

Informe Final de Robótica 1

Robot de limpieza láser

Agustín Costarelli. Legajo 10966

Agustín Lezcano. Legajo 11956

27 de noviembre de 2020

Índice

1. Introducción	2
2. Presentación formal y técnica	3
3. Denavit y Hartenberg	5
4. Cinemática Directa	8
5. Cinemática Inversa	9
6. Jacobiano	10
7. Trayectorias	13
8. Sensores y actuadores	17
8.1. Sensores de posición	18
8.2. Sensor de proximidad para el efector y para el brazo	18
8.3. Medición de polvo	18
8.4. Motor	19
9. Conclusiones	20

Resumen

En el siguiente trabajo se desarrolla la aplicación de un brazo robot de 6 grados de libertad para la tarea de limpieza láser. Se plantea buscar la limpieza de una figura cuadrada genérica, este proceso sucede en una fábrica, es decir que las piezas tienen que ir hasta la posición del brazo. Haciendo uso de matlab se implementará el conocimiento del curso de Robotica 1 para llevar a cabo la simulación de trayectoria y todo los estudios que esta conlleva para la correcta implementación de la tarea.

1. Introducción

Para el desarrollo del trabajo final se ha elegido la opción de un robot limpiador de superficies, equipado con un limpiador láser. La limpieza láser se utiliza para remover pintura, grasas, óxido y desechos de piezas en su mayoría metálicas pero también puede aplicarse en otros materiales como la limpieza de paredes y suelos.

Dada la complejidad de la tarea y para que el robot se adapte a distintas piezas a limpiar se eligió una cadena cinemática abierta de 6 grados de libertad rotacionales. En el extremo final del robot se propone un limpiador láser de fibra óptica. Para la aplicación deseada se buscó una máquina de gran potencia que es un equipo que pesa alrededor de 600 kg. Se analizarán los movimientos del brazo robótico para que no representen un problema al trabajar con la manguera del limpiador. El caso de estudio será la implementación del brazo limpiador en una planta, es decir que las piezas serán llevadas hasta la posición de trabajo del robot.

Para el desarrollo de este trabajo se optó por un cubo de 0,6 metros de lado como ejemplo de implementación, podría considerarse al mismo como un cubo de calibración para verificar el correcto funcionamiento del robot y el seguimiento de la consigna.

Para este desarrollo se realizaron simplificaciones tanto en el diseño, ya que se optó por trabajar con el Toolbox de Peter Corke, como en la dinámica del robot ya que sobrepasa los alcances de este estudio en particular. Entonces el análisis será desde el punto de vista cinemático.



(a) KUKA KR8



(b) Efecto a emplear

Figura 1: Brazo y efector designados para la tarea

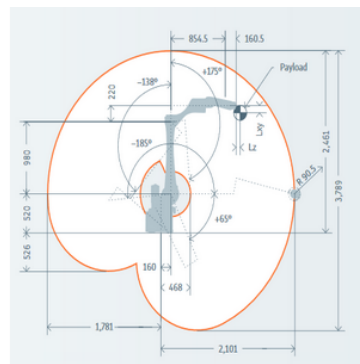
2. Presentación formal y técnica

El robot con el cual se trabajará es un KR 8 R2100-2 arc HW de la marca KUKA. Es un brazo que cuenta con todas sus articulaciones rotacionales, posee una capacidad de carga de 8kg y un alcance máximo de 2,1m. En las siguientes imágenes se puede ver al brazo con sus dimensiones junto con el espacio de trabajo y los límites articulares dados por el fabricante.

KR CYBERTECH arc		KR 8 R2100-2 arc HW	
Max. reach		2,101 mm	
Rated payload		8 kg	
Rated suppl. load, arm/link arm/rot.		10 kg / - / -	
Rated total load		18 kg	
Pose repeatability		±0.04 mm	
Number of axes		6	
Mounting position		Floor, ceiling, wall, angle	
Robot footprint		430.5 mm x 370 mm	
Weight (excluding controller), approx.		255 kg	

Axis data /		Speed with	
Range of motion		rated payload 8 kg	
Axis 1 (A1)	+ / -185°	200°/s	
Axis 2 (A2)	+65° / -185°	175°/s	
Axis 3 (A3)	+175° / -138°	190°/s	
Axis 4 (A4)	+ / -165°	430°/s	
Axis 5 (A5)	+140° / -115°	430°/s	
Axis 6 (A6)	+ / -350°	630°/s	

(a) Especificaciones



(b) Vista del espacio de trabajo

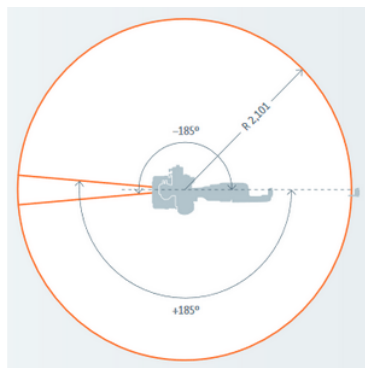
Figura 2: Especificaciones del fabricante

La tarea seleccionada es la limpieza láser que se utiliza para remover pinturas, óxido, grasas y otros materiales. Se utilizará una unidad láser QF-1000 de la marca P-Laser. Es una máquina de 1000W de potencia con manguera de 16m, la fuente del láser es fibra óptica. La alimentación de la máquina es de 380-440V, 50/60Hz. El peso de 600kg de dicha máquina no será soportado por el brazo sino que la misma estará apoyada en el suelo a una distancia prudencial de la base. El brazo fue seleccionado considerando la similitud que tiene la tarea con la de

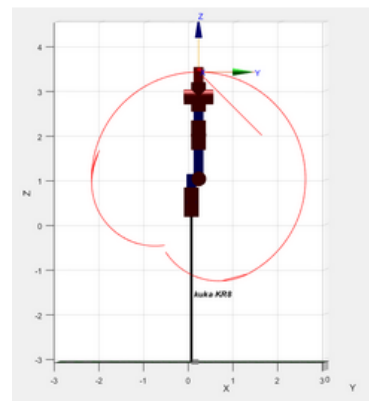


Figura 3: laser

pintura o soldadura que es la de seguir trayectorias, utilizando una herramienta de bajo peso y teniendo un buen alcance. Para lo cual KUKA promociona al KR 8 R2100-2 arc HW como una buena elección. La tarea desarrollada en este



(a) Otra vista



(b) Vista frontal

Figura 4: Vistas del espacio de trabajo

trabajo será la de limpiar cara frontal, superior y dos laterales de un cubo de

0.6m de lado. La cara frontal del mismo estará ubicada frente al robot a una distancia de 1,3 metros de la base y a 65 cm del suelo.

Si bien se puede observar que el objeto está en su totalidad dentro del espacio de trabajo, la limpieza de la cara posterior y en los extremos más alejados de las caras superior y laterales no es posible debido a que para ello el robot debería actuar en puntos singulares que lo ponen fuera de su área de trabajo.

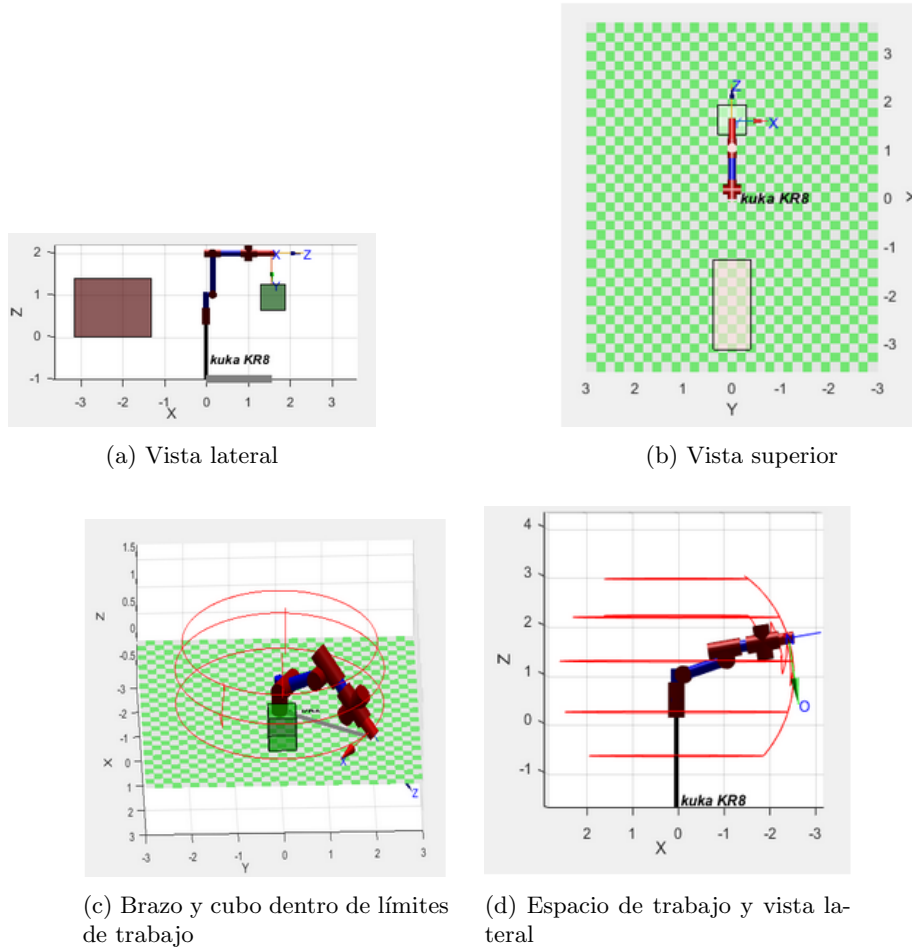


Figura 5: Vistas del espacio de trabajo

3. Denavit y Hartenberg

Basándonos en la información extraída de la hoja de datos del fabricante de la figuraX se realizó el siguiente modelo del robot para la confección de la tabla con los parámetros de denavit y hartenberg.

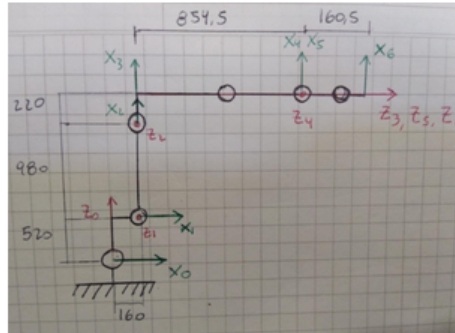


Figura 6: laser

Hay que aclarar que en una primera instancia se cometió un error en el cálculo en los parámetros de DH, no se consideró el valor de a_3 . Lo cual no impidió el desarrollo posterior del trabajo, pero sí modificó los resultados obtenidos no reflejan realmente a nuestro robot. Dado el tardío descubrimiento de este error no se pudo corregir la implementación más importante como la de cinemática inversa.

Articulación	θ_1	d_1	α_1	σ_1	
1	q_1	0,520	0,160	90°	0
2	q_2	0	0,980	0°	0
3	q_3	0	0,220	90°	0
4	q_4	0,8545	0	-90°	0
5	q_5	0	0	90°	0
6	q_6	0,1605	0	0°	0

Hay que aclarar que en una primera instancia se cometió un error en el cálculo en los parámetros de DH, no se consideró el valor de a_3 . Lo cual no impidió el desarrollo posterior del trabajo, pero sí modificó los resultados obtenidos no reflejan realmente a nuestro robot. Dado el tardío descubrimiento de este error no se pudo corregir la implementación más importante como la de cinemática inversa.

theta	Límites
q_1	$[-185^\circ, 185^\circ]$
q_2	$[-155^\circ, 95^\circ]$
q_3	$[-85^\circ, 228^\circ]$
q_4	$[-165^\circ, 165^\circ]$
q_5	$[-140^\circ, 115^\circ]$
q_6	$[-350^\circ, 350^\circ]$

Cuadro 1: Límites articulares

Con estos parámetros y considerando para que el brazo en posición inicial de home tuviera una postura adecuada se decidió adoptar los siguientes valores de offset $[0 \text{ pi}/2 \text{ } 0 \text{ pi}/2 \text{ } 0 \text{ } 0]$. Utilizando el toolbox de MATLAB se pudo generar brazo de la Figura ??.

Los límites articulares vienen dados por el fabricante como se puede ver en la figura 2 . Considerando la posición de offset que se le dio a nuestro brazo cuando se encuentra en home y dado el funcionamiento del toolbox de matlab. Los límites articulares finales se pueden ver en la Tabla 1.

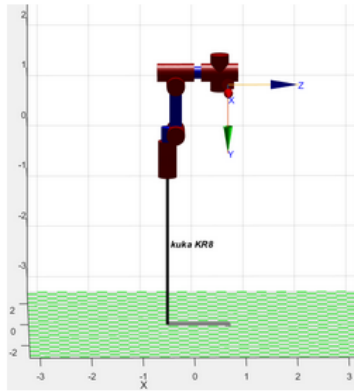


Figura 7: Brazo en posición de home

Según el tipo de limpiador (Wet, dry, etc) se usa una longitud de onda distinta. Para las distintas tareas el efector se puede colocar tanto a pequeñas distancias como a distancias de cientos de mm. En el caso elegido para este desarrollo se eligió una distancia de aplicación del limpiador de 250 mm de distancia y se consideró a la herramienta con una dimensión de 150 mm. Entonces la distancia a tener en cuenta entre el extremo del robot y el punto de aplicación del láser es de 400 mm.

Para evitar colisiones con el piso durante la tarea dada se optó por colocar al robot sobre una base elevada 500 mm al igual que la pieza a limpiar, por lo tanto al desarrollar el producto de matrices para lograr el posicionamiento y orientación del extremo del efector se debe tener en cuenta esta distancia. La representación del brazo con estas consideraciones se puede observar en la imagen 10a.

Lo anteriormente mencionado da como resultado la introducción de dos matrices homogéneas a la multiplicación de matrices que representan al sistema de nuestro brazo.

$${}^0T_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^6T_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.40 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_H = {}^0T_B \cdot {}^BT_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_H$$

4. Cinemática Directa

La función desarrollada para el cálculo de cinemática directa hace uso de parámetros de entrada que son: la posición articular deseada, la matriz de Denavit-Hatemberg y un objeto SerialLink del que se usan las matrices base, tool y los valores de offset. El parámetro que da como resultado es un booleano que indica si la posición es válida o no, la matriz de transformación Homogénea para la posición ingresada y un vector articular corregido en caso de que el introducido está fuera de los límites articulares. En el caso de que la consigna dada sea correcta devuelve un 1, en caso contrario un 0 y un mensaje de la ubicación de la posición articular más cercana.

Ingrese un vector de la forma [q0, q1, q2, q3, q4, q5, q6]
[180,0,90,0,90,300]

flag =

logical

1

T =

-0.5000	-0.8660	0	1.6745
0.8660	-0.5000	0	0
0	0	1.0000	1.3605
0	0	0	1.0000

Verificación: usando fkine

-0.5	-0.866-1.225e-16	1.675
0.866	-0.5 2.465e-32-1.393e-16	0
-6.123e-17-1.061e-16	1	1.361
0	0	1

(a) Salida sin error

Ingrese un vector de la forma [q0, q1, q2, q3, q4, q5, q6]
[180,0,90,0,90,600]
Error q1 qd1_6 valor mayor a 350 el nuevo valor será 350
flag =

logical

0

T =

-0.9848	-0.1736	0	1.6745
0.1736	-0.9848	0	0
0	0	1.0000	1.3605
0	0	0	1.0000

(b) Salida con error

Figura 8: Resultados de la función de Cinemática directa

Haciendo una verificación de los resultados obtenidos con nuestra función y lo que devuelve fkine se ve que hay una pequeña diferencia entre las matrices homogéneas debido a la aproximación en el cálculo pero es despreciable (son del orden de 1×10^{-17}).

5. Cinemática Inversa

La función de cinemática inversa desarrollada recibe como parámetros la posición articular actual, los parámetros de Denavit y Hartenberg, una matriz homogénea con la rotación y traslación deseada en el efector y un objeto del tipo SerialLink. Como resultado esta función devuelve un vector con los valores articulares necesarios para llegar a la posición objetivo y un booleano que cuando es 1 significa que la posición objetivo está dentro del espacio de trabajo caso contrario devuelve 0.

Para el cálculo de la cinemática inversa se hizo uso del método de Pieper que consiste en separar el problema en 2, analizando primero las 3 primeras articulaciones que posicionan al extremo del robot en un punto del espacio y una segunda parte donde se trabaja la orientación del extremo. Esto es posible siempre que los ejes Z de las últimas 3 articulaciones se crucen en un punto.

Las primeras 3 articulaciones fueron resueltas con método geométrico en la imagen 16 se observan los dibujos realizados en papel para los cálculos trigonométricos que fueron implementados en el código.

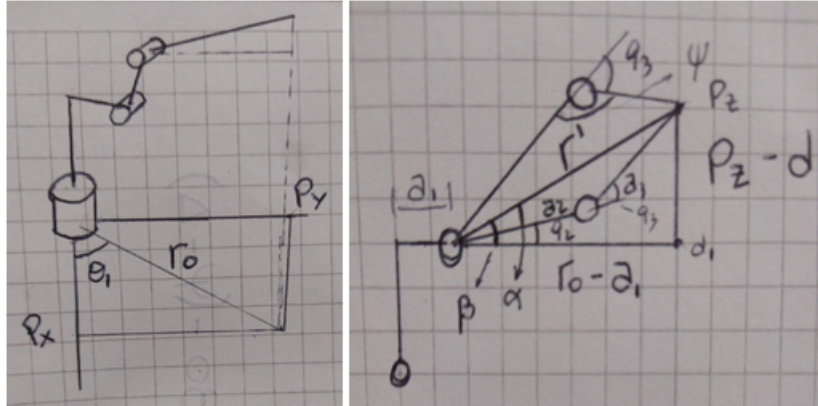
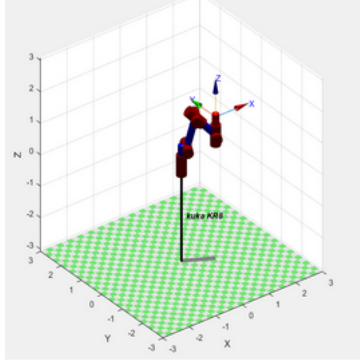


Figura 9: Modelo geométrico

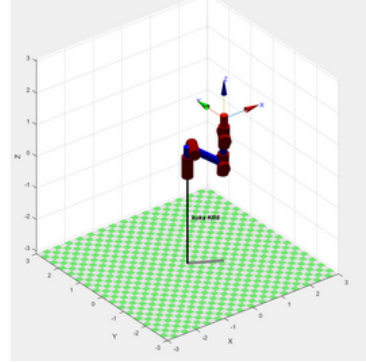
Con esta primera parte se obtuvieron 4 conjuntos de soluciones para un punto en el espacio. Las configuraciones son una para codo arriba y una codo abajo con la base orientada un ángulo q_1 y un conjunto de codo arriba y codo abajo pero con la base rotada 180° respecto a q_1 .

La parte del problema de la orientación tiene dos consideraciones cuando estamos frente a una solución degenerada que es cuando q_5 vale 0 y los ejes Z3 y Z5 se encuentran alineados. Caso contrario estaremos frente a una solución que se calculará de forma normal. El código contempla las dos posibilidades. Al igual que en la primera parte del problema tendremos dos conjuntos de soluciones equivalentes a hombro arriba y abajo para la orientación de la muñeca del robot. Por lo cual para cada una de las primeras 4 soluciones ya calculadas tendremos 2 posibles orientaciones de muñeca dando en total 8 posibles soluciones de como

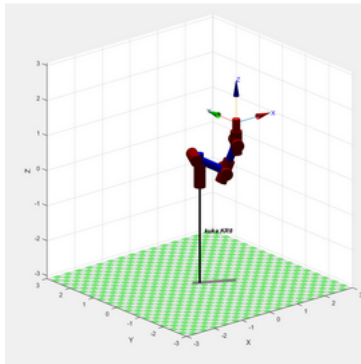
posicionar el brazo para alcanzar un punto en el espacio.



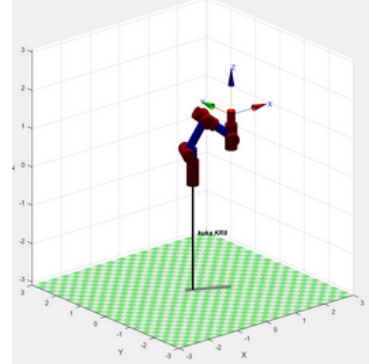
(a) Posición 1



(b) Posición 2



(c) Posición 3



(d) Posición 4

Figura 10: Distintas configuraciones

En las imágenes anteriores (Figura 10) se pueden observar los plots del brazo para 4 soluciones posibles las 4 soluciones que han quedado sin plot visualmente se verían igual porque no se podría apreciar el cambio de la rotación de las últimas 3 articulaciones.

Se puede ver que para el caso de la figura X el brazo se encuentra en una configuración que no es posible dado que colisiona consigo mismo.

6. Jacobiano

El uso del Jacobiano sirve para poder obtener las velocidades articulares en función a la velocidad angular y lineal y viceversa. También es útil para poder encontrar los puntos singulares, que son de importancia ya que al pasar el robot por estos puntos pierde como mínimo un grado de libertad y se restringe

su movimiento o la posibilidad de desarrollar velocidad en una dirección. El determinante del Jacobiano igual a 0 indica que hay un punto singular.

Para este trabajo no se ha determinado el determinante simbólico del robot porque al ser un robot de 6 grados de libertad se vuelve muy difícil la manipulabilidad del mismo, por lo tanto se optó por realizar aproximaciones numéricas a los mismos, eligiendo posiciones aleatorias y calculando el determinante en cada posición.

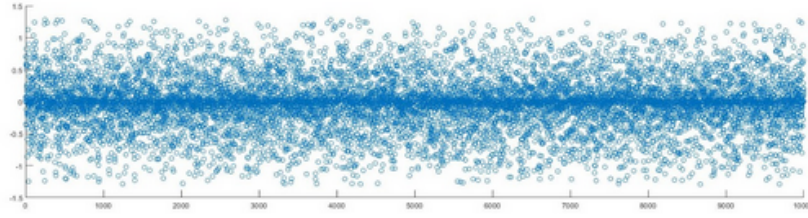


Figura 11: Modelo geométrico

En el caso particular de nuestro robot se procedió a generar 10.000 posiciones articulares de manera aleatoria con la condición de que se encontrasen dentro de los límites articulares del robot. Cabe destacar que pese a que a simple vista puede parecer que hay muchas posiciones articulares que sean singularidades, es decir que el determinante de su jacobiano sea igual a 0 (o cercano bajo ciertos límites), solamente fueron singulares aproximadamente un 1,4% de ellas. El umbral para determinar si una posición es articular o no fue fijado en 0.0009 para todos los valores absolutos del determinante, o sea que los valores menores se consideraron posiciones singulares.

También se almacenaron los valores máximo y mínimo de los determinantes, siendo el más pequeño de ellos (en valor absoluto) del orden de 1×10^{-34} y el mayor aproximadamente 1,3 (los valores de cada máximo o mínimo se imprimen en pantalla al ejecutar el programa). Tanto el número de posiciones singulares como los valores máximo y mínimo no son siempre iguales porque se obtienen de manera aleatoria pero con una población de muestreo de 10.000 el muestreo suficientemente grande para tener robustez en la obtención de datos.

Del muestreo previamente mencionado se buscaron 4 de las posiciones articulares singulares (de manera aleatoria) y se generaron las imágenes expuestas a continuación. Se pueden sacar las siguientes conclusiones: los puntos de singularidad se pueden deber a diversos factores como por ejemplo el alineamiento de dos o más articulaciones que se pueden generar por ejemplo al llegar al límite del espacio de trabajo (no necesariamente) o cuando uno de los nodos llega a su límite articular, afectando al movimiento del robot, esto quiere decir que aunque los otros nodos no hayan llegado a su límite articular y se esté dentro del espacio de trabajo puede haber casos en los que haya una singularidad para esa configuración dada. También hay casos en los que el robot mismo “vuelve sobre sí mismo” .

Cabe destacar que este estudio se limita a singularidades ya que si se quiere

hacer un análisis con mayor detalle se deben analizar los límites físicos de cada nodo (para que el robot no choque consigo mismo).

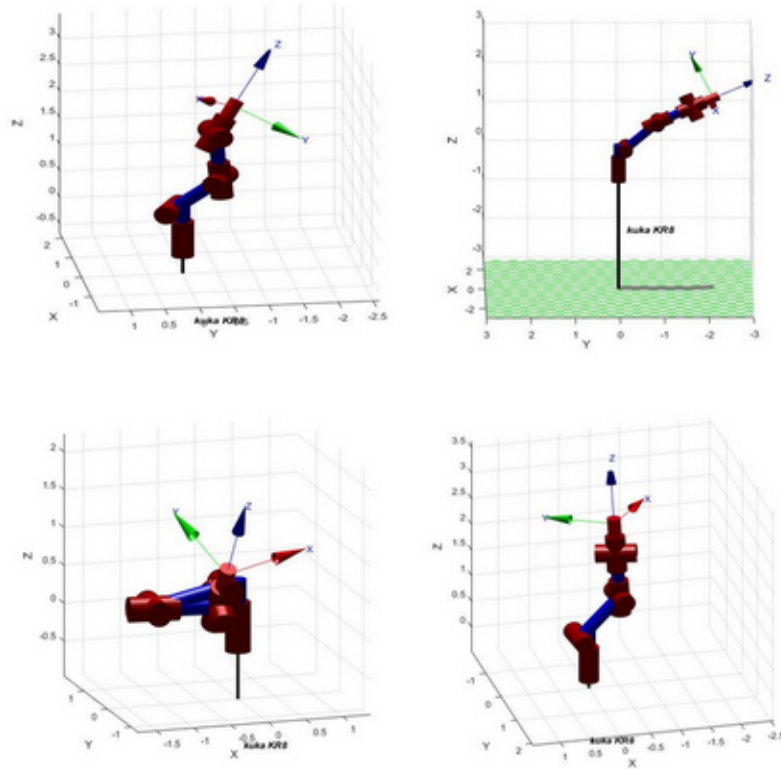


Figura 12: Modelo geométrico

Cabe destacar que este estudio se limita a singularidades ya que si se quiere hacer un análisis con mayor detalle se deben analizar los límites físicos de cada nodo (para que el robot no choque consigo mismo).

A partir de las conclusiones anteriores se puede inferir que las posiciones de trabajo del robot no deben estar cerca del espacio de trabajo ni muy cerca de la base del robot porque hay puntos singulares y se restringe el movimiento. Además se debe considerar estos puntos ya que en los mismos para llegar a determinadas velocidades finitas del extremo pueden corresponderle velocidades articulares infinitas. Las articulaciones que pueden llegar a estar más comprometidas cuando se trabaja en puntos cercanos a la singularidad son la tercera y cuarta ya que en la dirección del eje Z de las mismas no es posible un desplazamiento más allá del que le puede dar la rotación del quinto grado de libertad. También se puede hacer un análisis mediante los elipsoides de manipulabilidad, se analizará en una posición típica de trabajo y qué sucede cuando el extremo se acerca al límite del espacio de trabajo.

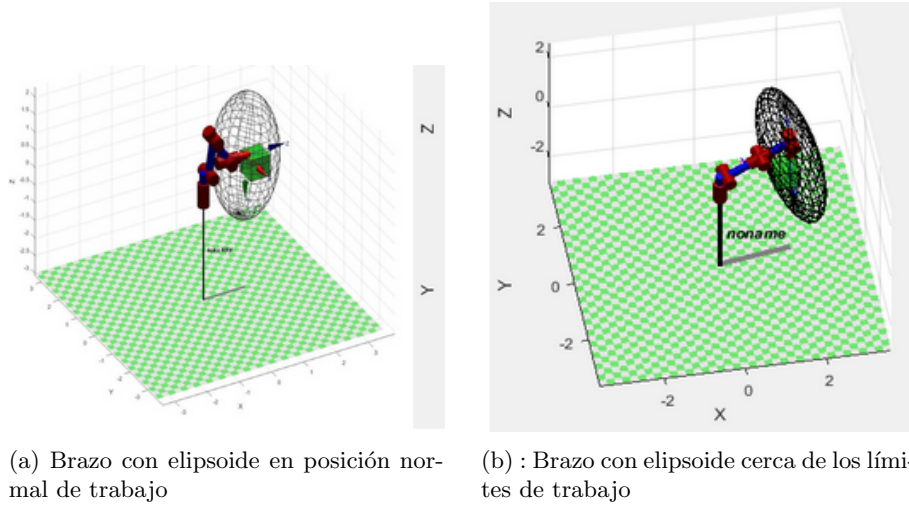


Figura 13: Elipsoides de manipulación

En la Figura 13a (posición típica de trabajo) el brazo tendrá mayores posibilidades de moverse en la dirección del eje Z. En direcciones X e Y las posibilidades son menores pero siguen existiendo con un margen de comodidad. En la imagen 24 se puede observar que el elipsoide tiene un eje con un eje de menor tamaño (aproximadamente en dirección X) lo que indica que tiene menos posibilidades de moverse en esa dirección.

7. Trayectorias

Para la generación de trayectorias se hizo uso de las funciones creadas anteriormente para cinemática directa e inversa. Se procedió a calcular la trayectoria necesaria para limpiar la cara frontal del cubo (la que está frente al brazo), la mitad de la cara superior y de sus caras laterales.

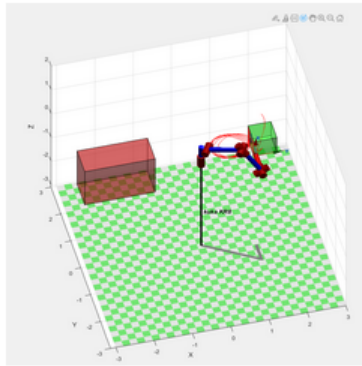
La operación de limpieza láser requiere que el efector siga una trayectoria suave a velocidad constante y en todo momento mantenga la misma distancia. Se consideró que el ancho del haz de láser es de 10 centímetros y está orientado a lo largo del eje Y del efector, el haz impacta a las superficies en dirección del eje Z. Por lo tanto para limpiar una cara del cubo el efector realiza recorridos en sentido del eje X y se mueve de a 10 centímetros en el eje Y.

Se plantearon dos soluciones posibles, una realizando interpolación en el espacio articular con la herramienta *jttraj* y otra en el espacio cartesiano con *ctrtraj* (RTB, Peter Corke). Estas herramientas hacen uso de distintos tipos de ajuste; *ctrtraj* hace uso de un ajuste del perfil de velocidad trapezoidal y *jttraj* un ajuste polinomial de quinto orden.

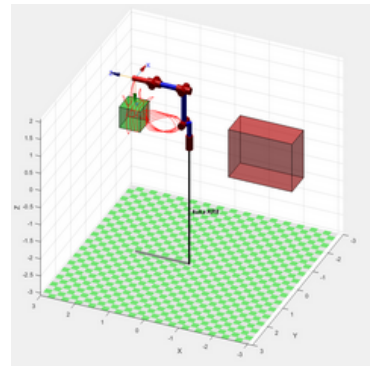
Lo primero fue definir la trayectoria que recorre el efector para lo cual se consideraron 40 puntos de la misma creando matrices de transformación que

se ingresan como parámetro a la función de cinemática inversa mencionada anteriormente. El brazo parte de home, realiza una maniobra de aproximación y luego se ubica en un vértice de la cara frontal del cubo y comienza a recorrerla. Cada vez que termina una cara realiza una maniobra de seguridad para no colisionar con la pieza ni con el brazo mismo y finaliza la limpieza volviendo al home.

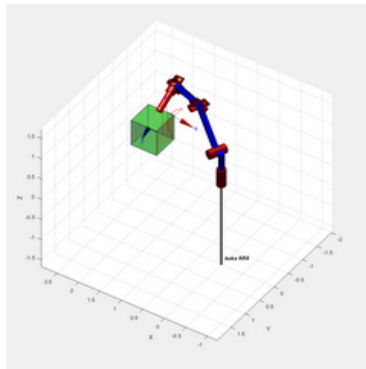
Las maniobras de seguridad no fueron consideradas en una primera instancia lo cual generaba que en la trayectoria el efector colisionara con el cubo o al querer orientar la herramienta diera vueltas innecesarias. Al agregar los puntos intermedios de maniobras de seguridad se realizan pequeños movimientos y rotaciones para que la interpolación no genere tantos movimientos no deseados. Para jtraj se transformaron las coordenadas cartesianas al espacio articular me-



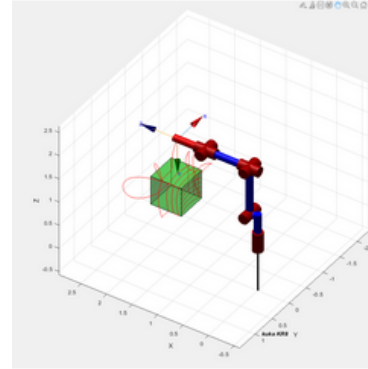
(a) Trayectoria Jtraj con problemas de camino



(b) Trayectoria Jtraj con problemas de camino



(c) Trayectoria Jtraj sin problemas de camino



(d) Trayectoria Jtraj sin problemas de camino

Figura 14: Comparación de Jtraj

diante nuestra función de cinemática inversa. Introduciendola en jtraj se obtiene

una matriz que en cada fila posee un vector posición del espacio articular, con esta matriz se indica el movimiento a seguir desde la primer fila como punto inicial hasta la última fila como punto final; junto con las posiciones articulares, la función permite obtener también la velocidad y la aceleración para cada articulación.

En una primera instancia la trayectoria generada no era satisfactoria debido a que durante la limpieza de la cara frontal se producía un movimiento no deseado de bucle como se observa en las imágenes 14a y 14b mientras que el resto de las caras las recorría sin ningún problema. Este error era producido por la misma interpolación de `jtraj`. Primero se propuso agregar mayor cantidad de puntos en el camino para la limpieza de la cara frontal pero esto no corrigió el problema. La solución se encontró cuando se colocaron puntos intermedios entre la posición de home y la primera arista del cubo. La nueva trayectoria obtenida se puede observar en las imágenes 14c y 14d la cual cumplió satisfactoriamente el recorrido propuesto sin colisiones.

En la segunda alternativa se usa la función `ctrj` que al interpolar en el espacio cartesiano devuelve un arreglo de matrices homogéneas la cual describe la serie de puntos del espacio cartesiano que tiene que recorrer el efector para llegar al objetivo. Luego mediante el empleo de nuestra función de cinemática inversa se transforma al espacio articular. En este caso el camino realizado cumple con la tarea satisfactoriamente como se puede ver en las imágenes 14c y 14d. Con la función `diff` de MATLAB se calculan las derivadas sucesivas para obtener la velocidad y aceleración de cada articulación.

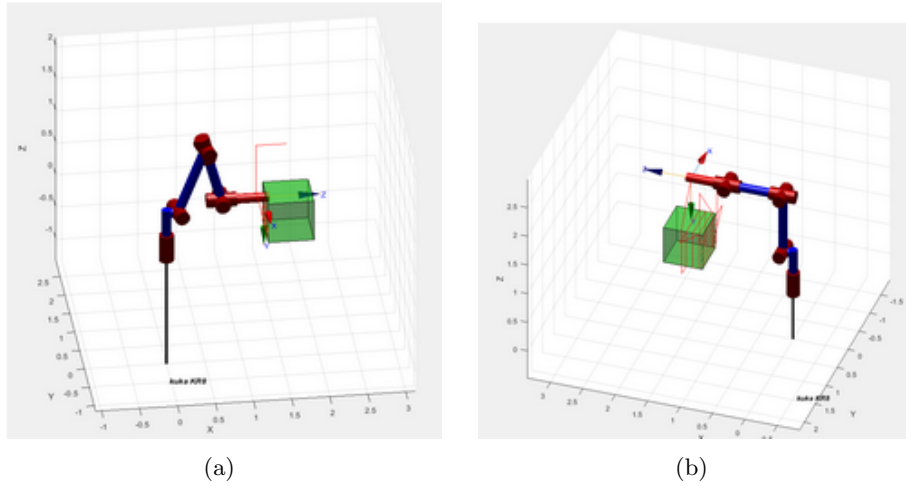


Figura 15: Trayectoria `ctrj` sin problemas

Luego de graficar las trayectorias para cada método se graficaron las aceleraciones, velocidades y posiciones articulares para cada uno.

En la imagen 16 se puede observar claramente cómo se realiza una interpolación con polinomio de orden 5 dado que las gráficas de velocidad y aceleración

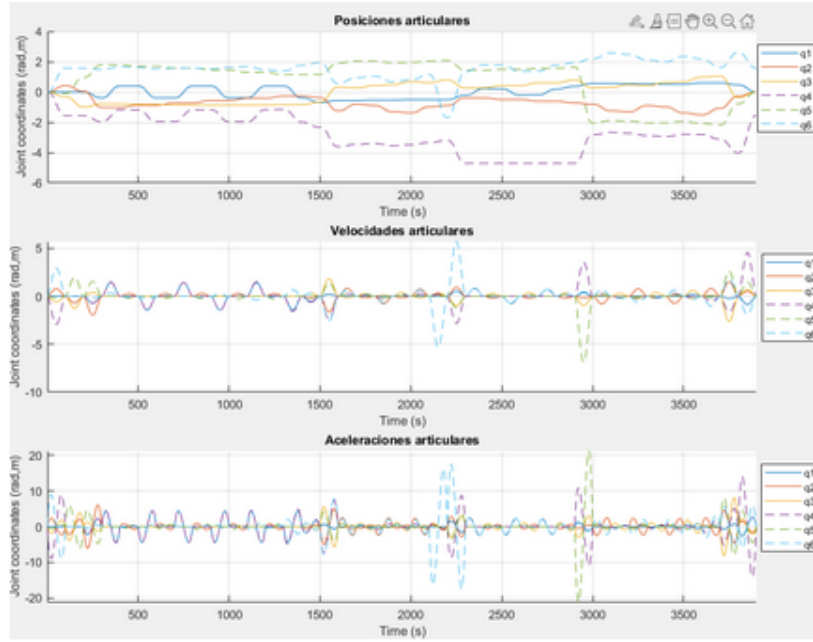


Figura 16: Posición, velocidad y aceleración para jtraj

tienen formas suaves. El comportamiento del brazo es aceptable y los valores de velocidad y aceleración están dentro de lo tolerable por nuestro brazo.

En la imagen 17 es visible que ctraj genera perfiles más cuadrados en las 3 gráficas esto se debe a la interpolación trapezoidal. Hay que señalar que en la posición se generan cambios bruscos, esto provoca picos de velocidad y aceleración, estos impulsos claramente están fuera de los rangos límites de velocidad como se puede ver en la imagen 33. Lo mencionado anteriormente implica que no se puedan llevar a cabo dado la exigencia de esa maniobra. Aunque se pudieran disminuir los valores de esos picos los cambios siguen siendo bruscos lo cual haría que nuestro robot sintiera los efectos de golpe del jerk que es la derivada de la aceleración afectando la vida útil de las piezas de nuestro brazo.

Haciendo un análisis solo en la trayectoria la mejor implementación para nuestra tarea es la de ctraj dado que sigue líneas rectas lo cual es requerido para que la limpieza sea precisa y se requiere que el láser se mantenga a una distancia constante. Con el uso de jtraj las trayectorias realizan pequeñas curvas acercando la distancia del efector y no limpiando en línea recta.

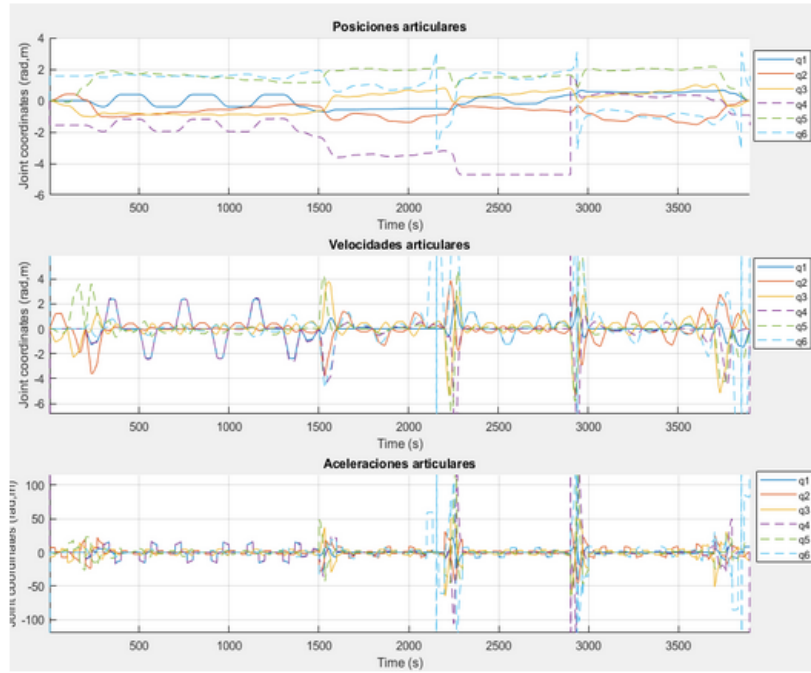


Figura 17: Posición, velocidad y aceleración para ctraj

Articulacion	Velocidad [$\frac{rad}{s}$]
q_1	3,5
q_2	3,05
q_3	3,32
q_4	7,5
q_5	7,5
q_6	11

Figura 18: Posición, velocidad y aceleración para ctraj

8. Sensores y actuadores

Para este desarrollo se plantearon sensores y actuadores genéricos con su debida alimentación, con algún ejemplo para cada uno. Como el caso de estudio es un robot industrial que ya cuenta con casi todos los sensores necesarios para el correcto funcionamiento del brazo, no hace falta especificar sensores internos, la única diferencia que se puede hacer es agregar sensores externos para la aplicación dada pero se desarrollarán distintos sensores internos y externos que podría llevar un robot similar. Como caso particular, para la tarea definida se

necesitarán sensores para el correcto funcionamiento del efector como lo son un sensor de posición y una posible implementación de un sensor de polvo.

8.1. Sensores de posición

Para cada articulación es necesario sensores de posición exacta y velocidad (con la aplicación de un sistema de control adecuado se podría estimar la velocidad en función del sensor de posición). Para el posicionamiento se pueden elegir sensores tanto ópticos como inductivos (“incoders”). Hay una pequeña diferencia en cuanto al precio pero se compensa con una mejor precisión al usar encoders inductivos. Como ejemplo de uso se propone **Zettlex Incoder Midi Ultra Range 75-300mm (CELERA)**. Dicho sensor puede elegirse para recibir alimentación de 5, 12 o 24 V y se elige para este caso el que soporta tensiones nominales de 24 V.



Figura 19: Incoder

8.2. Sensor de proximidad para el efector y para el brazo

se plantearon diferentes tipos de sensores, como sensores infrarrojos o inductivos, pero a fin de evitar inconvenientes al limpiar materiales ferromagnéticos por ejemplo se optaron por sensores de ultrasonido. Como caso particular se podría implementar el **Sick UM12 / UM12 UM12-1172271**. El sensor especificado puede tener límites de hasta 15 V o 30 V, se elige el sensor que soporta hasta 30 V. Con el sensor se buscará que el efector nunca se acerque a mas de 15cm de la pieza a limpiar.

8.3. Medición de polvo

La herramienta usada para la limpieza incluye la alimentación del láser y de una aspiradora que se usa para limpiar el polvo que puede estar en suspensión. De todas formas se podría usar un sensor para medir el nivel de polvo en el aire



Figura 20: Sensor de proximidad

para asegurar un ambiente seguro. Se propone usar el Sistema de medición de polvo por transmisión DUSTHUNTER T200.



Figura 21: Sensor de polvo

Para el sensor de proximidad se necesita una alimentación con valor nominal de 15 a 30 V: se propone una de 24 V. Para los encoders de posición se propone el de 24 V. Como consecuencia de lo mencionado anteriormente, se propone una fuente de 24 V.

8.4. Motor

DSH1 45–100 High Precision: Se plantea este servomotor síncrono monofásico porque es un motor de precisión y tiene buenas prestaciones de potencia para un tamaño reducido. Estos motores se pueden conectar a 220 V. Los motores deben llevar reductores de velocidad. Para el caso de la elección de un motor DSH1-100 por ejemplo, se necesitaría un reductor de 1:103 de relación para obtener resultados similares a los de las velocidades máximas de cada articulación ($3000 \text{ rpm} = 18000^\circ/\text{s}$ y se necesita 175° para una de las articulaciones). La alimentación del efector (y de la aspiradora) se realiza directamente desde la máquina de limpieza que está conectada a la red eléctrica.

Adicionalmente se podría usar un sensor de corriente como sensor interno.

Type	P_h		n_h		J		M_h		M_{max}	
	[kW]	[hp]	[min ⁻¹]		[kgcm ²]	[lb in ²]	[Nm]	[lbf ft]	[Nm]	[lbf ft]
DSH1-045	0.5-1.3	0.68-1.7	1000-4000		1.3-3.0	0.44-1.0	2.5-6.1	1.8-4.5	7.8-23	5.8-17
DSH1-056	0.6-2.6	0.81-3.6	1000-4000		4.2-11	1.4-3.7	5.7-13	4.2-9.6	15-44	11-32
DSH1-071	1.1-5.7	1.5-7.6	1000-4000		13-30	4.3-10	11-27	7.9-20	27-78	20-57
DSH1-100	2.1-8.2	2.8-11	1000-3000		41-95	14-32	21-57	15-42	40-125	29-92

Figura 22: Tabla de motor

9. Conclusiones

Con el trabajo realizado en este informe se pudo llevar a cabo satisfactoriamente la implementación de cinemática directa, cinemática inversa y estudio de singularidades con la finalidad de simular una trayectoria para el robot elegido y la tarea de limpieza propuesta a partir del modelado y análisis de los parámetros de Denavit- Hartenberg.

Las trayectorias desarrolladas cumplen desde el aspecto cinemático de la tarea pero requieren un estudio mucho más profundo para su aplicación en la realidad debido a que la generada con ctraj se adapta mejor a nuestras exigencias pero las velocidades desarrolladas están fuera de los límites funcionales de los motores. Posibles soluciones serían implementar más puntos en la trayectoria evitando cambios bruscos de posición o usar una trayectoria con jtraj y ctraj por tramos.

Como anteriormente se mencionó se cometió un error en la implementación de Denavit y Hartenberg que implicó trabajar con un robot que no era físicamente como el propuesto, si bien no implicó que no se pudiera desarrollar el trabajo el mayor cambio fue en el cálculo de la cinemática inversa. Si bien se desarrollaron posibles soluciones no se logró que se cumpla la consigna satisfactoriamente a pesar de haber encontrado el error y una posible implementación trigonométrica en el cálculo de cinemática inversa. Se deja esta implementación como deuda técnica y posterior corrección.