



TRABAJO FINAL ROBÓTICA I

SCARA BOLDO: PICK AND PLACE ARM



MANZANO, CLARIBEL ESTEFANÍA

Legajo: 11362

Prof. Carolina Díaz Baca – Prof. Eric Sánchez Ferreyra

ROBÓTICA I - FACULTAD DE INGENIERÍA - UNIVERSIDAD NACIONAL DE CUYO

2019

1 RESUMEN

En este informe se describirá un brazo de tipo SCARA, con una configuración cinemática RRPR. Su tarea se basa principalmente en Pick and Place, su aplicación será la del ordenado de piezas de Lego WeDo 2.0 del emprendimiento Cerebro Curioso, donde el ordenado de los mismos es indispensable para poder realizar las diferentes experiencias en los colegios sin que falte alguna parte crucial del proyecto propuesto y también para mantener el control de piezas en cada kit, ya que su costo es muy elevado. Anteriormente, esta tarea era realizada por los facilitadores del emprendimiento, pero en los últimos meses ha sido cada vez más difícil lograrlo por la poca disponibilidad de tiempo y especialmente por el costo que demanda por hora.

Se contará con visión artificial para el reconocimiento de las diferentes piezas que estarán sobre una plataforma especial, capaz de vibrar para separar estas sin que se superpongan en las imágenes. Además, se diseñará una bandeja especial para colocar cada tipo de pieza por separado en su correspondiente compartimiento

2 CONTENIDO

1 RESUMEN	1
3 INTRODUCCIÓN	3
4 PRESENTACIÓN TÉCNICA.....	4
4.1 ESPACIO DE TRABAJO	5
5 PARÁMETROS DENAVIT – HARTENBERG.....	7
6 CINEMÁTICA DIRECTA	9
7 CINEMÁTICA INVERSA	10
8 RELACIÓN DE VELOCIDADES	13
8.1 JACOBIANO	13
9 PLANIFICACIÓN DE TRAYECTORIA.....	15
10 VALIDACIONES.....	28
11 APORTE	32
12 ACTUADORES.....	35
13 SENSORES	38
14 CONCLUSIÓN	39

3 INTRODUCCIÓN

El brazo BOLD0 es un robot serie de tipo SCARA (*Selective Compliant Assembly Robot Arm*), con una configuración cinemática RRPR, que fue elegida después de la revisión de otras, como por ejemplo la configuración RPRR. A diferencia de esta, donde la segunda articulación debía levantar el peso del resto, en la elegida no se debe utilizar allí actuadores con gran torque, que generalmente son más pesados y agregan gran inercia al brazo. Se colocó entonces la articulación prismática en tercer lugar, logrando disminuir dificultades y costos de construcción.

Se adoptó un brazo tipo SCARA por la necesidad de los cuatro grados de libertad requeridos en la aplicación, posición en los ejes x_0 e y_0 para ubicar correctamente las piezas en su compartimiento, posición en z_0 , para poder tomar de la bandeja vibratoria la pieza requerida, y por último, una rotación final en el plano x_0y_0 , alrededor del eje z_0 , para orientar el efector final cuando se tome la pieza y también al colocarla correctamente en la bandeja (las piezas no entrarían en su compartimiento en cualquier orientación, ya que algunas son más largas que anchas y viceversa).

4 PRESENTACIÓN TÉCNICA

El Robot BOLDO, de tipo SCARA, tendrá como tarea principal el orden de piezas de Lego pertenecientes al kit WeDo 2.0. Por esto, podemos llamarlo Robot de Pick and Place, ya que tomará de una plataforma las diferentes piezas y las dejará en su correspondiente compartimiento en una bandeja especialmente diseñada.

En cuanto al diseño, fue realizado en conjunto con mi compañero, Nicolás Baca Coria, que también es parte de la empresa Cerebro Curioso.

Cuenta con cuatro articulaciones totalmente necesarias para su tarea, su configuración cinemática es RRPR, lo que le da un gran manejo y rapidez al efectuar su tarea.

La primera articulación posee los siguientes límites de trabajo, $\frac{3}{2}\pi > q_1 > -\frac{3}{2}\pi$, por esto, nuestro trabajo de trabajo se superpone y entonces, habrán puntos donde el brazo puede llegar de distintas maneras, es decir, tendremos dos ángulos diferentes para una misma posición.

En la segunda articulación tenemos, $\frac{2}{3}\pi > q_2 > -\frac{2}{3}\pi$. En la tercera, $50 > q_3 > 0$, donde el cero indica que el efector final está elevado y en 50, se encuentra a nivel de la base. Por último, para la última articulación, $\pi > q_4 > -\pi$, lo que le da la orientación al efector final, necesario para ubicar correctamente la pieza en su compartimiento.

La bandeja tendrá forma de arco, cuyos radios coinciden con el espacio de trabajo y su extensión va desde los 60° a los -60° desde el sistema en la base. También la plataforma vibratoria tendrá las mismas dimensiones y se ubicará desde 120° a 240° . Ambas quedarán a la altura de la base del robot, ya que esta se acoplará a otra base alta y más robusta.

Como efector final, se utilizará una ventosa neumática plana de vacío con adaptador de 8mm. Este efector fue elegido ya que las piezas de lego tienen gran variedad de tamaño y formas y esta ventosa es una herramienta capaz de adaptarse a estas prestaciones.

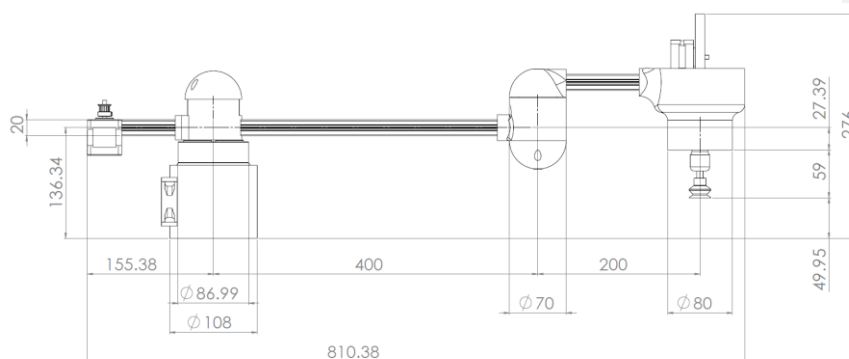


Figura 4.a: Plano con dimensiones finales

4.1 ESPACIO DE TRABAJO

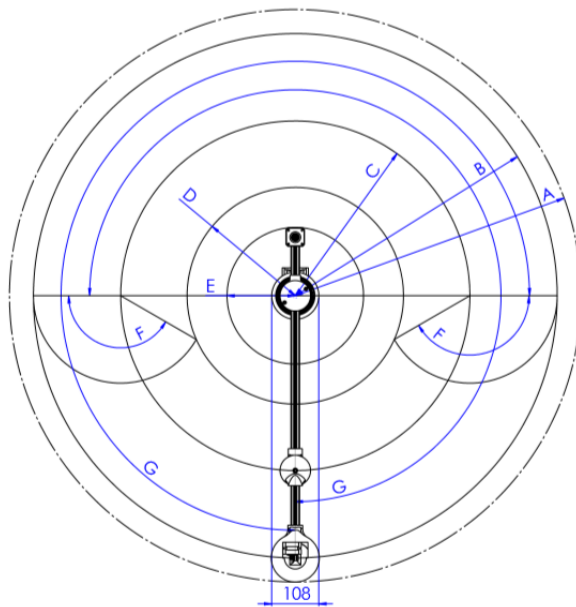


Figura 4.b: Plano con dimensiones del espacio de trabajo

- A Radio de trabajo máximo del brazo robótico ($r_A = 655 \text{ mm}$)
- B Radio de trabajo máximo efectivo ($r_B = 600 \text{ mm}$)
- C Arco generado por eje z de la segunda articulación ($r_C = 400 \text{ mm}$)
- D Radio de trabajo mínimo efectivo ($r_D = 247,87 \text{ mm}$)
- E Radio de seguridad para electrónica ($r_E = 156,32 \text{ mm}$)
- F Ángulo máximo de la segunda articulación (120°)
- G Ángulo máximo de la primera articulación (270°)

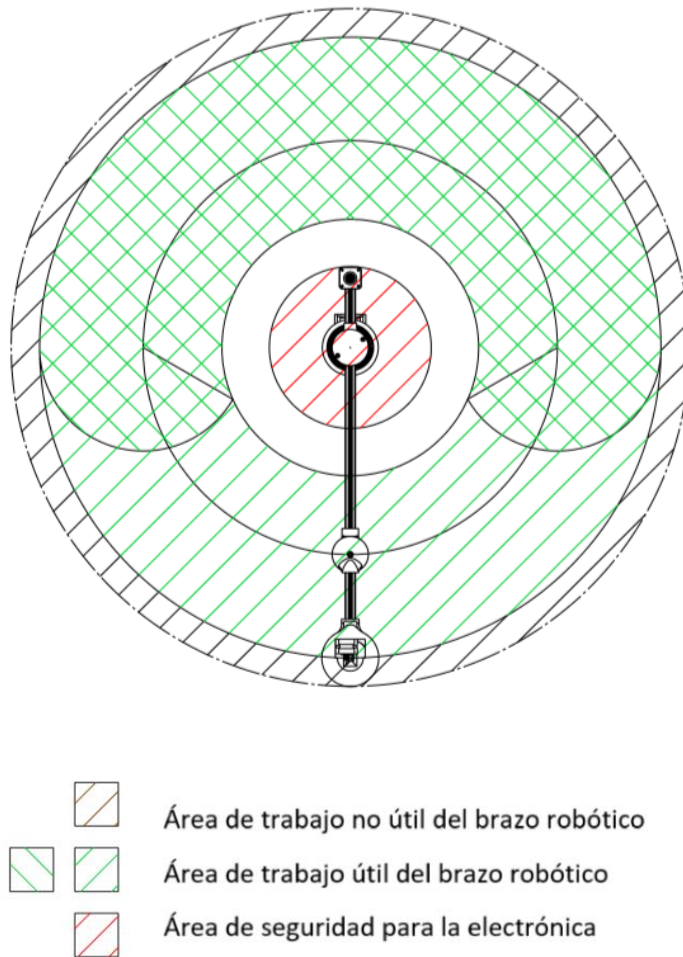


Figura 4.c: Espacio de trabajo

5 PARÁMETROS DENAVIT – HARTENBERG

Para obtener los parámetros de D-H, debemos definir algunas dimensiones del Robot:

$d_1 = 136,34 \text{ mm}$; $d_3 = 27,39 \text{ mm}$; $d_4 = 59 \text{ mm}$; $a_1 = 40 \text{ mm}$; $a_2 = 20 \text{ mm}$; y $\alpha_2 = \pi$.

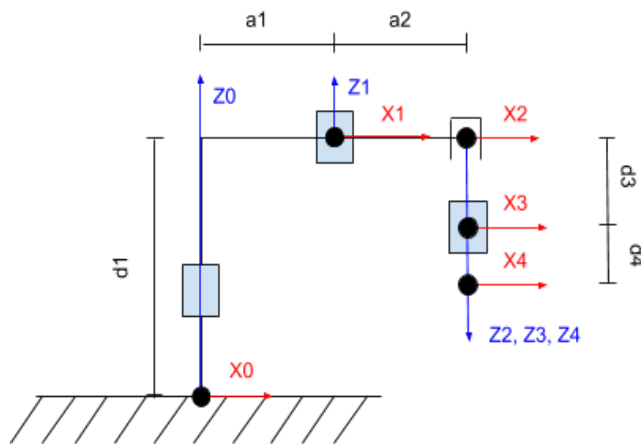


Figura 5.a: Sistemas de Denavit – Hartenberg

	θ	d	a	α	Tipo	Offset
Art 1	q_1	d_1	a_2	0	0	0
Art 2	q_2	0	a_2	α_2	0	0
Art 3	0	q_3	0	0	1	d_3
Art 4	q_4	d_4	0	0	0	0

Figura 5.b: Tabla de parámetros de Denavit - Hartenberg

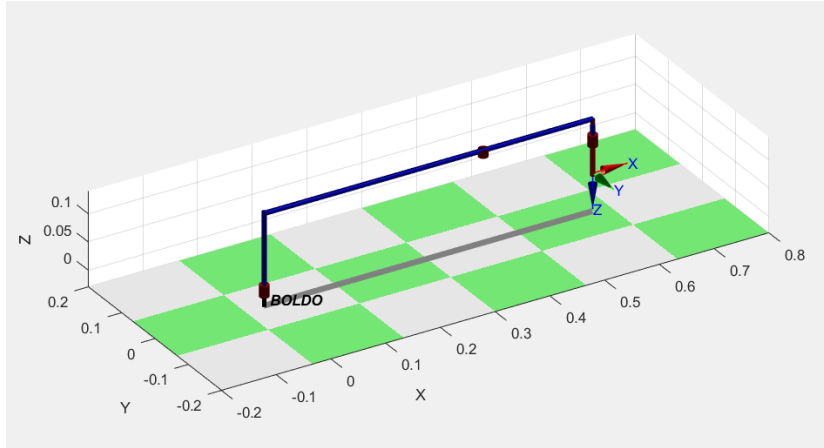


Figura 5.c: Simulación con parámetros D-H en Matlab en posición $q = [0, 0, 0, 0]$

6 CINEMÁTICA DIRECTA

Con los parámetros de Denavit – Hartenberg calculamos la cinemática directa de nuestro robot.

$${}^{i-1}T_i = Rot(z_{i-1}, \theta_i) \cdot Tras(z_{i-1}, d_i) \cdot Rot(x_i, \alpha_i) \cdot Tras(x_i, a_i) \quad (Ec. 6.1)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.2)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.3)$$

$${}^0T_4 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \quad (Ec. 6.4)$$

$${}^0T_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & a_1 \cdot \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & a_1 \cdot \sin q_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.5)$$

$${}^1T_2 = \begin{bmatrix} \cos q_2 & \sin q_2 & 0 & a_2 \cdot \cos q_2 \\ \sin q_2 & -\cos q_2 & 0 & a_2 \cdot \sin q_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.6)$$

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.7)$$

$${}^3T_4 = \begin{bmatrix} \cos q_4 & -\sin q_4 & 0 & 0 \\ \sin q_4 & \cos q_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.8)$$

$${}^0T_4 = \begin{bmatrix} \cos(q_1 + q_2 - q_4) & \sin(q_1 + q_2 - q_4) & 0 & a_2 \cdot \cos(q_1 + q_2) + a_1 \cdot \cos q_1 \\ \sin(q_1 + q_2 - q_4) & -\cos(q_1 + q_2 - q_4) & 0 & a_2 \cdot \sin(q_1 + q_2) + a_1 \cdot \sin q_1 \\ 0 & 0 & -1 & d_1 - q_3 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.9)$$

Validamos nuestros resultados en la sección VALIDACIONES.

7 CINEMÁTICA INVERSA

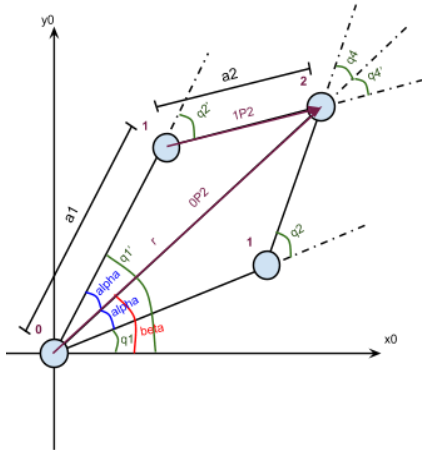


Figura 7.a: Vista simplificada superior del brazo robótico

Podemos obtener las variables articulares en función de la matriz de transformación homogénea 0T_4 , que puede ser nuestro dato de entrada. A continuación, utilizamos el Método de Pieper, dividiendo nuestro análisis.

$$\bar{q} = f({}^0T_4) \quad (Ec. 7.1)$$

$$\bar{q} = (q_1, q_2, q_3, q_4) \quad (Ec. 7.2)$$

Obtenemos q_1 a partir de la figura 7a, con el Método Geométrico:

$$q_1 = \beta \pm \alpha \quad (Ec. 7.3)$$

$$\alpha = \arccos\left(\frac{r^2 + a_2^2 - a_3^2}{2 \cdot a_2 \cdot r}\right) \quad (Ec. 7.4)$$

$$\beta = \arctan\left(\frac{y_{EF}}{x_{EF}}\right) \quad (Ec. 7.5)$$

$$r = \sqrt{x_{EF}^2 + y_{EF}^2} \quad (Ec. 7.6)$$

Para calcular q_2 , utilizamos el Método Matricial, con los vectores indicados en las figuras 7a y 7b:

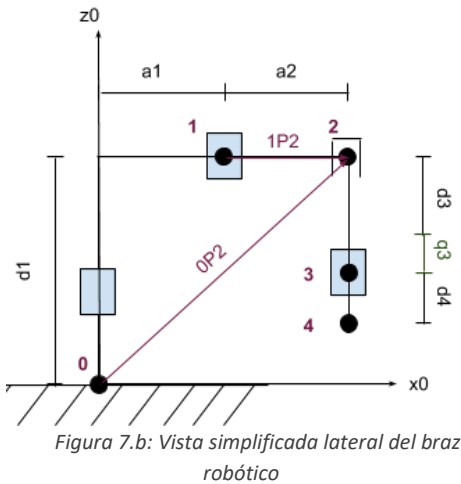


Figura 7.b: Vista simplificada lateral del brazo robótico

$${}^0P_2 = {}^0T_1 \cdot {}^1P_2 \Rightarrow {}^1P_2 = {}^1T_0 \cdot {}^0P_2 \quad (Ec. 7.7)$$

$${}^0T_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & a_1 \cdot \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & a_1 \cdot \sin q_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.5)$$

$${}^1T_0 = \begin{bmatrix} \cos q_1 & \sin q_1 & 0 & -a_1 \\ -\sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Ec. 6.6)$$

$${}^0P_2 = \begin{bmatrix} x_{EF} \\ y_{EF} \\ d_1 \\ 1 \end{bmatrix} \quad (Ec. 7.8)$$

$${}^1P_2 = \begin{bmatrix} x_{EF} \cdot \cos q_1 + y_{EF} \cdot \sin q_1 - a_1 \\ -x_{EF} \cdot \sin q_1 + y_{EF} \cdot \cos q_1 \\ 0 \\ 1 \end{bmatrix} \quad (Ec. 7.9)$$

$$q_2 = \arctan\left(\frac{{}^1P_2(2)}{{}^1P_2(1)}\right) \quad (Ec. 7.10)$$

$$q_2 = \arctan\left(\frac{-x_{EF} \cdot \text{sen } q_1 + y_{EF} \cdot \cos q_1}{x_{EF} \cdot \cos q_1 + y_{EF} \cdot \text{sen } q_1 - a_1}\right) \quad (\text{Ec. 7.11})$$

Deducimos q_3 , de la figura 7b:

$$d_1 = q_3 + d_3 + d_4 + z_{EF} \quad (\text{Ec. 7.12})$$

$$q_3 = d_1 - d_3 - d_4 - z_{EF} \quad (\text{Ec. 7.13})$$

Y finalmente, q_4 :

$${}^0R_4 = {}^0R_3 \cdot {}^3R_4 \Rightarrow {}^3R_4 = {}^0R_3 \cdot {}^0R_4 \quad (\text{Ec. 7.14})$$

$${}^0R_3 = {}^0R_1 \cdot {}^1R_2 \cdot {}^2R_3 \quad (\text{Ec. 7.15})$$

$${}^0R_3 = \begin{bmatrix} \cos(q_1 + q_2) & \text{sen}(q_1 + q_2) & 0 \\ \text{sen}(q_1 + q_2) & -\cos(q_1 + q_2) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{Ec. 7.16})$$

$${}^0R_4 = \begin{bmatrix} x_{x_{EF}} & y_{x_{EF}} & z_{x_{EF}} \\ x_{y_{EF}} & y_{y_{EF}} & z_{y_{EF}} \\ x_{z_{EF}} & y_{z_{EF}} & z_{z_{EF}} \end{bmatrix} \quad (\text{Ec. 7.17})$$

$${}^3R_4 = \begin{bmatrix} \cos q_4 & -\text{sen } q_4 & 0 \\ \text{sen } q_4 & \cos q_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{Ec. 7.18})$$

Multiplicando: ${}^3R_4 = {}^3R_0 \cdot {}^0R_4 \quad (\text{Ec. 7.19})$

$$\begin{bmatrix} \cos q_4 & -\text{sen } q_4 & 0 \\ \text{sen } q_4 & \cos q_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} x_{x_{EF}} \cdot \cos(q_1 + q_2) + x_{y_{EF}} \cdot \text{sen}(q_1 + q_2) & y_{x_{EF}} \cdot \cos(q_1 + q_2) + y_{y_{EF}} \cdot \text{sen}(q_1 + q_2) & z_{x_{EF}} \cdot \cos(q_1 + q_2) + z_{y_{EF}} \cdot \text{sen}(q_1 + q_2) \\ x_{x_{EF}} \cdot \text{sen}(q_1 + q_2) - x_{y_{EF}} \cdot \cos(q_1 + q_2) & y_{x_{EF}} \cdot \text{sen}(q_1 + q_2) - y_{y_{EF}} \cdot \cos(q_1 + q_2) & z_{x_{EF}} \cdot \text{sen}(q_1 + q_2) - z_{y_{EF}} \cdot \cos(q_1 + q_2) \\ -x_{z_{EF}} & -y_{z_{EF}} & -z_{z_{EF}} \end{bmatrix} \quad (\text{Ec. 7.20})$$

$$q_4 = \arctan\left(\frac{x_{x_{EF}} \cdot \text{sen}(q_1 + q_2) - x_{y_{EF}} \cdot \cos(q_1 + q_2)}{x_{x_{EF}} \cdot \cos(q_1 + q_2) + x_{y_{EF}} \cdot \text{sen}(q_1 + q_2)}\right) \quad (\text{Ec. 7.21})$$

Si definimos un ángulo: $\emptyset = q_1 + q_2 - q_4$, podemos obtener las soluciones en función de cuatro parámetros: $\bar{q} = f(x, y, z, \emptyset) = f(x_{EF}, y_{EF}, z_{EF}, \emptyset_{EF})$. Siendo \emptyset_{EF} un dato de entrada. Este se muestra en la Figura 7c, vemos que q_4 es negativo, ya que el eje z_4 tiene sentido contrario al eje z de la base.

$$q_4 = -(\emptyset_{EF} - q_2 - q_1) \quad (\text{Ec. 7.22})$$

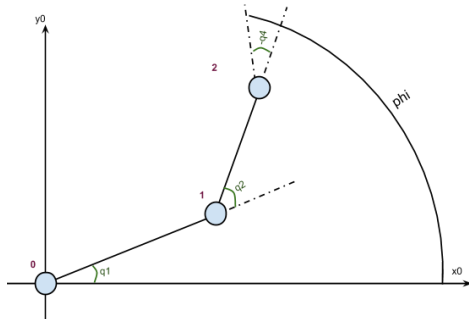


Figura 7.c: Coordenadas articulares del brazo robótico

Observamos que la cantidad de soluciones posibles es mayor a una. Hay zonas donde habrán sólo dos, especialmente en el primer y cuarto cuadrante, en algunos puntos cercanos a $\pm \pi/2$, tendremos tres y por último, en el segundo y tercer cuadrante habrán cuatro.

Para seleccionar las coordenadas más cercanas a la anteriores, se implementó un método de optimización, donde se decide al comparar los arcos recorridos por el brazo debido a la primera y segunda articulación, con sus respectivos pesos de

influencia. Se toma el menor arco de los mayores recorridos por cualquiera de las articulaciones, porque, por ejemplo, si hacemos la suma de ambos para cada solución y seleccionamos la menor, puede que, en esa suma, el mayor arco sea más grande que el mayor en otra solución y como el mayor arco en cada una de estas soluciones nos estará dando el tiempo de trayectoria o la velocidad con la que se realizará, es importante elegir el menor de los mayores.

Validamos nuestros resultados en la sección VALIDACIONES.

8 RELACIÓN DE VELOCIDADES

8.1 JACOBIANO

El Jacobiano, es una matriz formada por las derivadas parciales de primer orden de las funciones de cinemática directa, y es función de las coordenadas articulares de nuestro robot.

$$\dot{p} = J(q) \cdot \dot{q} \quad (Ec. 8.1) \quad \dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\emptyset} \end{bmatrix} \quad (Ec. 8.2) \quad \dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (Ec. 8.3)$$

Entonces, con nuestras ecuaciones de cinemática directa, obtenemos el Jacobiano:

$$x = a_1 \cdot \cos q_1 + a_2 \cdot \cos(q_1 + q_2) \quad (Ec. 8.4) \quad z = d_1 - q_3 - d_3 - d_4 \quad (Ec. 8.6)$$

$$y = a_1 \cdot \sin q_1 + a_2 \cdot \sin(q_1 + q_2) \quad (Ec. 8.5) \quad \emptyset = q_1 + q_2 - q_4 \quad (Ec. 8.7)$$

Derivando:

$$\dot{x} = -a_1 \cdot \sin(q_1) \cdot \dot{q}_1 - a_2 \cdot \sin(q_1 + q_2) \cdot \dot{q}_1 - a_2 \cdot \sin(q_1 + q_2) \cdot \dot{q}_2 \quad (Ec. 8.8)$$

$$\dot{y} = a_1 \cdot \cos(q_1) \cdot \dot{q}_1 + a_2 \cdot \cos(q_1 + q_2) \cdot \dot{q}_1 + a_2 \cdot \cos(q_1 + q_2) \cdot \dot{q}_2 \quad (Ec. 8.9)$$

$$\dot{z} = -\dot{q}_3 \quad (Ec. 8.10)$$

$$\dot{\emptyset} = -\dot{q}_4 \quad (Ec. 8.11)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\emptyset} \end{bmatrix} = \begin{bmatrix} -a_1 \cdot \sin(q_1) - a_2 \cdot \sin(q_1 + q_2) & -a_2 \cdot \sin(q_1 + q_2) & 0 & 0 \\ a_1 \cdot \cos(q_1) + a_2 \cdot \cos(q_1 + q_2) & a_2 \cdot \cos(q_1 + q_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (Ec. 8.12)$$

Siendo

$$J(q) = \begin{bmatrix} -a_1 \cdot \sin(q_1) - a_2 \cdot \sin(q_1 + q_2) & -a_2 \cdot \sin(q_1 + q_2) & 0 & 0 \\ a_1 \cdot \cos(q_1) + a_2 \cdot \cos(q_1 + q_2) & a_2 \cdot \cos(q_1 + q_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \quad (Ec. 8.13)$$

Y su determinante:

$$\det(J(q)) = (-a_1 \cdot \sin(q_1) - a_2 \cdot \sin(q_1 + q_2)) \cdot (a_2 \cdot \cos(q_1 + q_2)) + (a_1 \cdot \cos(q_1) + a_2 \cdot \cos(q_1 + q_2)) \cdot (a_2 \cdot \sin(q_1 + q_2)) \quad (Ec. 8.14)$$

Simplificando, obtenemos:

$$\det(J(q)) = a_1 \cdot a_2 \cdot \sin(q_2) \quad (Ec. 8.15)$$

Lo que nos da como singularidades los puntos donde $q_2 = n\pi$, y dentro del rango de trabajo de q_2 sólo se cumple para $n = 0$, entonces, nos encontraremos en un punto con singularidad cada vez que tengamos $q_2 = 0$, es decir, en todo nuestro límite de trabajo exterior.

También tendremos singularidades en cada punto del resto de nuestra frontera de trabajo, es decir, cuando q_2 sea máximo o mínimo en cualquier estado de q_1 y q_3 , con q_1 máximo y

mínimo para cualquier punto de q_2 y q_3 y, por último, cuando q_3 sea máximo o mínimo en cualquier punto de q_1 y q_2 .

Calculamos también velocidades articulares en función del Jacobiano y del vector de velocidades cartesianas.

$$\dot{q} = (J(q))^{-1} \cdot \dot{p} \quad (Ec. 8.16)$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} \frac{\cos(q_1 + q_2)}{a_1 \cdot \text{sen}(q_2)} & \frac{\text{sen}(q_1 + q_2)}{a_1 \cdot \text{sen}(q_2)} & 0 & 0 \\ \frac{-(a_1 \cdot \cos(q_1) + a_2 \cdot \cos(q_1 + q_2))}{a_2 \cdot a_1 \cdot \text{sen}(q_2)} & \frac{-(a_1 \cdot \text{sen}(q_1) + a_2 \cdot \text{sen}(q_1 + q_2))}{a_2 \cdot a_1 \cdot \text{sen}(q_2)} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \frac{-\cos(q_1)}{a_2 \cdot \text{sen}(q_2)} & \frac{-\text{sen}(q_1)}{a_1 \cdot \text{sen}(q_2)} & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} \quad (Ec. 8.15)$$

Para calcular las velocidades articulares en puntos de singularidad dados por un determinante nulo, debemos utilizar la pseudo inversa de nuestra matriz Jacobiana, esta se puede obtener en Matlab (incluida en el código) y por su extensión, no se mostrará en este informe.

Por último, mencionamos que en el código desarrollado en Matlab obtenemos los diferentes elipsoides de manipulabilidad, es decir, *la capacidad de cambio en posición y orientación del efector final en las diferentes configuraciones que va tomando nuestro robot en toda su trayectoria*.

9 PLANIFICACIÓN DE TRAYECTORIA

Para determinar las diferentes trayectorias que nuestro robot realizará, debemos tener presente el espacio en el que trabajamos y la aplicación para la cual se utilizará.

Para comenzar, estableceremos una posición articular de homing. Seleccionaremos por comodidad, $Q_h = [0, \frac{2}{3}\pi, 0, 0]$, $P_h = [0.3, 0.1732051, 0.05, 2.0944]$. Al encender nuestro robot, este debe ir de su posición actual hasta el homing (por conveniencia, la posición de homing será también la posición que tomará cada vez que finalice su tarea).

Luego del homing, se recibirá desde el procesamiento de visión artificial, el valor de nuestra nueva posición deseada, en coordenadas cartesianas x, y, z y \emptyset . Mediante nuestras ecuaciones de cinemática inversa, se calcularán las posibles soluciones en coordenadas articulares y se seleccionará una de ellas.

Ahora tenemos una posición articular anterior, la de homing, y una siguiente, en la plataforma vibradora. En esta última, la articulación q_3 se encontrará en su máximo valor o muy próximo a él, ya que la plataforma estará a nivel de la base. Para evitar colisiones con posibles bordes, para llegar a esta posición, estableceremos puntos intermedios, para el cual no es necesario tener gran precisión, es decir, nuestro efector final no debe pasar obligatoriamente por allí, para esto utilizamos la función `mstraj` del Toolbox de Peter Corke. Lo que hace esta función no es interpolar entre los puntos intermedios, sino que los aproxima.

Con esto, buscamos que nuestra trayectoria eleve la articulación prismática y la vuelva a bajar mediante una curva suave, evitando cualquier posible obstáculo o arrastre de las piezas. Para aumentar la precisión del efector final, antes de agarrar una pieza, nuestro brazo se detendrá sobre ella y bajará en línea recta hacia abajo, es decir que la única articulación que se moverá será la tercera, lo mismo a la hora de dejar una pieza en la bandeja. Esto se logró con la función `ctrj`, interpolando en coordenadas cartesianas. Y entre los puntos donde las articulaciones 1, 2 y 4 si se mueven, se interpoló en coordenadas articulares con la función `jtraj`, ya que no hay necesidad de que pasen por puntos específicos.

A continuación, se muestran las diferentes trayectorias para las articulaciones, sus velocidades y aceleraciones, primero desde una posición con la tercera articulación elevada (homing), hasta una final extendida, simulando un pick y luego desde ese pick al place, donde en ambos puntos q_3 es máxima y finalmente de allí a homing.

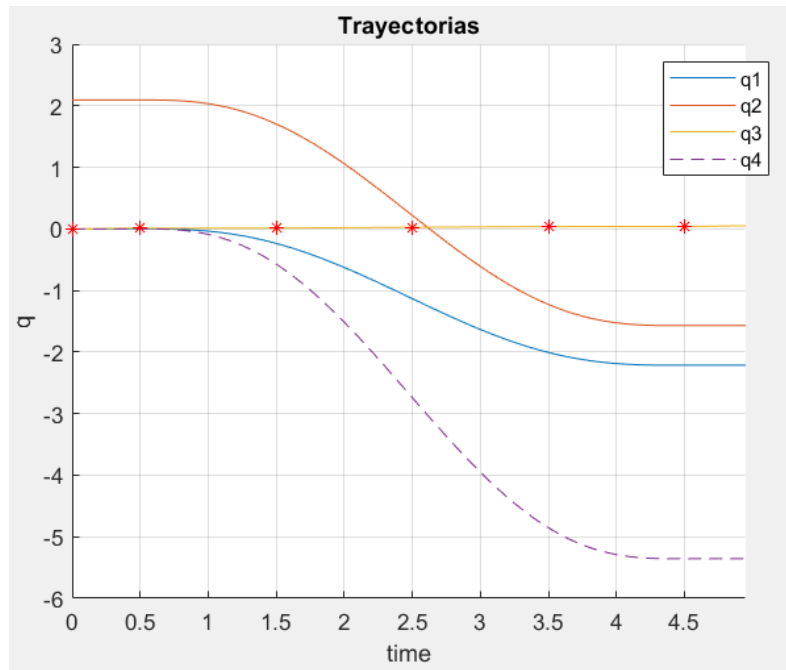


Figura 9.a: Gráfica de las trayectorias articulares en función del tiempo desde P_h hasta P_1

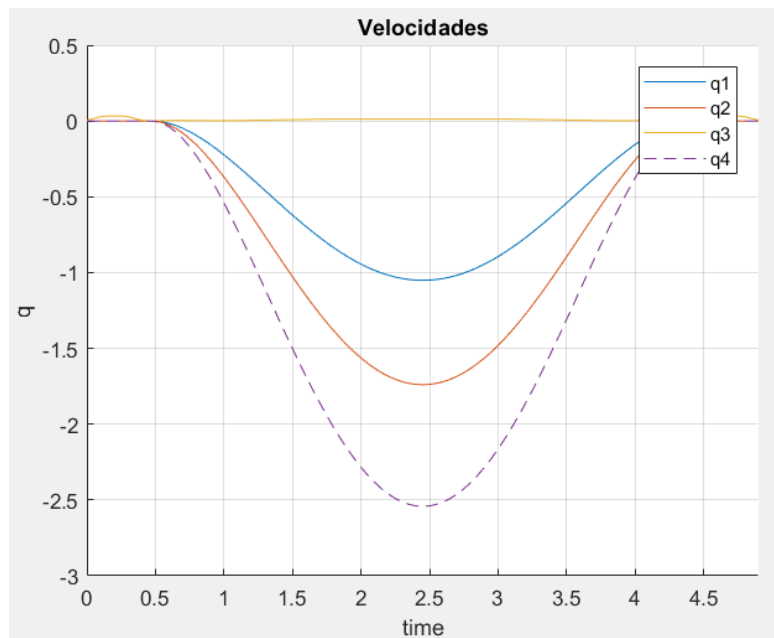


Figura 9.b: Gráfica de las velocidades articulares en función del tiempo desde P_h hasta P_1

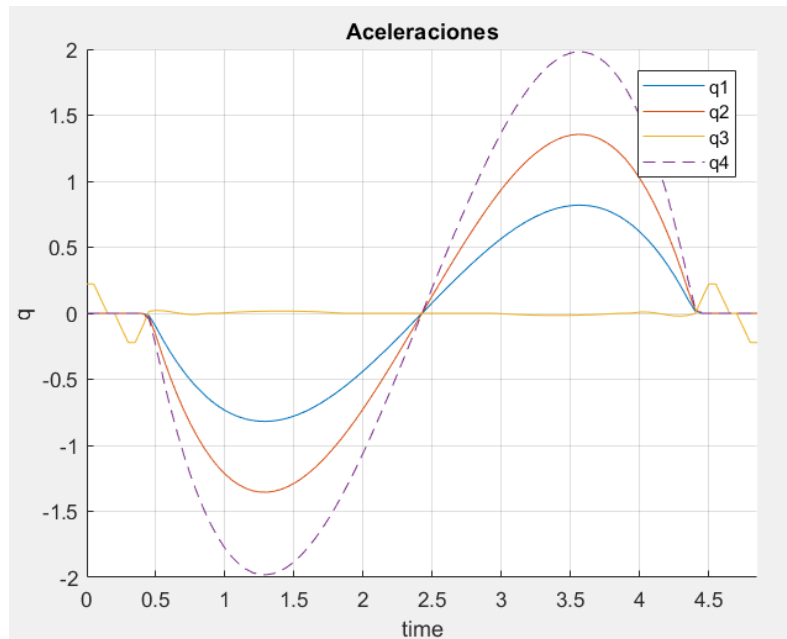


Figura 9.c: Gráfica de las aceleraciones articulares en función del tiempo desde P_h hasta P_1

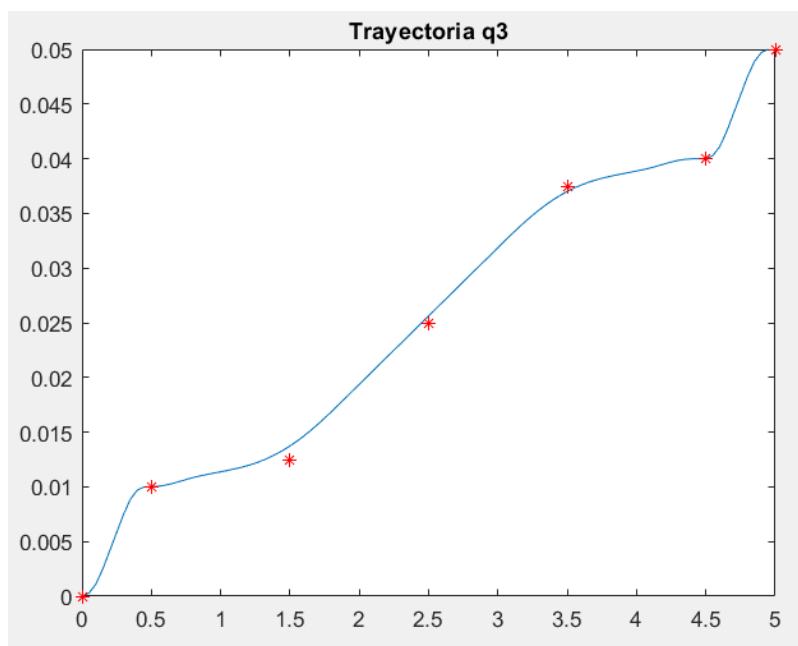


Figura 9.d: Gráfico de la trayectoria de la tercera articulación en función del tiempo desde P_h hasta P_1

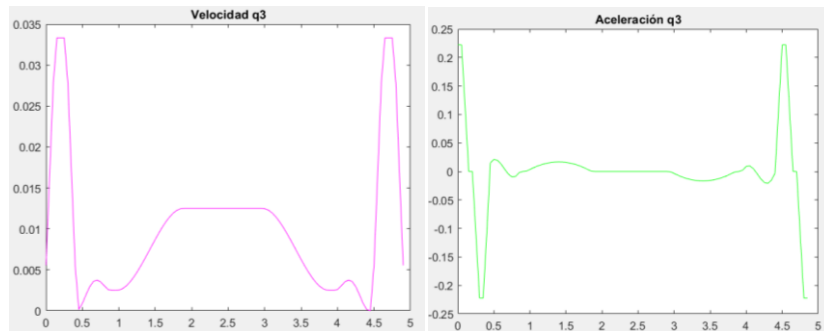


Figura 9.e: Gráficas de velocidad y aceleración de la tercera articulación en función del tiempo desde P_h hasta P_1

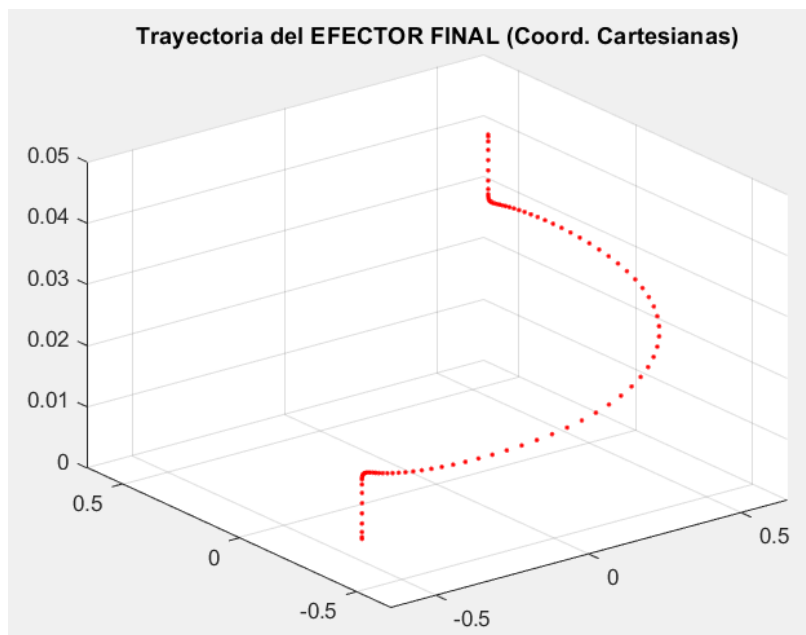


Figura 9.f: Trayectoria del efector final en coordenadas cartesianas desde P_h hasta P_1

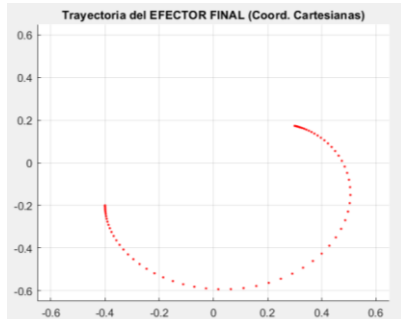


Figura 9.g: Proyección de la trayectoria en el plano x-y

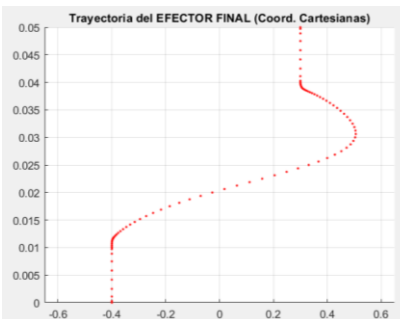


Figura 9.h: Proyección de la trayectoria en el plano x-z

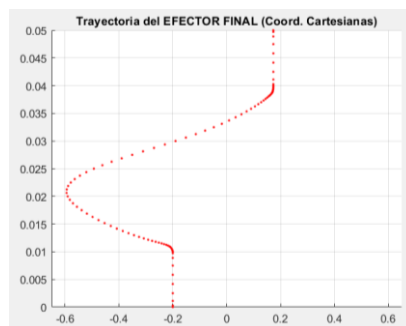


Figura 9.i: Proyección de la trayectoria en el plano y-z

Estas gráficas corresponden a una trayectoria desde homing:

$$P_h = [0.3, 0.1732051, 0.05, 2.0944] \text{ o } Q_h = [0, 2.0944, 0, 0],$$

hasta un punto de pick que llamaremos 1:

$$P_1 = [-0.4, -0.2, 0, 1.5708] \text{ o } Q_1 = [-2.2143, -1.5708, 0.05, -5.3559].$$

Las primeras tres coordenadas son xyz del espacio cartesiano en metros y el último parámetro es el ángulo \emptyset , en radianes, nombrado en la sección 7 de Cinemática Inversa.

Vemos que las velocidades requeridas no son tan elevadas como para exceder la velocidad límite de ninguna articulación.

El paso utilizado fue de 0,05 y el tiempo total es de 5 segundos.

Los asteriscos rojos marcan los puntos intermedios utilizados para conseguir la trayectoria, vemos que el efector final no pasa exactamente por estos puntos, sino que se aproxima a ellos.

Para la trayectoria de place, nos movemos desde el punto 1:

$$P_1 = [-0.4, -0.2, 0, 1.5708] \text{ o } Q_1 = [-2.2143, -1.5708, 0.05, -5.3559],$$

hasta el segundo punto, en la bandeja:

$$P_2 = [-0.5732051, -0.1, 0, 2.0944] \text{ o } Q_2 = [-0.3454, 0.5236, 0.05, -1.9162].$$

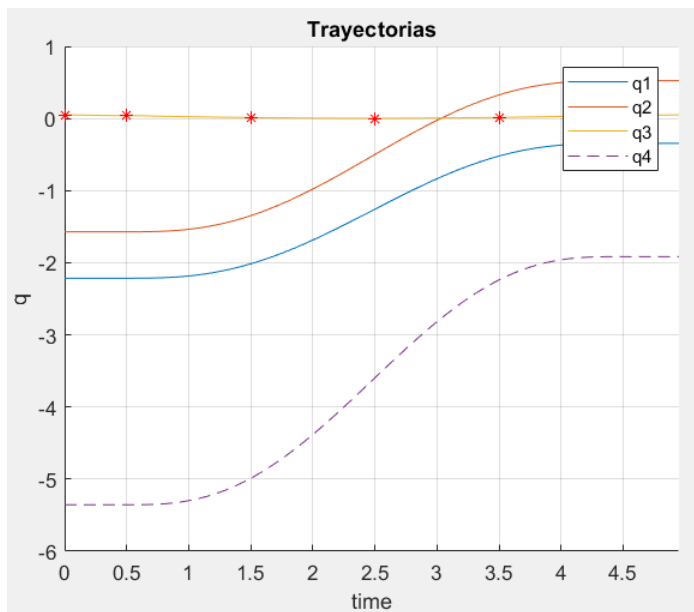


Figura 9.j: Gráfica de las trayectorias articulares en función del tiempo desde P_1 hasta P_2

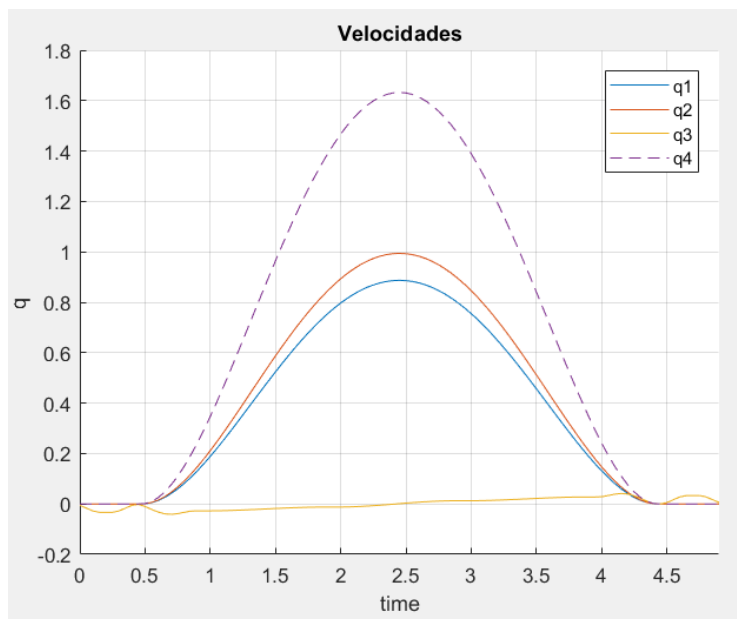


Figura 9.k: Gráfica de las velocidades articulares en función del tiempo desde P_1 hasta P_2

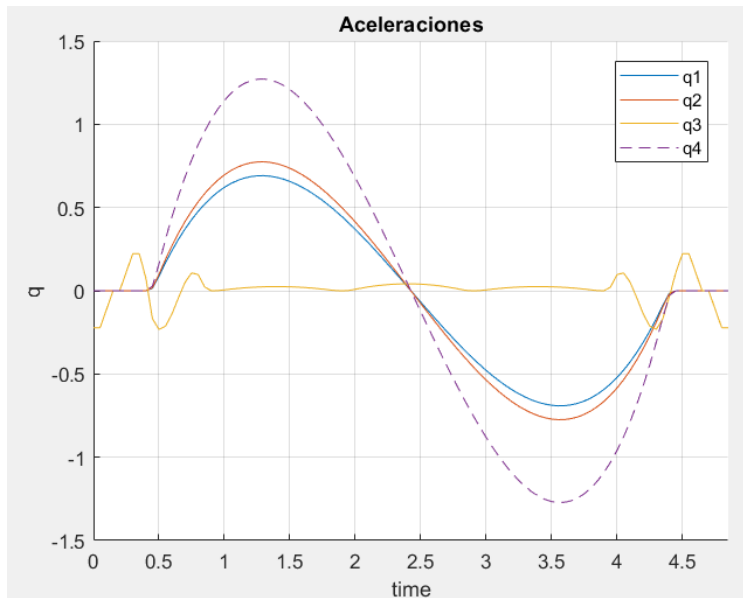


Figura 9.l: Gráfica de las aceleraciones articulares en función del tiempo desde P_1 hasta P_2

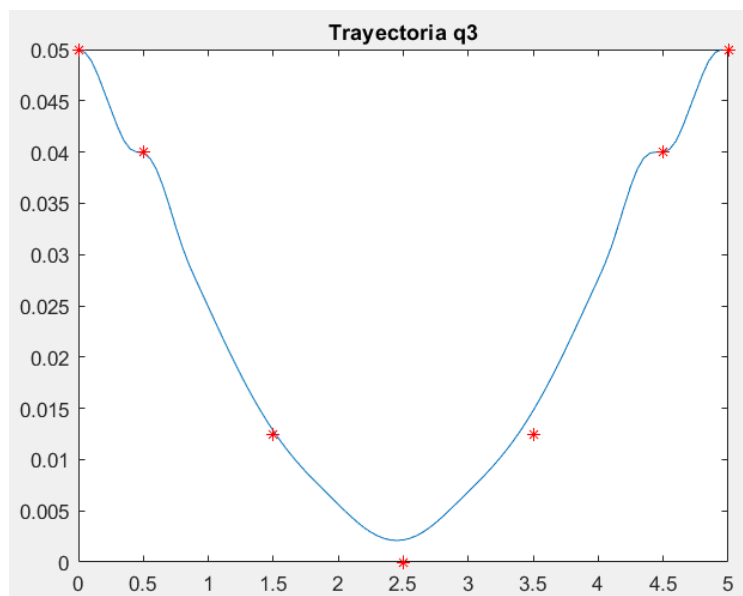


Figura 9.m: Gráfico de la trayectoria de la tercera articulación en función del tiempo desde P_1 hasta P_2

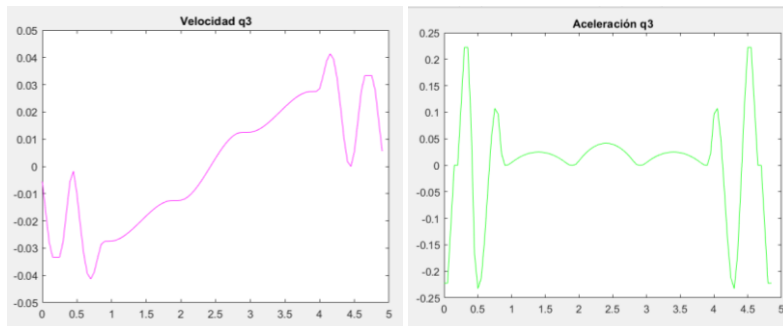


Figura 9.n: Gráficas de velocidad y aceleración de la tercera articulación en función del tiempo desde P_1 hasta P_2

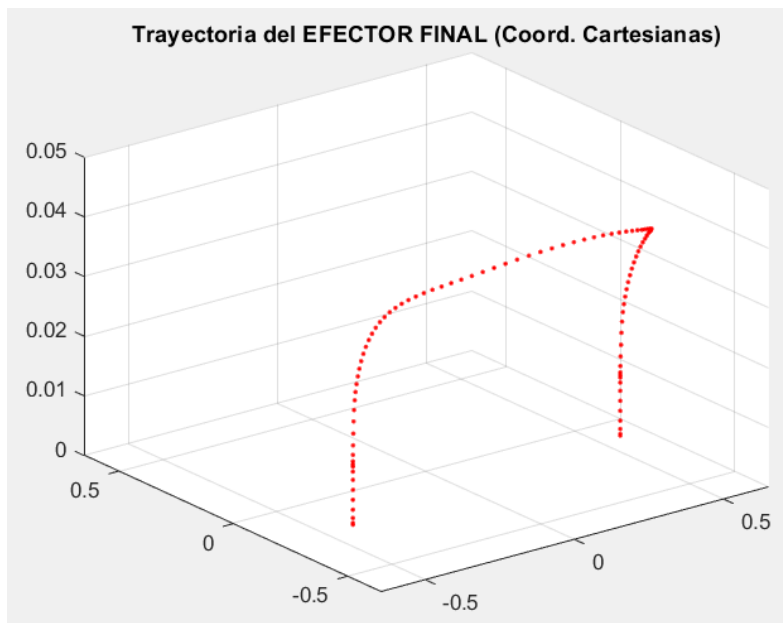


Figura 9.o.: Trayectoria del efector final en coordenadas cartesianas desde P_1 hasta P_2

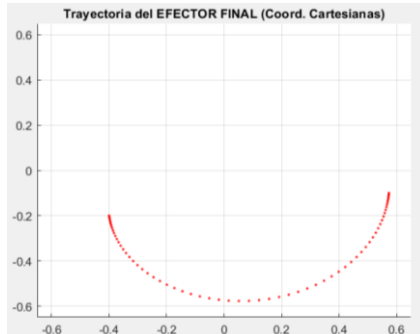


Figura 9.p: Proyección de la trayectoria en el plano x-y

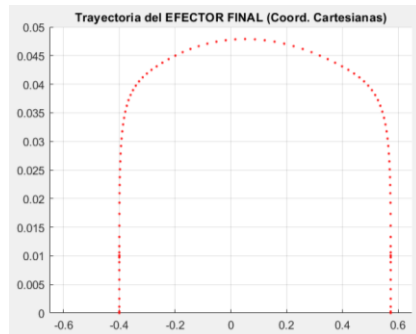


Figura 9.q: Proyección de la trayectoria en el plano x-z

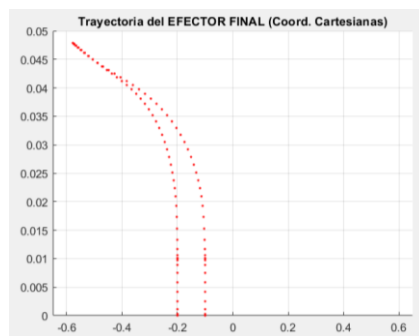


Figura 9.r: Proyección de la trayectoria en el plano y-z

Vemos que las velocidades requeridas no son tan elevadas como para exceder la velocidad límite de ninguna articulación. El paso utilizado fue de 0,05 y el tiempo total es de 5 segundos.

Para la última trayectoria desde el place hasta home, nos movemos desde el punto 2:

$P_2 = [-0.5732051, -0.1, 0, 2.0944]$ o $Q_2 = [-0.3454, 0.5236, 0.05, -1.9162]$, hasta la posición de homing:

$P_h = [0.3, 0.1732051, 0.05, 2.0944]$ o $Q_h = [0, 2.0944, 0, 0]$.

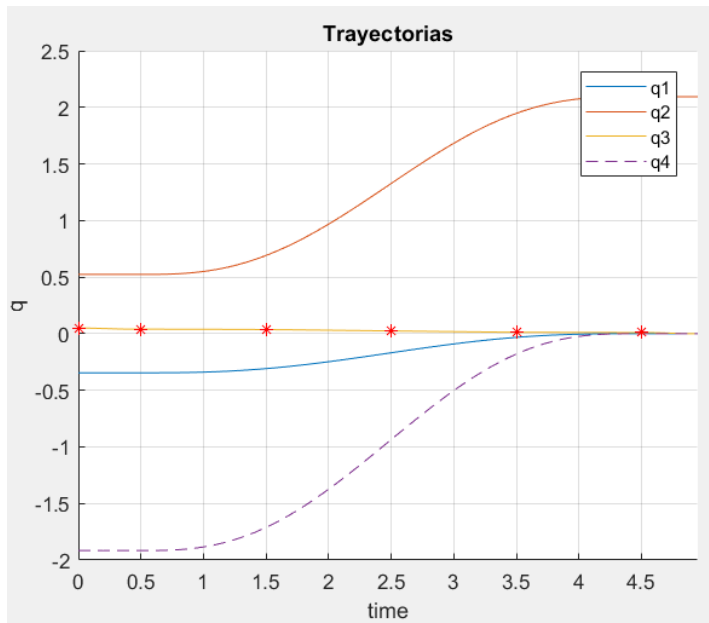


Figura 9.s: Gráfica de las trayectorias articulares en función del tiempo desde P_2 hasta P_h

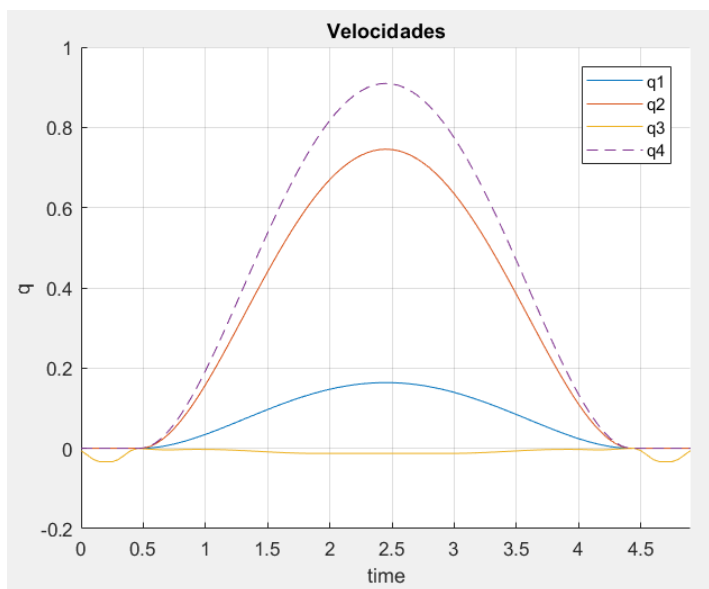


Figura 9.t: Gráfica de las velocidades articulares en función del tiempo desde P_2 hasta P_h

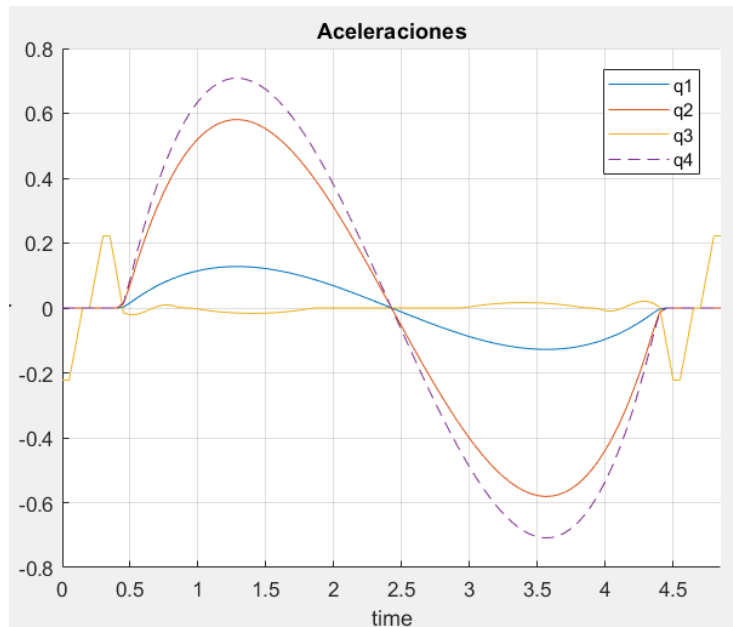


Figura 9.u: Gráfica de las aceleraciones articulares en función del tiempo desde P_2 hasta P_h

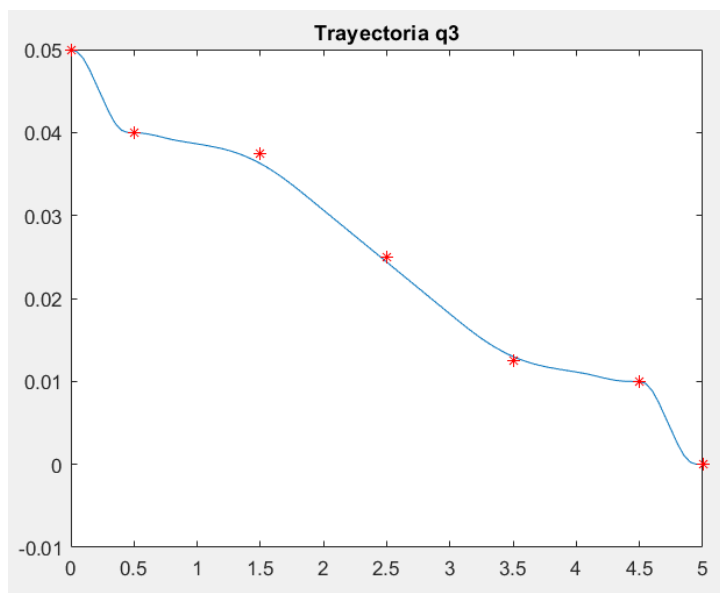


Figura 9.v: Gráfico de la trayectoria de la tercera articulación en función del tiempo desde P_2 hasta P_h

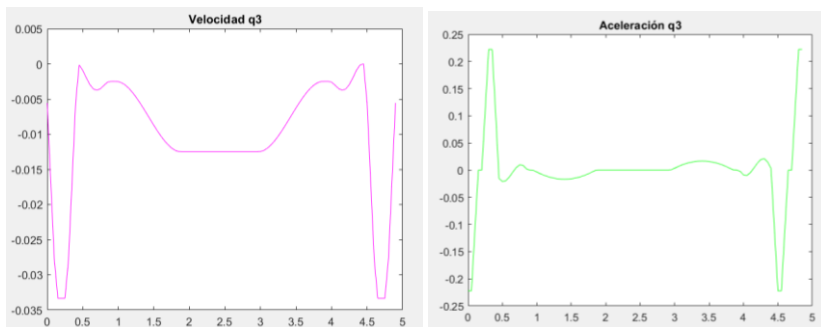


Figura 9.w: Gráficas de velocidad y aceleración de la tercera articulación en función del tiempo desde P_2 hasta P_h

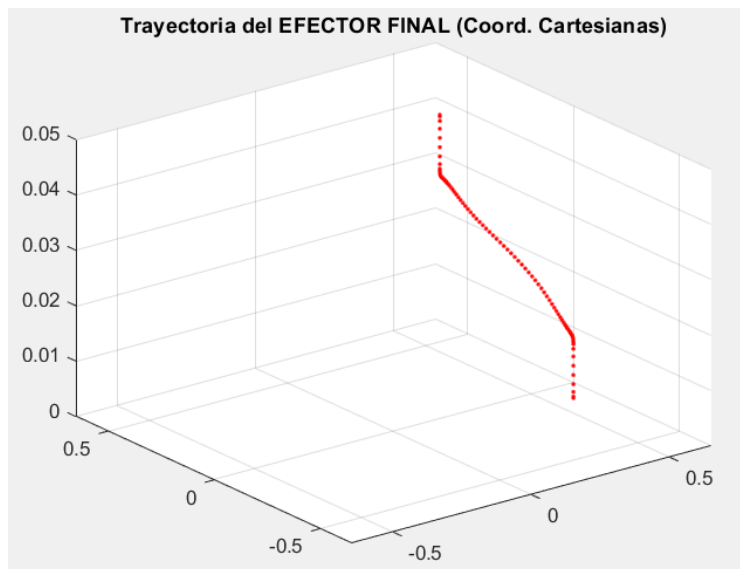


Figura 9.x.: Trayectoria del efector final en coordenadas cartesianas desde P_2 hasta P_h

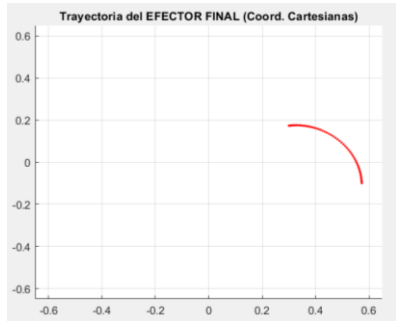


Figura 9.y: Proyección de la trayectoria en el plano x-y

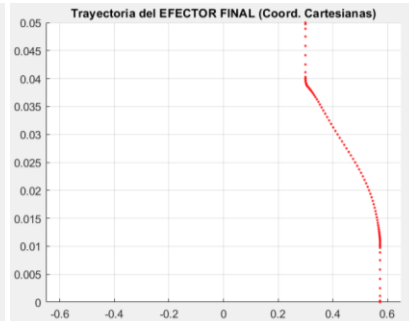


Figura 9.z: Proyección de la trayectoria en el plano x-z

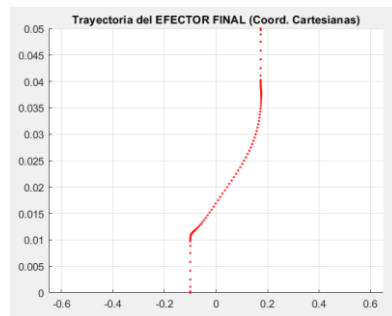


Figura 9.aa: Proyección de la trayectoria en el plano y-z

10 VALIDACIONES

Validamos el cálculo de nuestra Cinemática Directa comparando nuestro resultado analítico con el cálculo numérico.

Ingresamos una posición articular $Q_a_validar$, calculamos la transformación homogénea del efector final con la función `Cinem_Directa`, que obtiene el resultado a partir de nuestra matriz 0T_4 (Ec 6.9), y nos da $T_a_validar$ de la Figura 10.b.

Luego, obtenemos la Cinemática Directa del mismo vector, pero de manera numérica que realiza Matlab con la función `fkine` del Toolbox de Peter Corke, nos da T_fkine de la Figura 10.b.

```
21 - Q_a_validar = [pi, pi/2, 0.03, pi/2];
22
23 - T_a_validar = Cinem_Directa(Q_a_validar(1, :), a1, a2, d1, d3, d4, lim);
24
25 - T_fkine = R.fkine(Q_a_validar(1, :));
26
27 - phi = atan2(T_a_validar(2, 1), T_a_validar(1, 1));
28
29 - Q_validada = Cinem_Inversa(a1, a2, d1, d3, d4, T_a_validar,
30     T_a_validar(1, 4), T_a_validar(2, 4), T_a_validar(3, 4), phi, lim);
31
32 - error_cd = T_a_validar - T_fkine;
33
```

Figura 10.a: Código de validación de Cinemática Directa calculada analíticamente

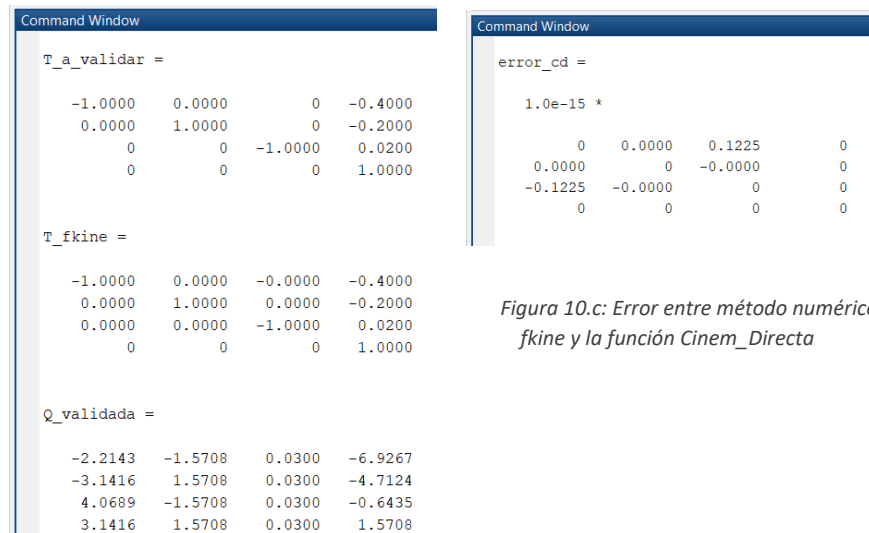


Figura 10.c: Error entre método numérico `fkine` y la función `Cinem_Directa`

Figura 10.b: Resultado de validación de Cinemática Directa calculada analíticamente

Comentado [NBC1]: Agregar figura

Comentado [NBC2]: ¡!

Comentado [NBC3]: ¡!

Como resultado, nos muestra que las matrices son prácticamente iguales. Si restamos ambas matrices nos da el error de la Figura 10.c. Lo suficientemente pequeño para considerar nuestro cálculo numérico como correcto.

Comentado [NBC4]: ¡!

Por último, $Q_validada$, nos muestra cómo nuestra $T_a_validar$ nos vuelve a dar el vector de posiciones articulares inicial. En realidad, devuelve cuatro posibles soluciones, entre las cuales encontramos $Q_a_validar$.

Si proponemos una matriz de transformación homogénea, podemos calcular su Cinemática Inversa con nuestras ecuaciones analíticas y con el método numérico para poder validar nuestros cálculos.

```
38 - T_a_validar2 = [-1, 0, 0, -0.4;
39 -               0, 1, 0, -0.2;
40 -               0, 0, -1, 0.02;
41 -               0, 0, 0, 1];
42
43 - Q_a_validar2 = Cinem_Inversa(a1, a2, d1, d3, d4, T_a_validar,
44 -                             T_a_validar(1, 4), T_a_validar(2, 4), T_a_validar(3, 4), phi, lim);
45
46 - Q_a_validar2_best = Best_Q(Q_a_validar2, Q_0, a1, a2);
47
48 - Q_ikine = R.ikine(T_a_validar2, Q_0, [1 1 1 0 0 1]);
49
50 - T_validada = Cinem_Directa(Q_a_validar2_best(1, :), a1, a2, d1, d3, d4, lim);
51
```

Figura 10.d: Código de validación de Cinemática Inversa calculada analíticamente

```
Command Window

Q_a_validar2 =

-2.2143 -1.5708 0.0300 -6.9267
-3.1416 1.5708 0.0300 -4.7124
4.0689 -1.5708 0.0300 -0.6435
3.1416 1.5708 0.0300 1.5708

Q_a_validar2_best =

3.1416 1.5708 0.0300 1.5708

Q_ikine =

3.1416 1.5708 0.0300 1.5708

T_validada =

-1.0000 0.0000 0 -0.4000
0.0000 1.0000 0 -0.2000
0 0 -1.0000 0.0200
0 0 0 1.0000
```

Ingresamos una transformación homogénea $T_a_validar2$, calculamos las coordenadas articulares con la función $Cinem_Inversa$, que utiliza nuestras ecuaciones de q_1 (Ec. 7.3), q_2 (Ec. 7.11), q_3 (Ec. 7.13) y q_4 con (Ec. 7.21) o (Ec. 7.22), y nos da $Q_a_validar2$, que puede variar entre dos, tres y cuatro soluciones según la posición del efector final. De estas soluciones calculamos la más cercana a la posición anterior, Q_0 , con nuestra función $Best_Q$, que finalmente nos da nuestro vector seleccionado $Q_a_validar2_best$, de la Figura 10.e.

Comentado [NBC5]: ¡!

Además, calculamos con métodos numéricos que realiza Matlab con la función $ikine$ del Toolbox de Peter Corke, nos da T_ikine de la Figura 10.e.

Comentado [NBC6]: ¡!

$Q_a_validar2$ nos muestra las cuatro soluciones posibles de la Cinemática Inversa.

Figura 10.f: Resultado de validación de Cinemática Inversa calculada analíticamente $Q_a_validar2_best$ es la solución más cercana a la anterior.

T_{ikine} es la transformación homogénea calculada numéricamente.

```
Command Window
error_ci =
1.0e-07 *
0.0001 -0.0001 0.0000 -0.2633
```

Figura 10.g: Error entre método numérico ikine y la función *Cinem_Inversa*

Los resultados muestran que ambas soluciones son casi las mismas, con el error mostrado en la Figura 10.g.

Por último, $T_{validada}$, nos muestra cómo esta solución nos vuelve a dar nuestra transformación homogénea del principio al aplicar Cinemática Directa.

También podemos validar los cálculos realizados para obtener el Jacobiano.

Al igual que las validaciones anteriores, utilizamos nuestra propia función y luego el método numérico. Diferenciando nos da el error.

```
69 - Q_0 = [pi, pi*2/3, 0, pi/2];
70
71 - J = Jacobiano(Q_0, a1, a2);
72
73 - J_jacob0 = R.jacob0(Q_0);
```

Figura 10.h: Código de validación del Jacobiano calculado analíticamente

```
Command Window
J_jacob0 =
0.1732 0.1732 -0.0000 0
-0.3000 0.1000 -0.0000 0
-0.0000 -0.0000 -1.0000 0
1.0000 1.0000 0 -1.0000

Rango del Jacobiano = 4
Determinante del Jacobiano = 0.069282
0 linearly dependent joints:

J =
0.1732 0.1732 0 0
-0.3000 0.1000 0 0
0 0 -1.0000 0
1.0000 1.0000 0 -1.0000
```

```
Command Window
error_J =
1.0e-15 *
0.0278 0.0555 0.1061 0
0 -0.0416 0.0612 0
0.0000 0.0000 0 0
0 0 0 0
```

Figura 10.j: Error entre el método numérico *jacob0* y la función *Jacobiano*

Figura 10.i: Resultado de validación del Jacobiano calculado analíticamente

También probamos con un vector Q_0 en una singularidad. Vemos que se pierde un grado de libertad, disminuye el rango de la matriz.

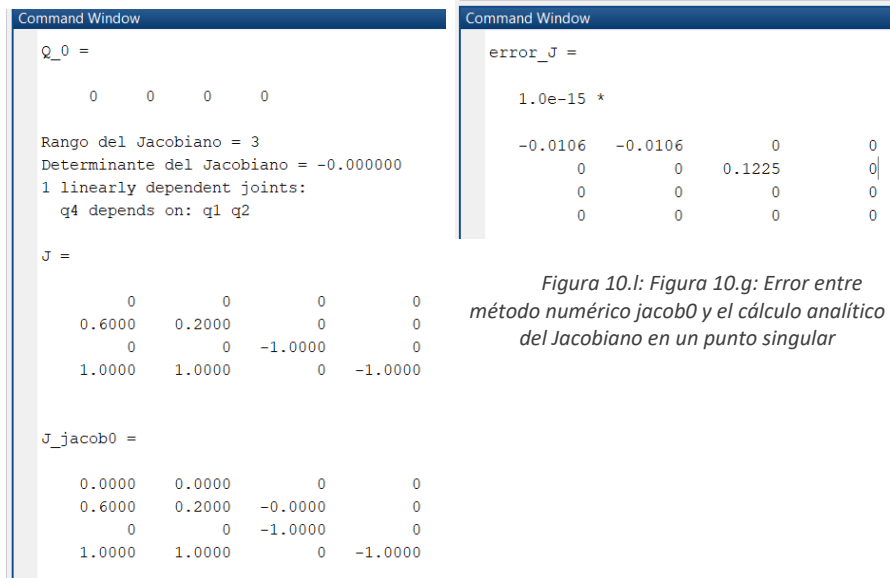


Figura 10.i: Resultado de validación del Jacobiano calculada analíticamente en un punto singular

Las funciones utilizadas pueden verificarse en el archivo adjunto GENERAL_BOLD0.m

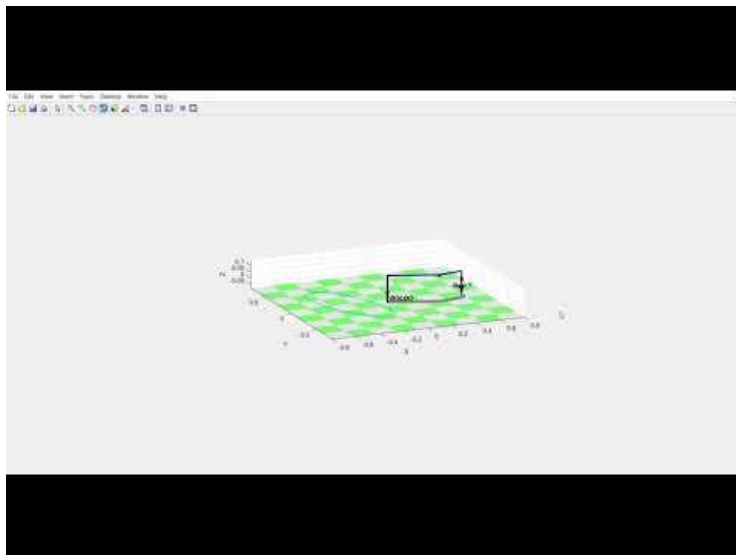
11 APORTE

Se realizó a partir de los cálculos descriptos anteriormente una simulación en Matlab del Robot, con el Toolbox de Peter Corke.

El código comienza con la matriz de Denavit y Hartenberg, con la que se crea el objeto R al utilizar la clase SerialLink. Luego definimos algunos parámetros importantes para su implementación, como, por ejemplo, los límites articulares, las velocidades máximas, la posición de homing, etc. Lo más importante a definir son los puntos a los que irá, es decir, estos deberían ser otorgados por el sistema de visión artificial, pero por ahora lo que hacemos es crear una matriz donde cada fila es un vector posición. Establecemos que comience del homing, luego que tome una pieza de la plataforma como segundo punto, y que la deje en la bandeja en tercer lugar. A partir de ahí, simulamos que ha terminado su tarea, por ende, deberá ir de nuevo a homing.

El resto de las funciones importantes ya fueron mencionadas a lo largo del informe.

El código se adjuntará en el archivo como anexo y también un video de la animación.





Además, realizamos el diseño real con el objetivo de su construcción, cabe destacar que el diseño aún se encuentra en revisión. Nuestro Robot tendrá partes impresas en 3D, en ABS para tener una mayor resistencia a los esfuerzos. También se utilizará perfiles Bosh 2020 de aluminio, para reducir el peso total.

Se agregó sobre el motor de la primera articulación un reductor planetario **38, 4: 1**, para transmitir un torque mayor, capaz de soportar el resto del brazo.

Desde el motor que se encuentra en voladizo hasta la segunda articulación, se utilizará una correa GT2, con reducción de **2, 13: 1**, que no sale en las figuras.

Agregaremos el archivo .stl como anexo y mediante el siguiente enlace se puede acceder al ensamblaje y sus partes. <https://drive.google.com/open?id=1zFLUjijKgD25rXvO59loRjSCTdiGk-26V>

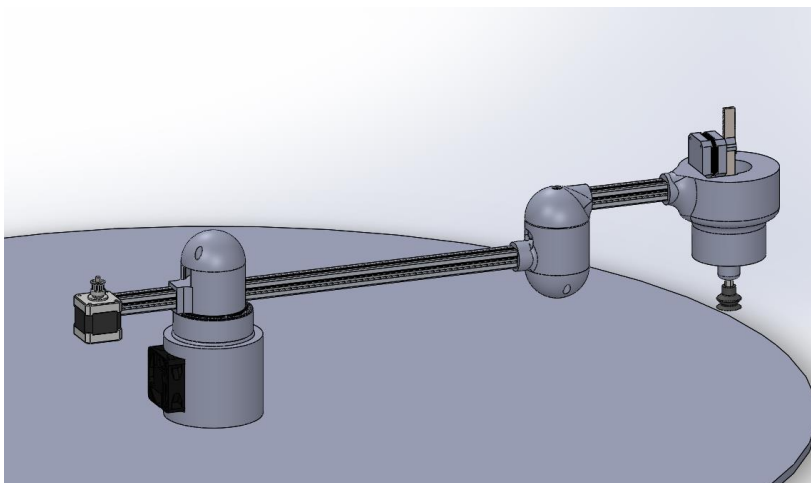


Figura 11a: Brazo en posición articular $Q = [0, 0, 0, 0]$

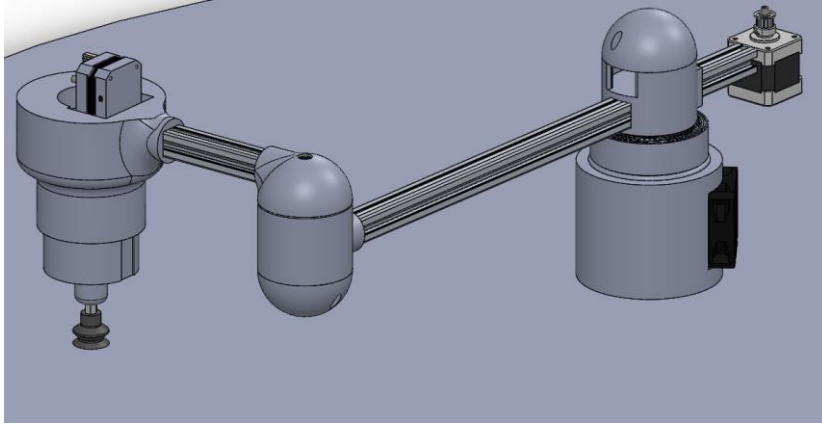


Figura 11b: Brazo en posición articular $Q = [\pi, \pi, 0, 0]$

12 ACTUADORES

Los actuadores a utilizar son motores reciclados de una fotocopidora en desuso, por lo tanto, sus datos técnicos no están disponibles o no son fáciles de conseguir.

Para la primera articulación (rotacional), se utilizará un NEMA23, modelo STP-58D3023-01,



con paso de $1,8^\circ/\text{step}$, voltaje $V = 2,0\text{ V}$, corriente máxima $I_{\text{máx}} = 2,1\text{ A}$ y velocidad máxima $n_{\text{máx}_m} = 500\text{ rpm}$, con una reducción de 38,4: 1, lo que nos da una velocidad final en la articulación de $\omega_{\text{máx}} = 1,3635\text{ rad/s}$. Fabricante SHINANO KENSHI CORP.

Figura 12.a: Motor paso a paso NEMA 23 para la primera articulación

La segunda articulación (rotacional), llevará un NEMA17, modelo B3124121A/AX050154A, con paso de $1,8^\circ/\text{step}$, voltaje $V = 3,57\text{ V}$, corriente máxima $I_{\text{máx}} = 1,7\text{ A}$ y velocidad máxima



$n_{\text{máx}_m} = 350\text{ rpm}$, con una transmisión por correa GT2, con reducción de 2,13: 1, (polea dentada menor: 15 dientes y mayor: 32 dientes), lo que nos da una velocidad final en la articulación de $\omega_{\text{máx}} = 17,2074\text{ rad/s}$. Fabricante JAPAN SERVO CO.

Figura 12.b: Motor paso a paso NEMA 17 para la segunda articulación



En la tercera articulación (prismática), utilizaremos un NEMA17, modelo AX050213A/ B70005806A, con paso de $1,8^\circ/\text{step}$, voltaje $V = 2,52\text{ V}$, corriente máxima $I_{\text{máx}} = 0,9\text{ A}$ y velocidad máxima $n_{\text{máx}_m} = 3,33\text{ rps}$, con un sistema piñón - cremallera, con diámetro $d_{p-\text{piñón}} = 10,2\text{ mm}$, por lo tanto la velocidad tangencial es $v_{tg} = n_{\text{máx}_m} \cdot r_{p-\text{piñón}} = 16,983\text{ mm/s}$. Fabricante JAPAN SERVO CO.

Figura 12.c: Motor paso a paso NEMA 17 para la tercera articulación.

Finalmente, para la última articulación (rotacional), utilizaremos un NEMA17 bipolar, modelo EM - 434, con paso de $1,8^\circ/\text{step}$, voltaje $V = 5 - 6\text{ V}$, corriente máxima $I_{\text{máx}} = 1,4\text{ A}$ y velocidad máxima $n_{\text{máx}} = 2,5\text{ rps}$, $\omega_{\text{máx}} = 15,708\text{ rad/s}$, sin reducción. Estos datos son empíricos, del siguiente enlace, donde dice que fueron obtenidos al probar un modelo muy similar EM – 483. <https://www.instructables.com/id/EM-483-Specifications/>

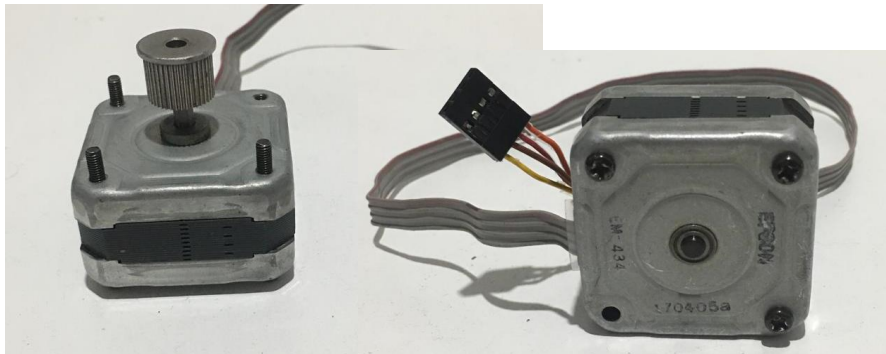


Figura 12.d: Motor paso a paso NEMA 17 para la cuarta articulación.

Además, para una segunda versión de nuestro robot, lo ideal sería seleccionar motores comerciales con prestaciones similares a los actuadores utilizados, para esto recurriremos al catálogo de motores paso a paso de SHINANO KENSHI CORP.

Para la primera articulación, elegimos el NEMA23, modelo SST57D3200, con paso de $1,8^\circ/\text{step}$, voltaje $V = 3,2\text{ V}$ y corriente máxima $I_{\text{máx}} = 2,0\text{ A}$.

SPECIFICATION

MODEL		STEP ANGLE	VOLTAGE	CURRENT	RESISTANCE	INDUCTANCE	HOLDING TORQUE	ROTOR INERTIA	NUMBER OF LEADS	WEIGHT	DIMENSION
SINGLE SHAFT	DOUBLE SHAFT	DEG.	V	A/Phase	Ω/Phase	mH/Phase	kg-cm	g-cm ²	LEAD	kg	L
SST57D1100	SST57D1101	1.8	3.9	1.0	3.9	6.5	4.2	110	6	0.4	39
SST57D1200	SST57D1201	1.8	2.0	2.0	1.0	1.5	4.2	110	6	0.4	39
SST57D1300	SST57D1301	1.8	1.6	3.0	0.53	0.74	4.2	110	6	0.4	39
SST57D2100	SST57D2101	1.8	5.4	1.0	5.4	10.7	7.6	200	6	0.5	49
SST57D2200	SST57D2201	1.8	3.0	2.0	1.5	3.0	7.6	200	6	0.5	49
SST57D2300	SST57D2301	1.8	2.1	3.0	0.7	1.3	7.6	200	6	0.5	49
SST57D3100	SST57D3101	1.8	6.7	1.0	6.7	13	9.7	250	6	0.6	54
SST57D3200	SST57D3201	1.8	3.2	2.0	1.6	3.6	9.7	250	6	0.6	54
SST57D3300	SST57D3301	1.8	2.5	3.0	0.83	1.8	9.7	250	6	0.6	54
SST57D4100	SST57D4101	1.8	7.6	1.0	7.6	16	11.7	330	6	0.75	64
SST57D4200	SST57D4201	1.8	4.0	2.0	2.0	4.5	11.7	330	6	0.75	64
SST57D4300	SST57D4301	1.8	2.85	3.0	0.95	2.1	11.7	330	6	0.75	64
SST57D5100	SST57D5101	1.8	9.2	1.0	9.2	20	14.3	430	6	0.95	75
SST57D5200	SST57D5201	1.8	4.8	2.0	2.4	5.7	14.3	430	6	0.95	75
SST57D5300	SST57D5301	1.8	3.42	3.0	1.14	2.6	14.3	430	6	0.95	75

Figura 12.e: Cuadro de características de motores NEMA 23 de fabricante SHINANO KENSHI

En la segunda articulación, elegimos el NEMA17, modelo SST42D2120, con paso de $1,8^\circ/\text{step}$, voltaje $V = 3,7\text{ V}$ y corriente máxima $I_{\text{máx}} = 1,2\text{ A}$.

SPECIFICATION

MODEL		STEP ANGLE	VOLTAGE	CURRENT	RESISTANCE	INDUCTANCE	HOLDING TORQUE	ROTOR INERTIA	NUMBER OF LEADS	WEIGHT	DIMENSION
SINGLE SHAFT	DOUBLE SHAFT	DEG.	V	A/Phase	Ω /Phase	mH/Phase	kg-cm	g-cm ²	LEAD	kg	L
SST42D1100	SST42D1101	1.8	3.7	0.95	3.9	3.6	1.9	27	6	0.18	31
SST42D1070	SST42D1071	1.8	5.3	0.7	7.6	6.8	1.9	27	6	0.18	31
SST42D1040	SST42D1041	1.8	10.5	0.35	30	21.7	1.7	27	6	0.18	31
SST42D1020	SST42D1021	1.8	16.5	0.22	75	53.0	1.7	27	6	0.18	31
SST42D120	SST42D121	1.8	3.7	1.2	3.1	4.2	3.2	48	6	0.27	38
SST42D2090	SST42D2091	1.8	5.1	0.9	5.7	6.8	3.2	48	6	0.27	38
SST42D2070	SST42D2071	1.8	6.7	0.7	9.5	11.8	3.2	48	6	0.27	38
SST42D2040	SST42D2041	1.8	12.0	0.4	30	34.3	3.2	48	6	0.27	38
SST42D2030	SST42D2031	1.8	18.8	0.25	75	72.8	3.0	48	6	0.27	38

Figura 12.f: Cuadro de características de motores NEMA 17 de fabricante SHINANO KENSHI

Para la tercera y cuarta articulación, elegimos el NEMA17, modelo SST42D1100, con paso de $1,8^\circ/\text{step}$, voltaje $V = 3,7\text{ V}$ y corriente máxima $I_{m\acute{a}x} = 0,95\text{ A}$.

SPECIFICATION

MODEL		STEP ANGLE	VOLTAGE	CURRENT	RESISTANCE	INDUCTANCE	HOLDING TORQUE	ROTOR INERTIA	NUMBER OF LEADS	WEIGHT	DIMENSION
SINGLE SHAFT	DOUBLE SHAFT	DEG.	V	A/Phase	Ω /Phase	mH/Phase	kg-cm	g-cm ²	LEAD	kg	L
SST42D1100	SST42D1101	1.8	3.7	0.95	3.9	3.6	1.9	27	6	0.18	31
SST42D1070	SST42D1071	1.8	5.3	0.7	7.6	6.8	1.9	27	6	0.18	31
SST42D1040	SST42D1041	1.8	10.5	0.35	30	21.7	1.7	27	6	0.18	31
SST42D1020	SST42D1021	1.8	16.5	0.22	75	53.0	1.7	27	6	0.18	31
SST42D120	SST42D121	1.8	3.7	1.2	3.1	4.2	3.2	48	6	0.27	38
SST42D2090	SST42D2091	1.8	5.1	0.9	5.7	6.8	3.2	48	6	0.27	38
SST42D2070	SST42D2071	1.8	6.7	0.7	9.5	11.8	3.2	48	6	0.27	38
SST42D2040	SST42D2041	1.8	12.0	0.4	30	34.3	3.2	48	6	0.27	38
SST42D2030	SST42D2031	1.8	18.8	0.25	75	72.8	3.0	48	6	0.27	38

Figura 12.g: Cuadro de características de motores NEMA 17 de fabricante SHINANO KENSHI

Podemos ver el catálogo en el siguiente enlace:
<http://www.aet.calu.edu/ftp/cet/335/SKCMotors.pdf>

13 SENSORES

Se utilizarán dos detectores ópticos de barrera (TCST2103) como fin de carrera de la primera articulación para realizar el homing, uno para indicar el final a $q_1 = 0$ y otro por si la rotación intenta ir hacia cero para un lado donde hay un límite físico, entonces se colocará en $q_1 = \pi$.

Además, para los límites de las otras articulaciones, pondremos seis finales de carrera del tipo Microswitch Pulsador Interruptor V-5F932DN.



Colocaremos uno en el máximo de la segunda articulación, en $q_2 = \frac{2}{3} \cdot \pi$ y en el mínimo, $q_2 = -\frac{2}{3} \cdot \pi$. También cuando q_3 esté totalmente estirada $q_3 = 50$ y cuando se encuentre elevada $q_3 = 0$. Finalmente, para la última articulación, en $q_4 = -\pi$ y en $q_4 = \pi$.

Figura 13.a: Fin de carrera pulsador

Por último, para el sistema de visión artificial, colocaremos una Cámara Raspberry pi HD.

14 CONCLUSIÓN

Para terminar con nuestro informe, debemos resaltar que se logró implementar correctamente los conceptos aprendidos en clase, gracias a esto, la simulación muestra un robot eficiente y que cumple con la tarea asignada dentro de los parámetros solicitados. Todo esto, fue también gracias a la implementación de un método para optimizar la trayectoria, donde se optaba por la solución más cercana a la posición anterior, disminuyendo el tiempo de trabajo y la velocidad requerida. El cálculo de trayectoria nos dio precisión al llegar a las posiciones deseadas y también nos permitió evadir obstáculos en el espacio de trabajo, sin exceder velocidades máximas.

Por otro lado, se pudo realizar un diseño que se adapta a las necesidades de la aplicación, con materiales y elementos reciclados, obteniendo un bajo costo de construcción. También fue pensado para minimizar el peso total, especialmente la parte que se encuentra en voladizo y también, para que las articulaciones soporten el menor momento posible, ya que, al ser muy extenso, genera muchos problemas de inercia.

Como el diseño aún se encuentra en desarrollo, nuestro futuro trabajo será terminar la revisión del mismo, imprimirlo, ensamblarlo y testear su funcionamiento. Aún se debe verificar si los actuadores son los correctos para esta aplicación. También se debe agregar el reconocimiento por visión artificial y calibrar la electrónica que controla los actuadores.

Otro punto a testear es la combinación de partes impresas en 3D y los perfiles de aluminio, para verificar el rendimiento.