperaturas de funcionamiento del robot son relativamente bajas, llegando a un máximo de 55°C.



En conjunto, todos estos sensores internos mencionados permiten al robot conocer su posición articular completa, su velocidad y su estado dinámico. Información de suma importancia a la hora de realizar un seguimiento exacto de las trayectorias generadas mediante los algoritmos de interpolación `jtraj()` y `ctraj()`.

### Sensor de visión externo:

Un problema con el que nos encontramos al momento de plantear el caso es de cómo obtener la posición exacta de la rueda una vez ingresa el auto a los pits. Para poder determinar dicha posición antes de la manipulación por parte de los robots, se integra un sistema de visión artificial basado en una cámara industrial montada en la celda.

En alguna sección del monoplaza como los alerones delanteros, el chasis o sobre el mismo neumático se debe colocar un pintado o grabado un código QR, el cual, al ser escaneado por la cámara, logra actuar como una referencia visual.



Mediante el procesamiento de imagen, el sistema de visión artificial logra identificar el marcador y estimar con exactitud su posición y orientación.

### Actuadores del sistema:

Los actuadores principales del robot KUKA que utilizamos son los servomotores eléctricos integrados en cada uno de sus seis ejes.

Estos motores de corriente alterna son controlados por variadores digitales, y ofrecen una combinación óptima entre velocidad, par y precisión, las cuales son características esenciales para tareas de montaje rápido como el cambio de neumáticos en un monoplaza de fórmula 1.



Cada uno de estos servomotores se encuentra acoplado a un reductor, que multiplica el par de salida y de esta manera asegura un movimiento fluido en toda la cadena cinemática. Gracias a este sistema de accionamiento es que el robot logra alcanzar velocidades angulares desde 158 °/s hasta 320 °/s y mantener la gran precisión de repetitividad mencionada previamente.

Otros actuadores utilizados en el sistema son los actuadores que se emplean en el efecto final, siendo estos un actuador neumático que tiene como objetivo el cierre y apertura del gripper y un motor neumático de impacto (llave de impacto) para apriete y afloje de la tuerca de sujeción.

El actuador lineal utilizado por la sujeción será un cilindro neumático de doble efecto que logre generar el empuje lineal que cierra y abre las mordazas, logrando una alta potencia específica y una respuesta rápida y un mantenimiento simple. Respecto al tamaño, bastaría con un cilindro con un diámetro de émbolo de 40 mm y una carrera de aproximadamente 60 mm (variable para el caso de neumáticos de diferentes tamaños). Este tipo de actuador permite lograr una velocidad de apertura y cierre de aproximadamente 0.3 segundos, la cual puede ser regulable con reguladores de caudal.

El actuador rotativo utilizado para ajuste y desajuste mediante un motor neumático de impacto. Este motor entrega pulsos de par elevados que permiten aflojar y apretar la tuerca central. El par de afloje que tiene este tipo de herramientas es de aproximadamente 4300 Nm, pero puede variar en función de la herramienta específica utilizada, pudiendo ser mucho mayor en caso de uno desarrollado en el ámbito de la fórmula 1, y llegando a velocidades de 15000 rpm en vacío (Crisara, 2023).

## Aporte

Este trabajo incluye varios aportes originales debido a que hemos incorporado dos robots que trabajan sobre el mismo problema. Además, hemos creado una simulación completa en Matlab donde no sólo se animan los robots, sino también las ruedas y el auto de fórmula 1. Por último, se diseñó un efecto final que combina dos funciones: gripper para el neumático y llave neumática para el manejo de la tuerca.

## Conclusiones

El desarrollo de este proyecto permite aplicar de forma integrada los conceptos de cinemática directa e inversa, planificación de trayectoria, análisis del Jacobiano y singularidades en un caso de estudio de un sistema complejo y realista. A través de la

simulación de dos robots KUKA KR 70 R2100 se logró modelar con precisión el comportamiento de robots, validar su capacidad operativa dentro de un entorno definido y optimizar los movimientos realizados para lograr un objetivo específico, en este caso, el cambio de un neumático de auto de competición.

La implementación de trayectorias obtenidas mediante interpolación en el espacio articular con `jtraj()` y en el espacio cartesiano con `ctraj()`, junto con el análisis comparativo entre ambas, demostró la importancia de considerar parámetros como la aceleración, la velocidad, y los límites articulares para asegurar una suavidad en los movimientos y extender la vida útil del robot. Además, el diseño del gripper y la definición del entorno de trabajo contribuyeron a una representación completa del sistema.

En conjunto, el proyecto permitió comprobar la viabilidad de automatizar tareas de precisión utilizando robots industriales, y que la simulación resulta una herramienta de suma utilidad a la hora de comprobar y ajustar diseños antes de pasar a la implementación real.

Como trabajo futuro, podría ser interesante agregar el desarrollo de los sensores de visión artificial o algún sistema de retroalimentación que permita mejorar la precisión del posicionamiento, así como sincronizar de mejor manera los robots, para lograr una operación más coordinada y eficiente.

## Referencias

KUKA Roboter GmbH. (2025). KR IONTEC series — KR 70 R2100 data sheet. Augsburg, Germany: KUKA Robotics.  
<https://my.kuka.com/s/product/kr-70-r2100/o1t1ooooooooAQtSAAW?language=es&tab=Details>

Corke, P. (2017). Robotics, Vision and Control: Fundamental Algorithms in MATLAB (2nd ed.). Springer.  
<https://doi.org/10.1007/978-3-319-54413-7>

Corke, P. (2020). *Robotics Toolbox for MATLAB: User's Guide*. Robotics and Control Laboratory, Queensland University of Technology.  
<https://petercorke.com/toolboxes/robotics-toolbox/>

MathWorks. (2024). Robotics Toolbox User's Guide. The MathWorks Inc.  
<https://www.mathworks.com/help/robotics/>

Pirelli & C. S.p.A. (2022). Formula 1 Tyre Technical Data — Season 2022. Milan, Italy.  
<https://www.pirelli.com/tyres/es-mx/competition/f1/llantasf1>

Crisara, M. (2023, 30 de marzo). Inside the clever engineering and savage power of Formula 1 wheel guns. Popular Mechanics.  
<https://www.popularmechanics.com/cars/a43328928/formula-1-wheel-guns-explainer/>

## Anexos

### Código Matlab

Se han desarrollado múltiples módulos de código matlab para este proyecto. A continuación se explica brevemente qué hace cada módulo, y cómo ejecutar el programa. El código se encuentra adjunto como archivo .zip.

- **robotA.m:** Contiene la definición de la matriz dh, base, tool y offsets para el robot A.
- **robotB.m:** Contiene la definición de la matriz dh, base, tool y offsets para el robot A.
- **cinv.m:** Calcula la cinemática inversa mediante un método geométrico, capaz de resolver las 8 soluciones o seleccionar la mejor o más cercana a la configuración articular del punto anterior.
- **generarEscena.m:** Genera todos los puntos necesarios para definir las trayectorias (posiciones de las ruedas, auto, etc) y se encarga de generar una posición y orientación aleatoria del auto.
- **gTrayectoria\_a:** Contiene una función para generar una trayectoria mediante interpolación articular.
- **gTrayectoria\_c:** Contiene una función para generar una trayectoria mediante interpolación cartesiana.
- **gTrayectoria\_ms:** Contiene una función para generar una trayectoria articular multi-segmento
- **trayectorias.m:** A partir de los puntos ya obtenidos, se generan las trayectorias combinando ctraj, traj, y mstraj, y las guarda en *trayectorias\_robots.mat* para ser utilizadas luego. También guarda un archivo *escena.mat* con datos útiles para la animación.
- **Animador.m:** Este módulo define una clase con métodos que se encargan de animar el stl del fórmula 1, junto con las dos ruedas (rueda nueva y rueda gastada).
- **InterfazDosRobots.m:** Este módulo crea una ventana gráfica con algunos botones y la ventana de animación principal.
- **main.m:** Este programa ejecuta el programa “trayectorias.m” para generar las trayectorias y gráficas del sistema y luego corre “InterfazDosRobots.m”, el cual permite la visualización de la simulación.

#### **Instrucciones de uso:**

El sistema se puede utilizar de dos maneras diferentes:

- El archivo a ejecutar para visualizar la animación es **InterfazDosRobots.m**. Si se ejecuta *trayectorias.m*, se vuelven a generar las trayectorias (con una nueva posición aleatoria del auto) y se muestran las gráficas de posiciones, velocidades y aceleraciones articulares/cartesianas para los dos robots (demora unos segundos). Si luego de ejecutar *trayectorias.m*, se ejecuta *InterfazDosRobots.m*, en la animación se verá la nueva posición del auto.
- La otra manera de comprobar los resultados es ejecutar el archivo *main.m*, el cual ejecuta el programa de *trayectorias.m* para generar una nueva trayectoria y mostrar



las gráficas y posteriormente ejecuta el programa InterfazDosRobots.m, el cual muestra la animación.

### Datasheet

Se adjunta como PDF la datasheet del robot.