

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y
AUTOMÁTICA INDUSTRIAL



"DISEÑO DE UN ROBOT PARALELO
CON ARTICULACIONES
HELICOIDALES"

TRABAJO FIN DE GRADO

Septiembre - 2020

AUTORA: Manuela Agudelo Romero

DIRECTOR: Adrián Peidró Vidal

“Respetar tus esfuerzos, respetarte a ti mismo.

El respeto propio lleva a la autodisciplina.

Cuando tienes ambos bajo tu cinturón,

ese es el verdadero poder”

Clint Eastwood





Agradecimientos

En primer lugar, a mi tutor académico Adrián Peidró Vidal por haberme ofrecido un trabajo que me permitiera aprender y crecer como persona. Indudablemente, sin su ayuda, consejos y ánimos, que me han acompañado durante todo el desarrollo de este trabajo, esta experiencia final del grado no hubiera sido tan agradable.

También a mi madre, quien me ha cuidado, comprendido y motivado en todo momento. El permitirme pasar menos tiempo a su lado y su esfuerzo y trabajo extras, me han dado la posibilidad de dedicarle más tiempo a este trabajo.

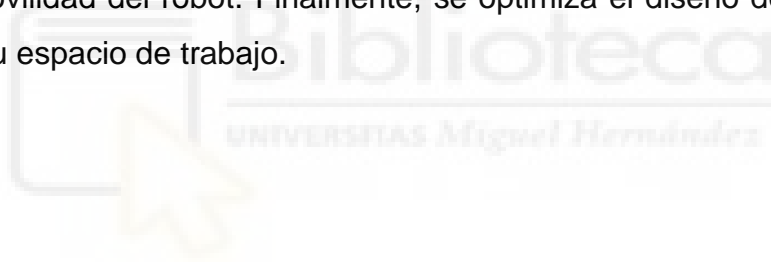
Asimismo, a mi pareja; quien también me ha mimado, atendido y colaborado durante todo este tiempo para que pudiera aprender y mejorar más. Su apoyo y ánimos han sido fundamentales.

Y finalmente, a mis amigos y compañeros del grado. En especial y decididamente, a Pilar Pardo Millá, quien además de animarme y aconsejarme, me ha asesorado según su experiencia y conocimientos.



Resumen

En este trabajo se diseña un robot paralelo de cuatro grados de libertad que genera movimientos de traslación en tres ejes y de rotación en un eje para el efector final (movimiento de Schönflies). Se utilizan articulaciones de tipo rotacional y prismática, pero la principal novedad es el uso de las de tipo helicoidal para producir operaciones de *pick and place* (recoger y colocar). De este modo, la configuración empleada es RPHH para cada uno de los dos brazos que conforman el robot. La cinemática directa e inversa se resuelven de forma sencilla a partir del análisis de la geometría del robot. También se analiza el espacio de trabajo según las variables geométricas que condicionan el diseño y se determinan las configuraciones singulares del robot mediante el cálculo de la matriz Jacobiana. Para visualizar el movimiento del robot, se realiza su simulación con la herramienta EJS. Tras analizar la influencia de cada parámetro geométrico, se eligen las medidas para construir un prototipo experimental y así validar la movilidad del robot. Finalmente, se optimiza el diseño del robot para maximizar su espacio de trabajo.





Abstract

In this work, a parallel robot with four degrees of freedom that generates translation movements in three axes and rotation in one axis for the end effector (Schöenflies movement) is designed. Rotational and prismatic type joints are used, but the main novelty is the use of the helical type to produce pick and place operations. In this way, the configuration used is RPHH for each of the two arms that make up the robot. Forward and inverse kinematics are easily solved from the analysis of the robot's geometry. The workspace is also analyzed according to the geometric variables that condition the design and the singular configurations of the robot are determined by calculating the Jacobian matrix. To visualize the movement of the robot, its simulation is made with the EJS tool. After analyzing the influence of each geometric parameter, the measurements are chosen to build an experimental prototype and thus validate the mobility of the robot. Finally, the design of the robot is optimized to maximize its workspace.





ÍNDICE GENERAL

Agradecimientos	V
Resumen	VII
Abstract.....	IX
Índice de figuras.....	XIII
Índice de tablas	XVIII
1. INTRODUCCIÓN	21
1.1. OTROS ROBOTS HELICOIDALES	21
1.2. OBJETIVO DEL TRABAJO DE FIN DE GRADO	23
1.3. ESTRUCTURA DE LA MEMORIA	23
2. ANALISIS DEL ROBOT PROPUESTO	25
2.1. DESCRIPCIÓN	25
2.2. CINEMÁTICA DIRECTA	28
2.3. CINEMÁTICA INVERSA	32
2.4. ESPACIO DE TRABAJO.....	36
2.5. SINGULARIDADES.....	38
2.6. SIMULACIÓN CON EJS	44
3. CONSTRUCCIÓN DE PROTOTIPO CON IMPRESIÓN 3D	48
3.1. TECNOLOGÍAS Y MATERIALES UTILIZADOS	48
3.2. MEDIDAS INICIALES.....	49
3.3. DESARROLLO.....	54
3.3.1. MODELADO 3D Y MONTAJE	54
3.3.2. IMPRESIÓN 3D.....	69
3.4. PROTOTIPO FINAL.....	79
3.5. FUNCIONAMIENTO.....	80
4. OPTIMIZACIÓN DEL DISEÑO	82

4.1. MAXIMIZACIÓN DEL ESPACIO DE TRABAJO 2D	88
4.2. MAXIMIZACIÓN DEL ESPACIO DE TRABAJO 4D	90
5. CONCLUSIONES	103
5.1. TRABAJOS FUTUROS	104
Bibliografía.....	105
Anexos	107
Anexo 1. Código implementado en EJS.....	108
Anexo 2. Análisis de sensibilidad.....	115
Anexo 3. Código implementado en Matlab	138



Índice de figuras

Figura 2.1: Alzado del robot	25
Figura 2.2: Planta del robot	25
Figura 2.3: Codo arriba del robot.....	29
Figura 2.4: Codo abajo del robot	29
Figura 2.5: Posición codo arriba.....	30
Figura 2.6: Posición codo abajo	30
Figura 2.7: Representación del paso de una rosca	31
Figura 2.8: Configuraciones solución para la cinemática inversa.....	35
Figura 2.9: Representación aproximada del funcionamiento del Algoritmo 2 en la nube de puntos.....	43
Figura 2.10: Ventana 3D de la interfaz creada con EJS.....	45
Figura 2.11: Ventana Planta de la interfaz creada con EJS	46
Figura 2.12: Ejemplos de configuraciones singulares del robot	47
Figura 3.1: Ventana 3D de EJS con los valores del diseño inicial para el análisis de sensibilidad.....	50
Figura 3.2: Ventana Planta de EJS con el espacio de trabajo generado por los valores del diseño inicial para el análisis de sensibilidad donde z y φ son constantes	51
Figura 3.3: Espacio de trabajo alcanzable cuando $d = 0$ y el resto de parámetros tienen los valores del diseño inicial	53
Figura 3.4: Visualización de la trayectoria de la hélice cilíndrica restada al cilindro para generar una articulación helicoidal.....	55
Figura 3.5: Comparativa entre la impresión de los cilindros usando una forma cuadrada (izquierda) o trapezoidal (derecha) para la trayectoria de la hélice cilíndrica	55
Figura 3.6: Visualización de los cilindros de cada articulación helicoidal	56

Figura 3.7: Representación de la trayectoria de la hélice cilíndrica y del resultado de unir ésta con la tuerca base	56
Figura 3.8: Ensamblaje de las articulaciones helicoidales y las varillas pasivas b_1 y b_2	57
Figura 3.9: Partes que conforman el modelo 3D de un actuador lineal	58
Figura 3.10: Transición de la pieza aux_act_cil.....	58
Figura 3.11: Representación de la coaxialidad entre el cilindro de la pieza aux_act_cil y el orificio del cilindro de la articulación helicoidal.....	59
Figura 3.12: Pieza aux_act_cil ensamblada con el actuador lineal y el cilindro de la articulación helicoidal	59
Figura 3.13: Transición de la pieza aux_lin_rot.....	59
Figura 3.14: aux_lin_rot original con insuficiente espacio para el tornillo de ajuste del eje del motor.....	60
Figura 3.15: aux_lin_rot con más espacio para el tornillo de ajuste del eje del motor	60
Figura 3.16: aux_lin_rot modificada para la inserción del hexágono metálico como soporte del eje del motor de corriente continua.....	61
Figura 3.17: Transición de los cilindros 3 y 4 del montaje ideal (izquierda) al fallo por movimiento del cilindro 4 (derecha)	62
Figura 3.18: Montaje de los cilindros 3 y 4 mediante la pieza aux_central para impedir el movimiento entre cilindros	62
Figura 3.19: Representación de una colisión entre actuadores lineales	63
Figura 3.20: Configuración solución para las colisiones de la configuración inicial	63
Figura 3.21: Diferencia entre el prototipo idealmente diseñado y el real.....	64
Figura 3.22: Posición en el prototipo de la pieza abrazadera.....	64
Figura 3.23: Prototipo ensamblado y equilibrado	65

Figura 3.24: Diferencia entre la posición del plug ideal (izquierda) y real (derecha)	65
Figura 3.25: Diferencia entre la posición correcta del prototipo (izquierda) y la posición real (derecha).....	66
Figura 3.26: Montaje de las piezas aux_origen y aux_hex para enlazar los actuadores.....	67
Figura 3.27: Montaje de las piezas aux_t y aux_mitad.....	67
Figura 3.28: Prototipo final ensamblado	68
Figura 3.29: Impresiones fallidas al imprimir un cilindro de las articulaciones helicoidales.....	69
Figura 3.30: Diseño con cuadrados y borde exterior impreso en la zona central de la base para una correcta nivelación.....	70
Figura 3.31: Comparativa entre cuadrados según la cota de la boquilla a la placa	71
Figura 3.32: Impresión de la tuerca de una articulación helicoidal después de la calibración	71
Figura 3.33: Impresiones de una tuerca y la pieza aux_act_cil tras variar los parámetros de temperatura, velocidad, relleno, diámetro de la boquilla y altura de capa	74
Figura 3.34: Diseño de impresión para la prueba del cubo	75
Figura 3.35: Comprobación de los grosores antes (izquierda) y después del ajuste del flujo de material a extruir (derecha).....	75
Figura 3.36: Visualización de las costuras en las piezas (rojo) y la posición nueva de las costuras (azul) tras la configuración de la alineación de costuras en Z.	76
Figura 3.37: Impresión del cilindro de una articulación helicoidal tras modificar los parámetros del flujo y las costuras	76
Figura 3.38: Pieza final en la que los hilos son de un diámetro muy pequeño y fácilmente eliminables de las piezas	78

Figura 3.39: Acabado superficial de una tuerca después de realizar la calibración y de variar todos los parámetros explicados	78
Figura 3.40: Prototipo físico ensamblado	79
Figura 3.41: Prototipo físico ensamblado desde una vista más cercana.....	80
Figura 4.1: Comparativa entre los valores de los pasos de las articulaciones helicoidales mínimo, el utilizado para el prototipo construido y el máximo.....	84
Figura 4.2: Valores mínimos y máximos de z_{01} y z_{02} tomando como origen el punto verde de acuerdo con el esquema de la Figura 2.1	85
Figura 4.3: Valores mínimos de c_{10} y c_{20} respecto del efector final (punto C rojo)	86
Figura 4.4: Valores máximos de c_{10} y c_{20} respecto del efector final (punto C rojo)	86
Figura 4.5: Representación de la distancia d	87
Figura 4.6: Nube de puntos del espacio de trabajo 3D (XYZ) empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2	94
Figura 4.7: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2.....	94
Figura 4.8: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2.....	95
Figura 4.9: Valores absolutos mínimo y máximo para los pasos de las articulaciones helicoidales en.....	96
Figura 4.10: Robot optimizado final.....	99
Figura 4.11: Nube de puntos del espacio de trabajo 3D (XYZ) empleando una malla de $n=17$ puntos	100
Figura 4.12: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos	100
Figura 4.13: Nube de puntos del espacio de trabajo en el plano YZ empleando una malla de $n=17$ puntos	101

Figura 4.14: Ejemplo de configuración que consigue uno de los espacios de trabajo máximos con valores aproximados entre 0.47 m y 0.27 m..... 102

Figura 4.15: Ejemplo de configuración que consigue uno de los espacios de trabajo mínimos con valores aproximados entre 0.27 m y 0.08 m 102



Índice de tablas

Tabla 3.1: Valores de los parámetros recomendados para la impresora CR-10s	77
Tabla 4.1: Valores elegidos para los vectores <i>minimos</i> y <i>maximos</i>	83
Tabla 4.2: Valores elegidos de los diseños iniciales para la optimización del espacio de trabajo 2D y 4D	88
Tabla 4.3: Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 2D	89
Tabla 4.4: Puntos y porcentaje de malla alcanzada empleando $n = 201$ puntos en la discretización de cada eje X e Y cuando $z = \varphi = 0$	90
Tabla 4.5: Análisis 1. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D en un primer intento	91
Tabla 4.6: Puntos y porcentaje de malla alcanzada en un primer intento empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 1.....	91
Tabla 4.7: Ejecuciones necesarias para el estudio de un único diseño inicial .	92
Tabla 4.8: Valores actualizados de los vectores <i>minimos</i> y <i>maximos</i> para evitar valores no válidos de las articulaciones helicoidales 3 y 4 en la ejecución del algoritmo de optimización.....	92
Tabla 4.9: Análisis 2. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D tras reajustar los valores absolutos mínimos de λ_3 y λ_4	93
Tabla 4.10: Puntos y porcentaje de malla alcanzada empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 2.....	93
Tabla 4.11: Puntos y porcentaje de malla alcanzada empleando $n=17$ puntos en la discretización de cada eje X, Y, Z y PHI con los diseños óptimos a los que se llega con la optimización del espacio 2D cuando $z = \varphi = 0$	95

Tabla 4.12: Valores actualizados de los vectores *minimos* y *maximos* para asegurar la existencia de las articulaciones helicoidales y que no degeneren en una rotación pura o una traslación pura 97

Tabla 4.13: Análisis 3. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D tras ajustar adecuadamente los valores de los pasos para las articulaciones helicoidales 97

Tabla 4.14: Puntos y porcentaje de malla alcanzada empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 3..... 97





1. INTRODUCCIÓN

Existen múltiples topologías de robots paralelos, por lo que el campo de aplicación de este tipo de robots es muy amplio. Sirve en el sector industrial en operaciones como las de *pick and place* (recoger y colocar) o en aplicaciones en centros de mecanizado; pero también destacan en áreas de la robótica de servicio, como el desarrollo de dispositivos de rehabilitación y diagnóstico o la cirugía robótica [1]. Las elevadas velocidades de movimiento, la alta precisión de posicionamiento o la gran capacidad de carga que pueden ejercer sus efectores, los hace ideales para este tipo de aplicaciones en contraposición con los robots serie [2].

Por otro lado, dentro de las posibles articulaciones a usar en un robot, las de tipo helicoidal son aquellas que constituyen una combinación entre un movimiento rotacional y otro lineal a lo largo de un eje articular y que proporcionan un grado de libertad [3]. Este tipo de articulaciones son raramente empleadas y, además, se suelen utilizar para variar la altura en el eje Z del efector final y su orientación. Sin embargo, la novedad del robot paralelo diseñado en este documento, en particular para aplicaciones de *pick and place* (recoger y colocar), sí emplea estas articulaciones para el movimiento completo del robot, tanto para la traslación del efector final en los tres ejes XYZ como para la orientación del mismo.

1.1. OTROS ROBOTS HELICOIDALES

Como se ha comentado, dado que las articulaciones helicoidales no son empleadas habitualmente, solamente se han encontrado unos pocos robots con este tipo de articulaciones.

El primero de ellos se aplica en la cirugía laparoscópica y está pensado para sostener firmemente y dar orientación a la cámara utilizada en la laparoscopia [4]. De este modo, se reduce la carga de trabajo a los cirujanos, llegando incluso a poder suprimir la necesidad de un auxiliar quirúrgico que realice esta tarea. Se trata de un robot paralelo de seis grados de libertad que tiene una capacidad de orientación mucho mayor que otros robots implementados con

el mismo propósito. Esto es debido al uso de dos mecanismos planares compuestos por una combinación de partes circulares y triangulares, en lugar de una combinación con partes triangulares únicamente, como emplean otros robots similares. El enlace entre estos dos mecanismos se realiza mediante una varilla, lisa en un extremo, para generar una articulación prismática; y roscada por el otro, para producir un movimiento helicoidal.

Otro robot paralelo encontrado se inspira en la conocida plataforma de Gough-Stewart ([5], [6]), que utiliza articulaciones universales, esféricas y prismáticas. En este caso, el denominado robot 6-RRRPRR [7] de seis grados de libertad, emplea articulaciones universales ortogonales sin intersección y también articulaciones activas de tipo helicoidal para generar el mismo movimiento que el robot paralelo original. Esta configuración hace necesario un método específico para el modelado y resolución de la cinemática del robot, cuya eficacia consigue obtener un buen rendimiento del robot y una alta precisión en el rango de micras. En este robot, las articulaciones helicoidales sí se emplean para variar la altura y orientación del efector final.

Un tercer robot paralelo que también está diseñado a partir de la estructura de un robot anterior, la del Delta [8], es el robot de cuatro grados de libertad Heli4 [9]. Éste supera las limitaciones del robot Delta gracias a su simetría y a la robustez de la placa móvil utilizada. Además, su geometría ha sido especialmente estudiada para asegurar la ausencia de las singularidades, sobre todo de aquellas que son menos comunes y que aparecen por las articulaciones esféricas utilizadas en los paralelogramos de los brazos del robot. En este caso, la articulación helicoidal también se utiliza concretamente para variar la altura y orientación del efector final, que es el uso más habitual.

Otro robot que también utiliza la articulación helicoidal con esta misma finalidad es el denominado Peppermill [10]. Este robot paralelo de cuatro grados de libertad utiliza la topología CRRH para cada uno de sus dos brazos y emplea el movimiento de Schönflies [11], que permite trasladar el efector final en el espacio cartesiano XYZ y realizar una rotación. Otro robot paralelo

con similares rasgos al anterior, con cuatro grados de libertad y mismo movimiento de Schönflies, es el que utiliza la topología CRPH en sus dos brazos [12]. La novedad de estos dos robots (2-CRRH y 2-CRPH) radica en su simplicidad, ya que consta únicamente de dos brazos y esto hace posible la formación de una cadena cinemática cerrada de un solo bucle. Utiliza un número menor de brazos que los necesarios para los mecanismos generales de cuatro grados de libertad con movimiento de Schönflies, como los robots Delta [8] o Heli4 [9], que necesitan cuatro brazos para generar cuatro grados de libertad. Además, la orientación del efector final generada por los movimientos diferenciales de los pares helicoidales coaxiales con pasos distintos da como resultado una estructura más compacta que los mecanismos convencionales con movimiento de Schönflies. Las características de estos dos robots, resultantes del estudio de la cinemática, espacio de trabajo y singularidades de cada uno, los hacen idóneos para operaciones rápidas de *pick and place* (recoger y colocar).

1.2. OBJETIVO DEL TRABAJO DE FIN DE GRADO

El objetivo del trabajo es diseñar y modelar matemáticamente un robot paralelo de cuatro grados de libertad que emplee, en especial, articulaciones de tipo helicoidal para generar el movimiento del robot con un óptimo espacio de trabajo.

1.3. ESTRUCTURA DE LA MEMORIA

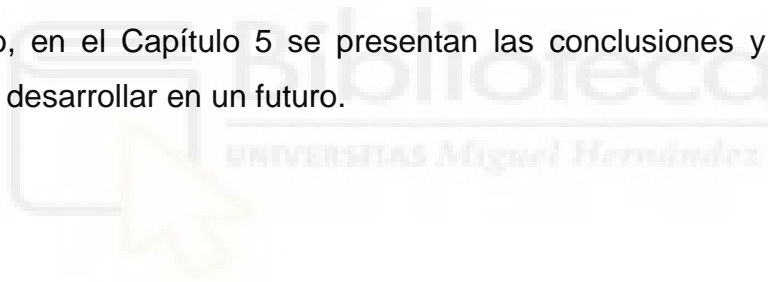
En el Capítulo 1 se presenta una breve introducción en la que se plantea la novedad del robot propuesto en comparación con otros que también emplean articulaciones helicoidales; y se definen los objetivos del trabajo de fin de grado.

El Capítulo 2 describe el robot propuesto; se resuelve la cinemática directa e inversa; se explican los algoritmos empleados para el cálculo y representación del espacio de trabajo y singularidades; y se detalla la interfaz creada que modela el movimiento del robot.

El proceso de construcción del prototipo que valida el movimiento del robot final se describe en el Capítulo 3. En orden, se especifican los materiales y tecnologías necesarias para su construcción; se establecen las medidas en base al análisis de sensibilidad para el diseño del prototipo; y se describe el proceso completo seguido. Este proceso detalla el modelaje 3D e impresión 3D de las piezas, al igual que el ensamblaje del prototipo entero y las soluciones llevadas a cabo para mejorar la estructura del robot. Finalmente, se presentan capturas del prototipo físico construido y se realizan pruebas de movimiento en lazo abierto para ratificar el funcionamiento esperado.

El método utilizado para la optimización y maximización del espacio de trabajo del robot se explica en el Capítulo 4. Realizando distintas pruebas de maximización del espacio de trabajo y examinando los resultados, se consigue la obtención de las medidas que optimizan el robot ideado.

Por último, en el Capítulo 5 se presentan las conclusiones y los posibles trabajos a desarrollar en un futuro.



2. ANALISIS DEL ROBOT PROPUESTO

2.1. DESCRIPCIÓN

La Figura 2.1 y la Figura 2.2 muestra la representación esquemática del robot paralelo planar propuesto. Éste emplea dos articulaciones de rotación (R_1 y R_2), dos articulaciones prismáticas (P_1 y P_2) y, cuatro de tipo helicoidal (H_1 , H_2 , H_3 y H_4). Contiene dos piernas ($R_1P_1H_1H_3$ y $R_2P_2H_2H_4$) que conectan el punto C, considerado el efector final, con la base fijada OXYZ.

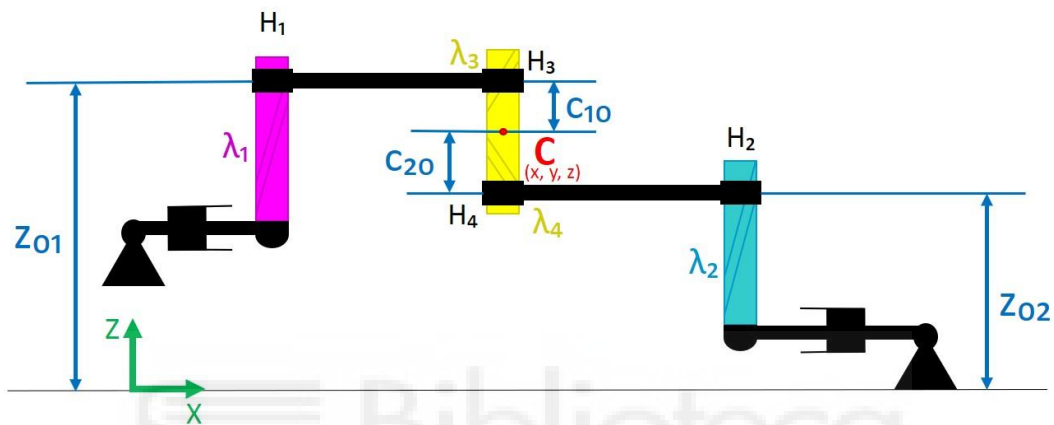


Figura 2.1: Alzado del robot

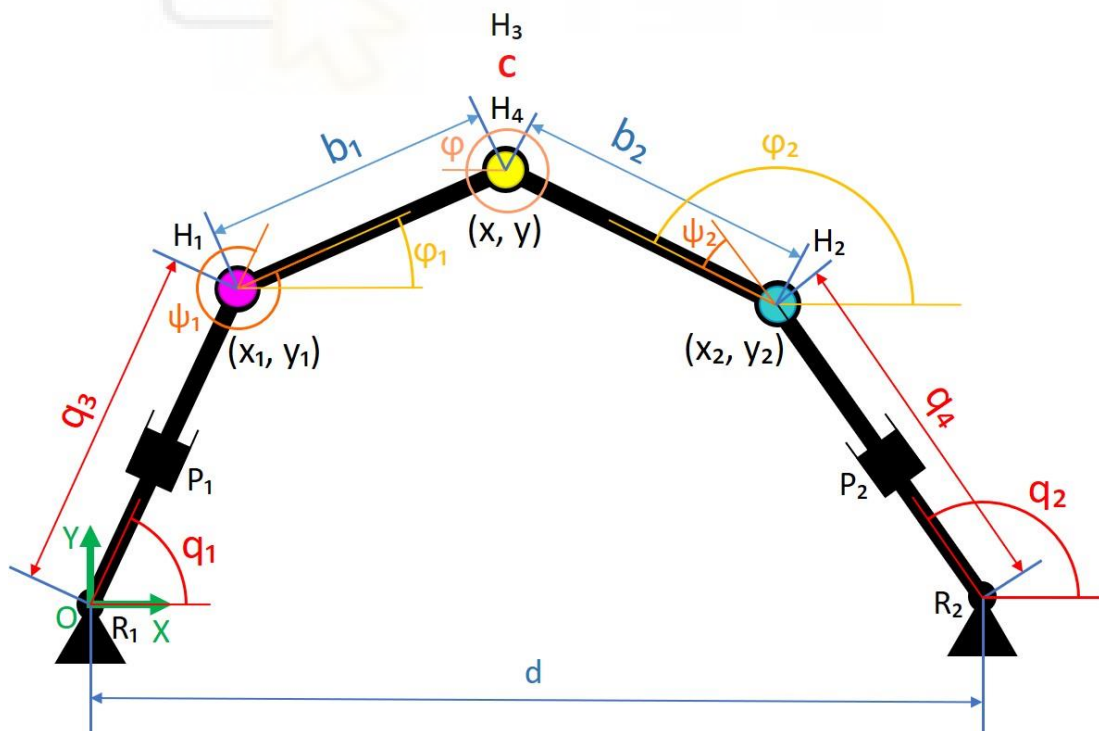


Figura 2.2: Planta del robot

El robot tiene 4 grados de libertad ($n=4$), por lo que genera movimientos de traslación en tres ejes y de rotación en un eje para el efector final (esto puede demostrarse mediante un análisis basado en la teoría de screws [13]). Gracias a las articulaciones helicoidales, este efector final (punto C) puede moverse libremente en el plano XYZ, aunque todos los eslabones del robot se mueven de forma planar por el XY. El efector final viene definido por su posición en coordenadas cartesianas con respecto al sistema de referencia OXYZ fijado en la base y por el eje de rotación de este mismo ($X = [x, y, z, \varphi]^T$). Las articulaciones activas son las R_1 , R_2 , P_1 y P_2 , y el vector de coordenadas articulares activas asociado a estas mismas es $\theta_a = [q_1, q_2, q_3, q_4]^T$, mientras que el vector de coordenadas articulares pasivas son $\theta_p = [\psi_1, \psi_2]^T$.

El método geométrico, el cual se basa en la descomposición de la geometría espacial del robot en varios problemas geométricos planos haciendo uso de herramientas geométricas y trigonométricas, es el que usará para resolver los problemas de cinemática del robot paralelo.

Para ello, hay que obtener una ecuación vectorial para cada pierna del robot, las cuales se combinan en una sola matriz para conseguir representar las restricciones geométricas entre todas las variables del robot en una sola ecuación vectorial no lineal.

Uno de los caminos es el descrito desde el origen del marco OXY hasta el efector final (punto C) a través de la secuencia de puntos $OR_1H_1H_3C$ quedando la ecuación vectorial de la siguiente manera:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_3 \cos q_1 + b_1 \cos(q_1 + \psi_1) \\ q_3 \sin q_1 + b_1 \sin(q_1 + \psi_1) \\ z_{01} + \psi_1 \lambda_1 + \varphi \lambda_3 + c_{10} \end{bmatrix} \quad (2.1)$$

El segundo camino posible es similar al anterior, pasando por los puntos $OR_2H_2H_4C$:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_4 \cos q_2 + d + b_2 \cos(q_2 + \psi_2) \\ q_4 \sin q_2 + b_2 \sin(q_2 + \psi_2) \\ z_{02} + \psi_2 \lambda_2 + \varphi \lambda_4 + c_{20} \end{bmatrix} \quad (2.2)$$

Como se puede apreciar, se utiliza trigonometría básica para obtener las coordenadas x e y . Sin embargo, para la obtención de la coordenada z , se necesita comprender el funcionamiento de las articulaciones helicoidales. Éstas vienen definidas por su paso, que representa cuánto avanza la articulación helicoidal longitudinalmente por cada radián de rotación (a la articulación H_1 le corresponde un λ_1 , a H_2 , λ_2 , y así sucesivamente). Por tanto, para hallar la coordenada z , se deben tener en cuenta las medidas iniciales en la coordenada z (z_{02} y c_{20}), pero también, el avance de las articulaciones helicoidales ($\psi_2 \lambda_2$ y $\varphi \lambda_4$) en esta coordenada.

Para ver la obtención y explicación completa de las ecuaciones anteriores, se precisa ver las ecuaciones del capítulo siguiente ([Cinemática directa](#)).

Combinándolas se obtiene la siguiente matriz:

$$F(X, \theta_a, \theta_p) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} = \begin{bmatrix} x - q_3 \cos q_1 - b_1 \cos(q_1 + \psi_1) \\ y - q_3 \sin q_1 - b_1 \sin(q_1 + \psi_1) \\ z - z_{01} - \psi_1 \lambda_1 - \varphi \lambda_3 - c_{10} \\ x - q_4 \cos q_2 + d - b_2 \cos(q_2 + \psi_2) \\ y - q_4 \sin q_2 - b_2 \sin(q_2 + \psi_2) \\ z - z_{02} - \psi_2 \lambda_2 - \varphi \lambda_4 - c_{20} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.3)$$

La ecuación es un sistema de 6 ecuaciones con 10 incógnitas ($x, y, z, \varphi, q_1, q_2, q_3, q_4, \psi_1, \psi_2$), por lo que hay $10 - 6 = 4$ variables libres. Esto confirma la sentencia anterior de que el robot tiene 4 grados de libertad.

Esta ecuación vectorial permitirá definir la cinemática directa, inversa, el espacio de trabajo y el análisis de singularidades del robot paralelo propuesto, como se explica en los siguientes capítulos.

2.2. CINEMÁTICA DIRECTA

En la cinemática directa (*FK Forward Kinematics*) se busca determinar la posición y orientación del extremo o efector final (x, y, z, φ) respecto de un sistema de referencia de la base. Para ello, se suponen conocidas las coordenadas articulares activas del robot (q_1, q_2, q_3, q_4) y los parámetros geométricos del mismo $(d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4)$.

Además, a partir de las coordenadas articulares activas conocidas (q_1, q_2, q_3, q_4) , se pueden calcular las coordenadas (x_1, y_1) y (x_2, y_2) que posicionan los ejes de las articulaciones helicoidales H_1 y H_2 en la Figura 2.2.

$$x_1 = q_3 * \cos q_1 \quad (2.4)$$

$$y_1 = q_3 * \sin q_1 \quad (2.5)$$

$$x_2 = q_4 * \cos q_2 + d \quad (2.6)$$

$$y_2 = q_4 * \sin q_2 \quad (2.7)$$

Dado que utilizando el método geométrico se emplean ecuaciones trigonométricas, se esperaría obtener infinitas soluciones para la cinemática debido a la periodicidad de éstas. Sin embargo, puesto que el robot está compuesto por dos piernas con forma antropomórfica y el giro de sus articulaciones es restringido, en la realidad sólo existen dos soluciones prácticas para la resolución de la cinemática. A estas dos soluciones se las suele llamar “codo arriba” (Figura 2.3) y “codo abajo” (Figura 2.4).

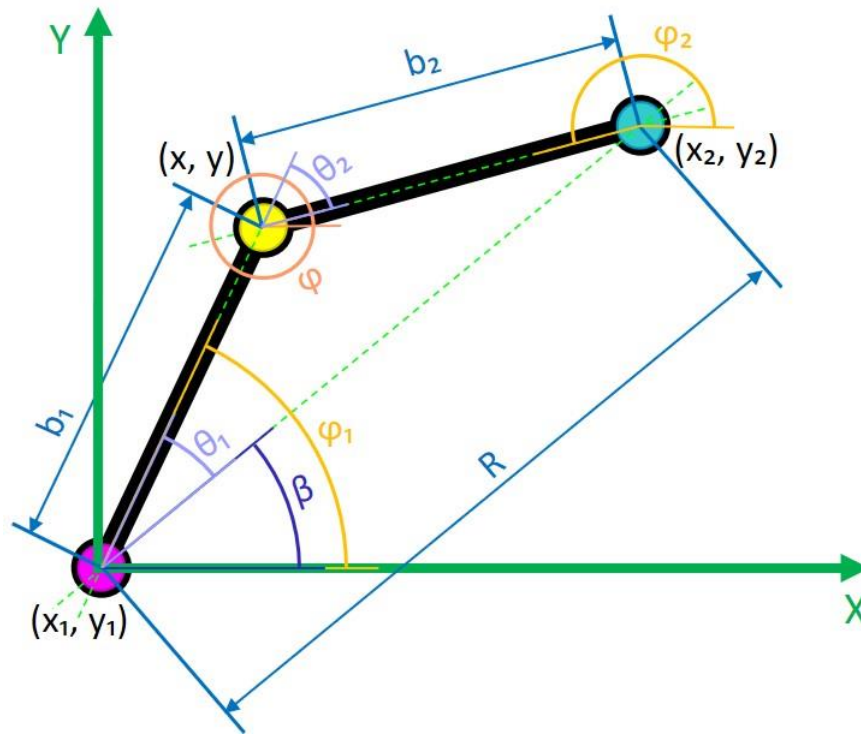


Figura 2.3: Codo arriba del robot

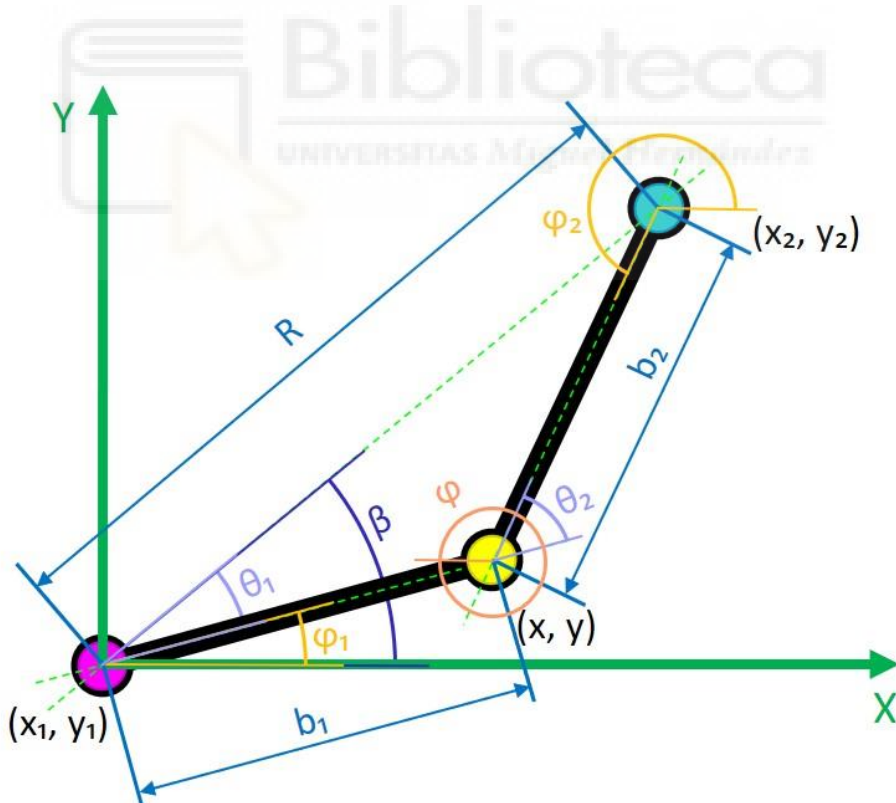


Figura 2.4: Codo abajo del robot

Centrando la atención en los triángulos imaginarios que forman las articulaciones helicoidales (H_1 , H_2 , H_3 y H_4) desde la planta del robot, como se

muestran en las figuras anteriores (Figuras 2.3 y 2.4), y teniendo en cuenta los parámetros supuestos conocidos ($q_1, q_2, q_3, q_4, d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4$), se llegan a los siguientes cálculos:

$$\tan \beta = \frac{y_2 - y_1}{x_2 - x_1} \rightarrow \beta = \text{atan2}(y_2 - y_1, x_2 - x_1) \quad (2.8)$$

$$R^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \rightarrow R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.9)$$

$$b_2^2 = b_1^2 + R^2 - 2b_1R * \cos \theta_1 \rightarrow \theta_1 = \pm \text{acos} \left(\frac{b_1^2 + R^2 - b_2^2}{2b_1R} \right)$$

Para poder variar el valor de θ_1 dependiendo del signo, se le agrega una variable binaria, la cual tomará valores +1 o -1 a conveniencia.

$$\theta_1 = \sigma * \text{acos} \left(\frac{b_1^2 + R^2 - b_2^2}{2b_1R} \right) \quad (2.10)$$

$$\varphi_1 = \beta + \theta_1 \quad (2.11)$$

La ecuación (2.10) confirma, por tanto, la existencia de las dos soluciones “codo arriba” y “codo abajo”, las cuales se muestran más claramente en las Figuras 2.5 y 2.6.

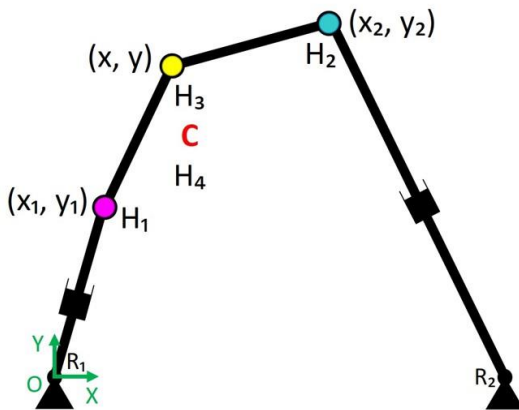


Figura 2.5: Posición codo arriba

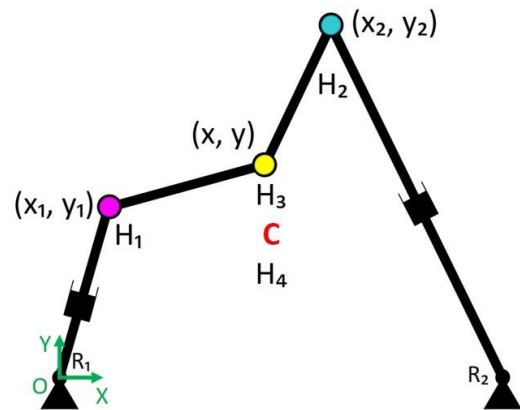


Figura 2.6: Posición codo abajo

Como se aprecia en las Figuras 2.5 y 2.6, se puede llegar a hallar la posición (x, y) del efector final (punto C) a través de cualquier solución. Por ello, tomando, por ejemplo, el camino descrito desde el origen del marco OXY

hasta el efector final (punto C) a través de la secuencia de puntos $OR_1H_1H_3C$, las ecuaciones son las siguientes:

$$x = x_1 + b_1 \cos \varphi_1 \quad (2.12)$$

$$y = y_1 + b_1 \sin \varphi_1 \quad (2.13)$$

De esta manera, se ya se puede calcular la incógnita restante.

$$\tan \varphi_2 = \frac{y-y_2}{x-x_2} \rightarrow \varphi_2 = \text{atan2}(y - y_2, x - x_2) \quad (2.14)$$

De la Figura 2.2 se averiguan las siguientes ecuaciones:

$$\varphi_1 = q_1 + \psi_1 \rightarrow \psi_1 = \varphi_1 - q_1 \quad (2.15)$$

$$\varphi_2 = q_2 + \psi_2 \rightarrow \psi_2 = \varphi_2 - q_2 \quad (2.16)$$

Para el cálculo de la coordenada z en cada camino, hay que considerar el efecto de las articulaciones helicoidales. Este viene determinado por su paso, al igual que una rosca, y por el ángulo girado.

Como se observa en la Figura 2.7, el paso es la distancia entre dos puntos situados en la misma generatriz. En otras palabras, es la distancia que se avanza por cada vuelta que se gira. Como se había comentado anteriormente, cada articulación helicoidal tiene un paso asociado ($H_1, H_2, H_3, H_4 \rightarrow \lambda_1, \lambda_2, \lambda_3, \lambda_4$).

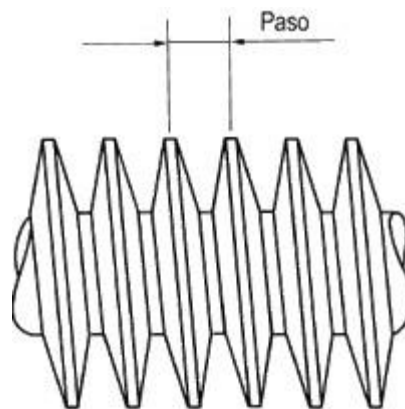


Figura 2.7: Representación del paso de una rosca

Por tanto, considerando el incremento de altura que provocan las articulaciones helicoidales y observando la Figura 2.1, se puede afirmar que la posición en el eje z del efector final es equivalente a la suma de: la altura inicial de las tuercas (z_{01} y z_{02}); los incrementos en z debido a los giros entre hélices y tuercas de las articulaciones helicoidales ($\psi_1 * \lambda_1$ y $\varphi * \lambda_3$); y, la altura entre tuercas y el efector final (c_{10} y c_{20}). Las ecuaciones para cada camino, respectivamente, son:

$$z = z_{01} + \psi_1 * \lambda_1 + \varphi * \lambda_3 + c_{10} \quad (2.17)$$

$$z = z_{02} + \psi_2 * \lambda_2 + \varphi * \lambda_4 + c_{20} \quad (2.18)$$

Igualando estas dos expresiones, se consigue despejar el valor de la orientación (φ) del efector final.

$$\begin{aligned} z = z &\rightarrow z_{01} + \psi_1 * \lambda_1 + \varphi * \lambda_3 + c_{10} = z_{02} + \psi_2 * \lambda_2 + \varphi * \lambda_4 + c_{20} \rightarrow \\ \varphi(\lambda_3 - \lambda_4) &= z_{02} - z_{01} + \psi_2 * \lambda_2 - \psi_1 * \lambda_1 + c_{20} - c_{10} \rightarrow \\ \varphi &= \frac{z_{02} - z_{01} + \psi_2 * \lambda_2 - \psi_1 * \lambda_1 + c_{20} - c_{10}}{\lambda_3 - \lambda_4} \end{aligned} \quad (2.19)$$

Finalmente, al sustituir el valor anterior de φ en cualquiera de las ecuaciones (2.17) o (2.18), la cinemática directa queda resuelta con los valores de (x, y, z, φ) hallados.

2.3. CINEMÁTICA INVERSA

En la cinemática inversa (*IK Inverse Kinematics*) se busca determinar las coordenadas articulares activas (q_1, q_2, q_3, q_4) del robot que permiten llevar el extremo o efector final del mismo a una posición (x, y, z) y orientación (φ) establecidas, suponiendo conocidos también los parámetros geométricos del robot ($d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4$).

Como se comentó en el capítulo de descripción, se puede resolver la cinemática inversa empleando las dos matrices (2.1) y (2.2). La matriz (2.1) corresponde a la primera pierna, y la (2.2), a la segunda pierna.

Partiendo de la matriz (2.1), se hace notoria la posibilidad de despejar la incógnita ψ_1 al conocer el resto de variables.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_3 \cos q_1 + b_1 \cos(q_1 + \psi_1) \\ q_3 \sin q_1 + b_1 \sin(q_1 + \psi_1) \\ z_{01} + \psi_1 \lambda_1 + \varphi \lambda_3 + c_{10} \end{bmatrix} \rightarrow \begin{bmatrix} x = q_3 \cos q_1 + b_1 \cos(q_1 + \psi_1) \\ y = q_3 \sin q_1 + b_1 \sin(q_1 + \psi_1) \\ \psi_1 = \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1} \end{bmatrix}$$

$$\psi_1 = \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1} \quad (2.20)$$

Siguiendo con las dos primeras ecuaciones, se decide separar la suma de q_1 y ψ_1 , tanto del seno como del coseno.

$$\begin{bmatrix} x = b_1 \cos q_1 \cos \psi_1 - b_1 \sin q_1 \sin \psi_1 + q_3 \cos q_1 \\ y = b_1 \sin q_1 \cos \psi_1 + b_1 \cos q_1 \sin \psi_1 + q_3 \sin q_1 \end{bmatrix} \quad (2.21)$$

Después se resuelve este sistema lineal de tamaño 2x2 despejando las incógnitas de $\sin q_1$ y $\cos q_1$.

$$\cos q_1 = \frac{b_1 x \cos \psi_1 + b_1 y \sin \psi_1 + q_3 x}{2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2} \quad (2.22)$$

$$\sin q_1 = \frac{b_1 y \cos \psi_1 - b_1 x \sin \psi_1 + q_3 y}{2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2} \quad (2.23)$$

Utilizando la identidad fundamental de la trigonometría, se consigue dejar una ecuación únicamente con la incógnita q_3 .

$$\begin{aligned} (\cos q_1)^2 + (\sin q_1)^2 &= 1 \rightarrow \\ \left(\frac{b_1 x \cos \psi_1 + b_1 y \sin \psi_1 + q_3 x}{2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2} \right)^2 + \left(\frac{b_1 y \cos \psi_1 - b_1 x \sin \psi_1 + q_3 y}{2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2} \right)^2 &= 1 \rightarrow \\ \frac{x^2 + y^2}{2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2} &= 1 \\ x^2 + y^2 &= 2b_1 q_3 \cos \psi_1 + b_1^2 + q_3^2 \end{aligned}$$

Resolviendo y despejando la incógnita q_3 se obtienen dos soluciones:

$$q_3 = -b_1 \cos \psi_1 - \sqrt{(-b_1^2 * (\sin \psi_1)^2 + x^2 + y^2)}$$

$$q_3 = -b_1 \cos \psi_1 + \sqrt{(-b_1^2 * (\sin \psi_1)^2 + x^2 + y^2)}$$

Como se aprecia, la única diferencia entre ellas es el signo que acompaña a la raíz cuadrada, por lo que convirtiéndolo en variable (σ_1), se conseguiría tener las dos soluciones de q_3 en una sola ecuación cambiando el valor de σ_1 a +1 o -1, según se desee.

$$q_3 = -b_1 \cos \psi_1 + \sigma_1 * \sqrt{(-b_1^2 * (\sin \psi_1)^2 + x^2 + y^2)} \quad (2.24)$$

Además, el valor de q_1 se consigue con las ecuaciones (2.22) y (2.23):

$$\tan q_1 = \frac{\sin q_1}{\cos q_1}$$

$$q_1 = \text{atan2}(\sin q_1, \cos q_1) \quad (2.25)$$

Si hacemos el mismo proceso con la segunda pierna, se logra averiguar también los valores de q_2 y q_4 , consiguiendo resolver completamente la cinética inversa hallando todos los valores de las variables articulares (q_1, q_2, q_3, q_4).

$$\psi_2 = \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2} \quad (2.26)$$

$$q_4 = -b_2 \cos \psi_2 + \sigma_2 * \sqrt{(-b_2^2 * (\sin \psi_2)^2 + x^2 - 2 d x + y^2 + d^2)} \quad (2.27)$$

La variable (σ_2) también es una variable binaria que toma los valores de +1 o -1, según convenga, para conseguir las dos soluciones de q_4 en una sola ecuación.

$$\cos q_2 = \frac{b_2 (x-d) \cos \psi_2 + b_2 y \sin \psi_2 + q_4 (x-d)}{2b_2 q_4 \cos \psi_2 + b_2^2 + q_4^2} \quad (2.28)$$

$$\sin q_2 = \frac{b_2 y \cos \psi_2 + b_2 (d-x) \sin \psi_2 + q_4 y}{2b_2 q_4 \cos \psi_2 + b_2^2 + q_4^2} \quad (2.29)$$

$$q_2 = \text{atan2}(\sin q_2, \cos q_2) \quad (2.30)$$

Cada pierna es como un robot serie de dos grados de libertad y, por consiguiente, dos soluciones con valores de variables articulares diferentes, sitúan al efector final con la misma posición y orientación.

Dado que existen dos soluciones para cada pierna del robot, hay cuatro soluciones para la cinemática inversa (Figura 2.8), las cuales pueden ser caracterizadas por el signo escogido en σ_1 y σ_2 de q_3 y q_4 , respectivamente. Asimismo, puesto que el valor de q_3 y q_4 varía, el valor de q_1 y q_2 varía también.

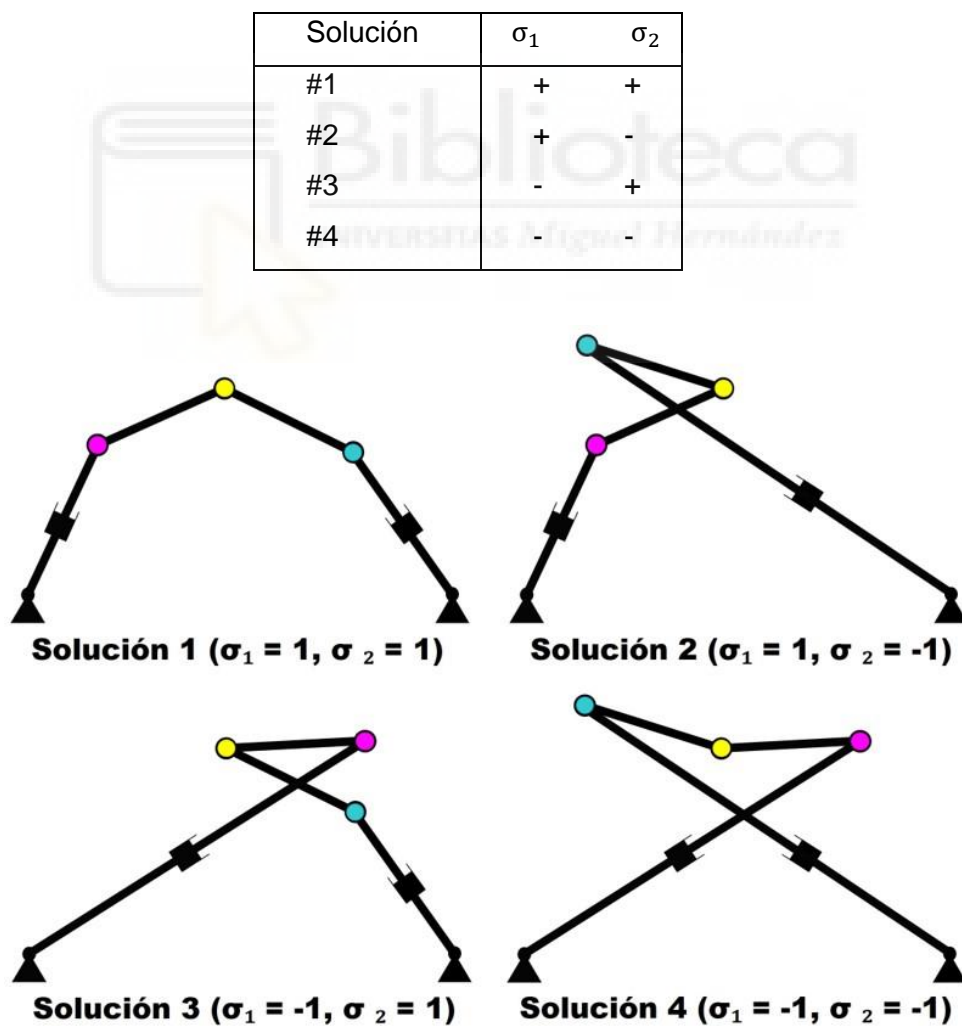


Figura 2.8: Configuraciones solución para la cinemática inversa

2.4. ESPACIO DE TRABAJO

El espacio de trabajo del robot propuesto es el conjunto de puntos X del espacio 4D (x, y, z, φ) que puede alcanzar el efector final para todos los valores posibles de las coordenadas articulares activas θ_a . En general, las coordenadas articulares activas pueden ser restringidas en algunos intervalos, pero en este caso se considerará que $q_1, q_2 \in [-\pi, \pi]$, por lo que los actuadores pueden efectuar revoluciones completas, y que, q_3, q_4 están limitados entre $[q_{3_min}, q_{3_max}]$ y $[q_{4_min}, q_{4_max}]$.

El espacio de trabajo de un robot paralelo puede ser calculado de varias formas. El enfoque más directo consiste en discretizar y barrer el espacio de las coordenadas operacionales X , y resolver la cinemática inversa para todos los puntos de esta discretización. Cada vez que se resuelve la cinemática inversa, se obtiene el vector de coordenadas articulares θ_a para las piernas del robot, y el conjunto de todos los puntos operacionales X , que dan solución real a la cinemática inversa y cumplen con los límites articulares, constituyen una aproximación discreta del espacio de trabajo del robot.

Este algoritmo tiene algunos inconvenientes debido a su simplicidad. En primer lugar, requiere demasiados cálculos. Con este algoritmo, se deben barrer el plano XYZ y el intervalo $\varphi = [-\pi, \pi]$, correspondiente a la orientación del efector final. Si los ejes de x, y, z, φ están discretizados en n puntos cada uno, entonces la cinemática inversa debe resolverse n^4 veces, por lo que el valor de n no podrá ser demasiado grande para evitar que el algoritmo emplee un tiempo excesivo calculando. Además, el espacio de trabajo se representará mediante la nube de puntos generada a partir de la discretización, la cual estará limitada por el número de puntos n y que, al no poder tener un valor muy alto, la nube de puntos será más espaciosa y menos precisa de lo deseable. Por otro lado, no todos los vectores producen una solución factible para la cinemática inversa, ya que ésta requiere el cálculo de raíces cuadradas, como en la ecuación (2.27), que puede producir un resultado imaginario si el radicando es negativo.

Por ello, después de resolver la cinemática inversa para cada punto de la discretización, se comprueba que cada valor del vector de variables articulares θ_a tiene una solución real. Además, solo aquellos vectores que cumplan con los límites articulares de longitudes ($q_{3_min}, q_{3_max}, q_{4_min}, q_{4_max}$) serán aceptadas como solución y, en tal caso, se almacenarán las coordenadas del punto estudiado para representarlo gráficamente en la nube de puntos total.

Algoritmo 1: Cálculo y representación del espacio de trabajo 4D

```

1:  $\varphi = \varphi_{min}$ 
2: for  $m = 1$  to  $N_\varphi$  with step= $d_\varphi$  do
3:    $z = z_{min}$ 
4:   for  $k = 1$  to  $N_z$  with step= $d_z$  do
5:      $x = x_{min}$ 
6:     for  $i = 1$  to  $N_x$  with step= $d_x$  do
7:        $y = y_{min}$ 
8:       for  $j = 1$  to  $N_y$  with step= $d_y$  do
9:         Calcular variables articulares ( $q_1, q_2, q_3, q_4$ ) con la cinemática inversa
10:        if reales( $q_1, q_2, q_3, q_4$ ) then
11:          if  $q_3 \geq q_{3\_min}$  and  $q_3 \leq q_{3\_max}$  and  $q_4 \geq q_{4\_min}$  and  $q_4 \leq q_{4\_max}$  then
12:            Guardar coordenadas  $x, y, z, \varphi$ 

```

El algoritmo 1 resume los pasos necesarios para obtener una aproximación del espacio de trabajo. Éste discretiza cada eje en n puntos (N_{phi}, N_z, N_y, N_x), y genera un espacio de cuatro dimensiones (x, y, z, φ). Los límites mínimos y máximos del espacio de trabajo en el plano XYZ estudiado se han establecido de forma aproximada teniendo en cuenta los resultados obtenidos.

Con el algoritmo anterior se genera un espacio de trabajo 4D en el espacio (x, y, z, φ). No obstante, en ciertas ocasiones es conveniente visualizar un subespacio 2D, ya sea por agilizar el tiempo en los cálculos, o por proporcionar una representación gráfica planar más intuitiva. En ese caso, se dejarían constantes las variables z y φ , por ejemplo, y simplemente se

ejecutaría el código comenzando por la línea 5 del Algoritmo 1, para obtener una sección del espacio de trabajo en el plano XY.

2.5. SINGULARIDADES

En un robot paralelo existen dos tipos de singularidades:

I) De tipo 1 o serie. Corresponden a configuraciones en las que el efector final no puede trasladarse o rotar en alguna o algunas direcciones del espacio, ya que requieren velocidades articulares no acotadas para obtener velocidades lineales y angulares finitas del efector final. En este tipo de singularidades, se produce la pérdida de grados de libertad y, por lo tanto, la pérdida de destreza o capacidad de movimiento.

II) De tipo 2 o paralelo. Representan configuraciones en las que, pese a que se bloqueen todos los actuadores, el efector final se traslada o rota en alguna o algunas direcciones del espacio, aunque sea de forma limitada mediante movimientos diferenciales, lo que hace necesario un par de actuación infinito para resistir fuerzas externas finitas. En este tipo de singularidades, se produce una pérdida de control o rigidez del robot.

Como punto de partida para hallar los puntos singulares, se deben obtener las ecuaciones que relacionan las variables articulares (q_1, q_2, q_3, q_4) activas con las de posición y orientación del efector final (x, y, z, φ) a través de los parámetros geométricos. Así pues, se deben eliminar las variables pasivas (ψ_1, ψ_2) de estas. Esto se consigue despejando ψ_1 y ψ_2 de las terceras ecuaciones de las matrices (2.1) y (2.2), como ya se había hecho en el capítulo anterior obteniendo las ecuaciones (2.20) y (2.26), y sustituyendo éstas en las demás ecuaciones.

$$\left[\begin{array}{l} x = q_3 \cos q_1 + b_1 \cos(q_1 + \psi_1) \\ y = q_3 \sin q_1 + b_1 \sin(q_1 + \psi_1) \\ \psi_1 = \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1} \end{array} \right] \rightarrow \left\{ \begin{array}{l} x = q_3 \cos q_1 + b_1 \cos \left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1} \right) \\ y = q_3 \sin q_1 + b_1 \sin \left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1} \right) \end{array} \right.$$

$$\begin{bmatrix} x = q_4 \cos q_2 + d + b_2 \cos(q_2 + \psi_2) \\ y = q_4 \sin q_2 + b_2 \sin(q_2 + \psi_2) \\ \psi_2 = \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2} \end{bmatrix} \rightarrow \begin{cases} x = q_4 \cos q_2 + d + b_2 \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \\ y = q_4 \sin q_2 + b_2 \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \end{cases}$$

Pasando los términos al mismo lado de cada ecuación se logran las ecuaciones deseadas, esto es, un vector F dependiente de $X = (x, y, z, \varphi)$ y $\theta_a = (q_1, q_2, q_3, q_4)$.

$$F(X, \theta_a) = 0 = \begin{bmatrix} q_3 \cos q_1 + b_1 \cos\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) - x \\ q_3 \sin q_1 + b_1 \sin\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) - y \\ q_4 \cos q_2 + d + b_2 \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) - x \\ q_4 \sin q_2 + b_2 \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) - y \end{bmatrix} \quad (2.31)$$

Derivando el vector anterior con respecto al tiempo se obtiene la ecuación en la que intervienen las velocidades del efector final y las de las variables articulares activas.

$$\left[\frac{\partial F}{\partial X}\right] \cdot \left[\frac{\partial X}{\partial t}\right] + \left[\frac{\partial F}{\partial \theta_a}\right] \cdot \left[\frac{\partial \theta_a}{\partial t}\right] = 0 \quad (2.32)$$

Siendo J_X y J_{θ_a} las matrices jacobianas en función de F con respecto a los vectores X y θ_a , respectivamente.

$$J_X = \left[\frac{\partial F}{\partial X}\right] = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} & \frac{\partial F_1}{\partial z} & \frac{\partial F_1}{\partial \varphi} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} & \frac{\partial F_2}{\partial z} & \frac{\partial F_2}{\partial \varphi} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial y} & \frac{\partial F_3}{\partial z} & \frac{\partial F_3}{\partial \varphi} \\ \frac{\partial F_4}{\partial x} & \frac{\partial F_4}{\partial y} & \frac{\partial F_4}{\partial z} & \frac{\partial F_4}{\partial \varphi} \end{bmatrix} \quad (2.33)$$

$$J_{\theta_a} = \left[\frac{\partial F}{\partial \theta_a} \right] = \begin{bmatrix} \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial q_2} & \frac{\partial F_1}{\partial q_3} & \frac{\partial F_1}{\partial q_4} \\ \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial q_2} & \frac{\partial F_2}{\partial q_3} & \frac{\partial F_2}{\partial q_4} \\ \frac{\partial F_3}{\partial q_1} & \frac{\partial F_3}{\partial q_2} & \frac{\partial F_3}{\partial q_3} & \frac{\partial F_3}{\partial q_4} \\ \frac{\partial F_4}{\partial q_1} & \frac{\partial F_4}{\partial q_2} & \frac{\partial F_4}{\partial q_3} & \frac{\partial F_4}{\partial q_4} \end{bmatrix} \quad (2.34)$$

Para una mejor visualización de las siguientes matrices, no se sustituyen los valores de $\psi_1 = \frac{z-z_{01}-\varphi \lambda_3 - c_{10}}{\lambda_1}$ y $\psi_2 = \frac{z-z_{02}-\varphi \lambda_4 - c_{20}}{\lambda_2}$, aunque se tienen en cuenta en el cálculo de estas:

$$J_X = \begin{bmatrix} -1 & 0 & -\frac{b_1 \sin(q_1 + \psi_1)}{\lambda_1} & \frac{b_1 \lambda_3 \sin(q_1 + \psi_1)}{\lambda_1} \\ 0 & -1 & \frac{b_1 \cos(q_1 + \psi_1)}{\lambda_1} & -\frac{b_1 \lambda_3 \cos(q_1 + \psi_1)}{\lambda_1} \\ -1 & 0 & -\frac{b_2 \sin(q_2 + \psi_2)}{\lambda_2} & \frac{b_2 \lambda_4 \sin(q_2 + \psi_2)}{\lambda_2} \\ 0 & -1 & \frac{b_2 \cos(q_2 + \psi_2)}{\lambda_2} & -\frac{b_2 \lambda_4 \cos(q_2 + \psi_2)}{\lambda_2} \end{bmatrix} \quad (2.35)$$

$$J_{\theta_a} = \begin{bmatrix} -b_1 \sin(q_1 + \psi_1) - q_3 \sin q_1 & 0 & \cos q_1 & 0 \\ b_1 \cos(q_1 + \psi_1) + q_3 \cos q_1 & 0 & \sin q_1 & 0 \\ 0 & -b_2 \sin(q_2 + \psi_2) - q_4 \sin q_2 & 0 & \cos q_2 \\ 0 & b_2 \cos(q_2 + \psi_2) + q_4 \cos q_2 & 0 & \sin q_2 \end{bmatrix} \quad (2.36)$$

Calculando el determinante de cada matriz y anulándolos, se obtienen las ecuaciones que definen las singularidades de tipo paralelo y serie.

Singularidades tipo paralelo:

$$\det(J_X) =$$

$$\frac{b_1 b_2 (\lambda_4 - \lambda_3) \cos\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right)}{\lambda_1 \lambda_2} + \frac{b_1 b_2 (\lambda_3 - \lambda_4) \sin\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right)}{\lambda_1 \lambda_2} = 0 \quad (2.38)$$

Singularidades tipo serie:

$$\begin{aligned}
 \det(J_{\theta_a}) = & -\cos\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) \\
 & \cdot \left[b_1 b_2 \cos q_1 \cos q_2 \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \right. \\
 & + b_1 b_2 \cos q_1 \sin q_2 \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \\
 & \left. + b_1 q_4 \cos q_1 \right] - \sin\left(q_1 + \frac{z - z_{01} - \varphi \lambda_3 - c_{10}}{\lambda_1}\right) \\
 & \cdot \left[b_1 b_2 \sin q_1 \cos q_2 \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \right. \\
 & \left. + b_1 b_2 \sin q_1 \sin q_2 \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) + b_1 q_4 \sin q_1 \right] \\
 & - b_2 q_3 \cos q_2 \cos\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) \\
 & - b_2 q_3 \sin q_2 \sin\left(q_2 + \frac{z - z_{02} - \varphi \lambda_4 - c_{20}}{\lambda_2}\right) - q_3 q_4 = 0 \quad (2.37)
 \end{aligned}$$

De nuevo, para representar el conjunto de puntos del espacio de trabajo para los que se alcanza una singularidad (*Singularity loci*), se necesita utilizar una aproximación discreta que encuentre los puntos más próximos al valor cero.

Algoritmo 2: Cálculo y representación de las curvas de singularidades para z y φ constantes

- 1: $x = x_{min}$
- 2: for $i = 1$ to N_x with step= d_x do
- 3: $y = y_{min}$
- 4: for $j = 1$ to N_y with step= d_y do
- 5: Calcular variables articulares (q_1, q_2, q_3, q_4) con la cinemática inversa
- 6: Evaluar $\det(J_X)$ y almacenar su valor en cada punto (x, y) .
- 7: $x = x_{min}$
- 8: for $i = 1$ to N_x with step= d_x do
- 9: $y = y_{min}$
- 10: for $j = 1$ to N_y with step= d_y do

```

11:   for  $d_i = -1$  to 1 with step=1 do
12:     for  $d_j = -1$  to 1 with step=1 do
13:       if  $d_i \neq 0$  or  $d_j \neq 0$  then
14:          $i_{vecino} = i + d_i$ 
15:          $j_{vecino} = i + d_j$ 
16:         if dentro_límites_espacio ( $i_{vecino}$  and  $j_{vecino}$ ) then
17:           if reales ( $\theta_{a_{actual}}$  and  $\theta_{a_{vecino}}$ ) and  $\det(J_{X_{actual}}) * \det(J_{X_{vecino}}) < 0$  then
18:             Interpolar para hallar  $(x_{singular}, y_{singular})$  con  $\det(J_X) = 0$ 

```

El algoritmo 2 resume los pasos necesarios para obtener una aproximación de las singularidades. Discretizando en n puntos los ejes OX y OY, se barre este espacio 2D y se resuelve la cinemática inversa para hallar las variables articulares. Siendo ya conocidas las variables articulares, se puede proceder a evaluar el determinante de la matriz jacobiana J_x en cada punto del barrido. Una vez guardados estos puntos, se vuelve a barrer el espacio XY y cada punto estudiado se compara con sus puntos vecinos para lograr averiguar si entre estos puntos se produce el valor nulo del determinante de J_x que representa un punto singular de tipo paralelo. Es decir, a excepción de los puntos en el límite del espacio estudiado, cada punto tiene ocho puntos vecinos. Por tanto, se crea un subespacio 2D compuesto por los ocho puntos vecinos y el punto actual como origen del mismo, con las coordenadas (d_i, d_j) representando las de cada punto. Al recorrer este subespacio, se va comparando cada valor del determinante de J_x del punto vecino con el del actual, evitando comparar valores con los del propio punto actual $((d_i, d_j) \neq (0, 0))$. Si el punto vecino tiene un valor negativo y el punto actual, un valor positivo, o viceversa, significa que entre ellos se produce el valor nulo buscado para la ecuación de las singularidades. Por esto, se establece como condición que los valores de los puntos comparados sean reales y que el producto entre sus determinantes sea negativo para realizar una interpolación lineal que calcule de forma más precisa las coordenadas del punto en el cual el valor del determinante de J_x es nulo.

$$\frac{x_{vecino} - x_{actual}}{\det(J_{x_{vecino}}) - \det(J_{x_{actual}})} = \frac{x_{singular} - x_{actual}}{\det(J_{x_{sing_paralelo}}) - \det(J_{x_{actual}})} \rightarrow$$

$$x_{singular} = \frac{(x_{vecino} - x_{actual}) \cdot (0 - \det(J_{x_{actual}}))}{\det(J_{x_{vecino}}) - \det(J_{x_{actual}})} + x_{actual} \quad (2.38)$$

$$\frac{y_{vecino} - y_{actual}}{\det(J_{y_{vecino}}) - \det(J_{y_{actual}})} = \frac{y_{singular} - y_{actual}}{\det(J_{y_{sing_paralelo}}) - \det(J_{y_{actual}})} \rightarrow$$

$$y_{singular} = \frac{(y_{vecino} - y_{actual}) \cdot (0 - \det(J_{y_{actual}}))}{\det(J_{y_{vecino}}) - \det(J_{y_{actual}})} + y_{actual} \quad (2.39)$$

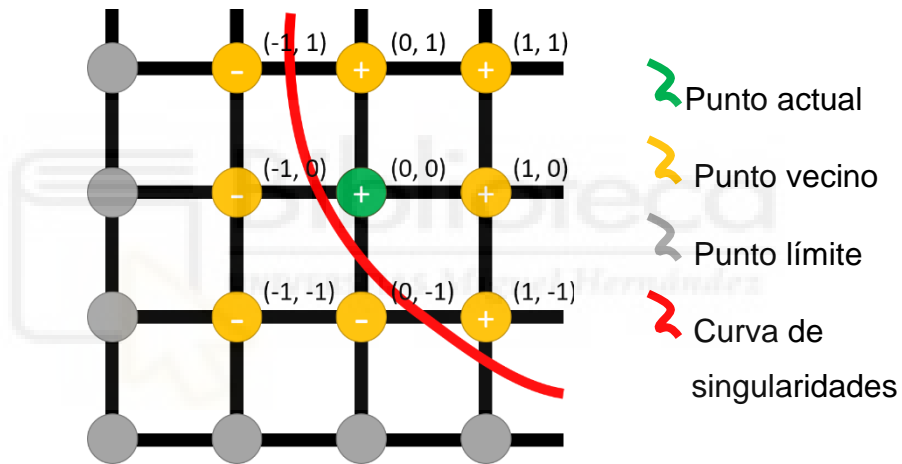


Figura 2.9: Representación aproximada del funcionamiento del Algoritmo 2 en la nube de puntos

El conjunto de todos los puntos singulares encontrados trazará de forma aproximada la curva ecuación de las singularidades de tipo paralelo.

Por otro lado, otra curva interesante a representar es la frontera entre valores reales e imaginarios del determinante de la matriz jacobiana J_x , que sería lo mismo que comprobar si la solución de la cinemática inversa en cada punto es real o imaginaria. La condición para ello es que se produzca un cambio, entre el punto estudiado y el punto vecino, de valor real a imaginario, o viceversa. Si es así, se asume por simplicidad, que las coordenadas del punto frontera se encuentran equidistantes entre los dos puntos.

Algoritmo 2: Continuación

```
19:         if (real( $\det(J_{x_{actual}})$ ) and imaginario( $\det(J_{x_{vecino}})$ )) or
           (imaginario( $\det(J_{x_{actual}})$ ) and real( $\det(J_{x_{vecino}})$ )) then
20:              $x_{frontera} = (x_{actual} + x_{vecino})/2$ 
21:              $y_{frontera} = (y_{actual} + y_{vecino})/2$ 
```

Además, se puede demostrar que las singularidades de tipo serie, donde $\det(J_{\theta_a}) = 0$, son aquellos puntos en los que la solución a la cinemática inversa cambia de real a imaginaria o viceversa. Esto se puede ver claramente en la propia ecuación (2.37) del determinante de la matriz jacobiana J_{θ_a} . Si los valores que devuelve la cinemática inversa (q_1, q_2, q_3, q_4) son reales, el determinante será real, si no, el determinante tendrá un resultado imaginario. Asimismo, cuando se detecta el cambio del $\det(J_x)$ de real a imaginario o viceversa en el Algoritmo 2, se está detectando la frontera del espacio de trabajo o lo que es lo mismo, la frontera entre puntos alcanzables e inalcanzables, que darían solución real e imaginaria a la cinemática inversa, respectivamente. Por tanto, no es necesario utilizar el determinante de J_{θ_a} para representarlos, se puede comprobar lo mismo utilizando el Algoritmo 2 ya implementado.

De esta forma, quedan resueltos los dos tipos de singularidades en un solo algoritmo.

2.6. SIMULACIÓN CON EJS

Para poder visualizar la apariencia del robot junto con su espacio de trabajo y singularidades, según la configuración de este, se programan los algoritmos vistos en capítulos anteriores con la herramienta de software “Easy Java Simulations (EJS)”. La programación de dichos algoritmos se encuentra en el [Anexo 1](#) de este documento.

Usando esta herramienta, también se puede ver, de forma aproximada, los movimientos que efectuará el robot. Para ello, se crean dos ventanas de

visualización, una para la del robot en 3D, y otra en 2D para la de la planta del robot.

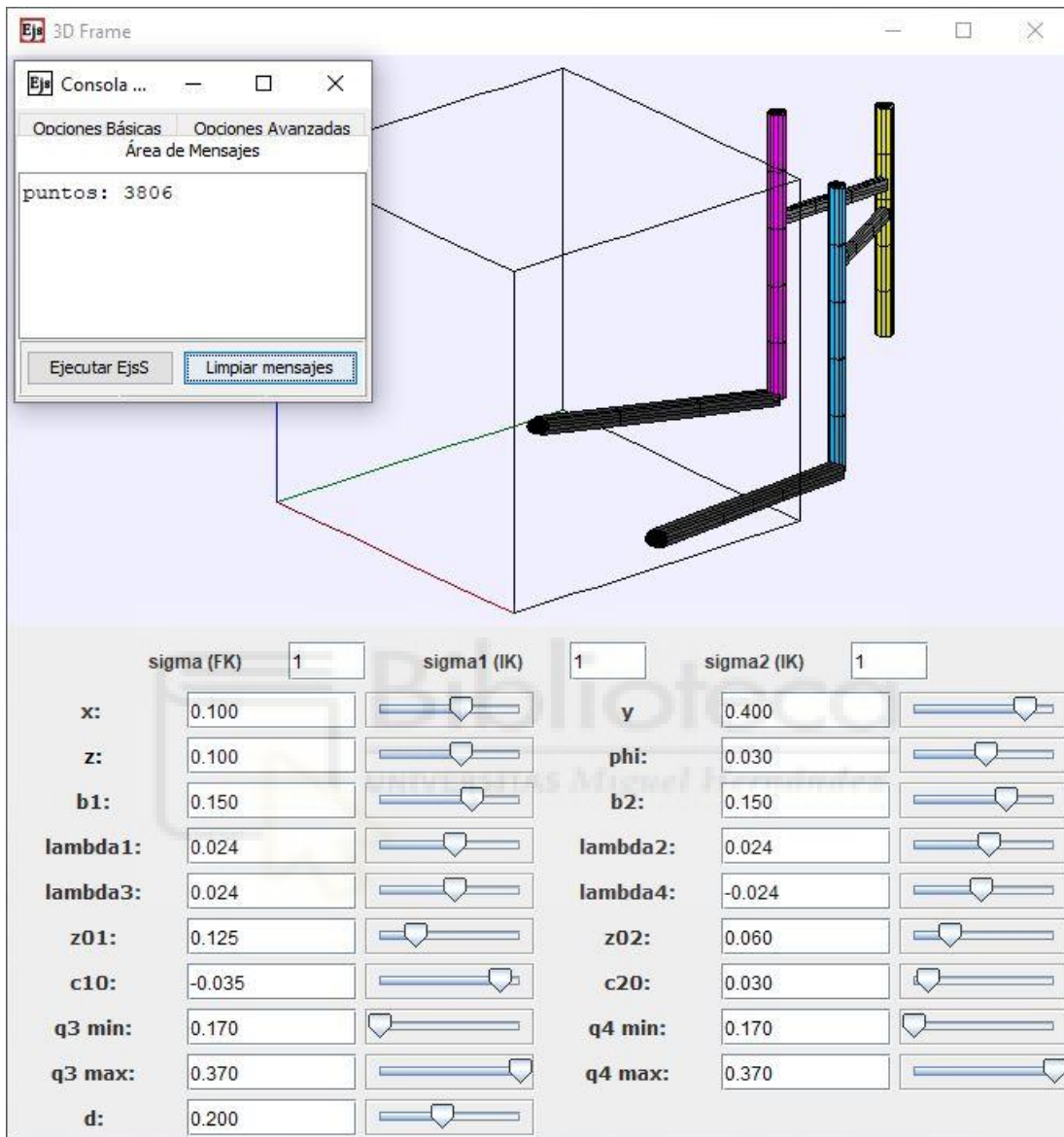


Figura 2.10: Ventana 3D de la interfaz creada con EJS

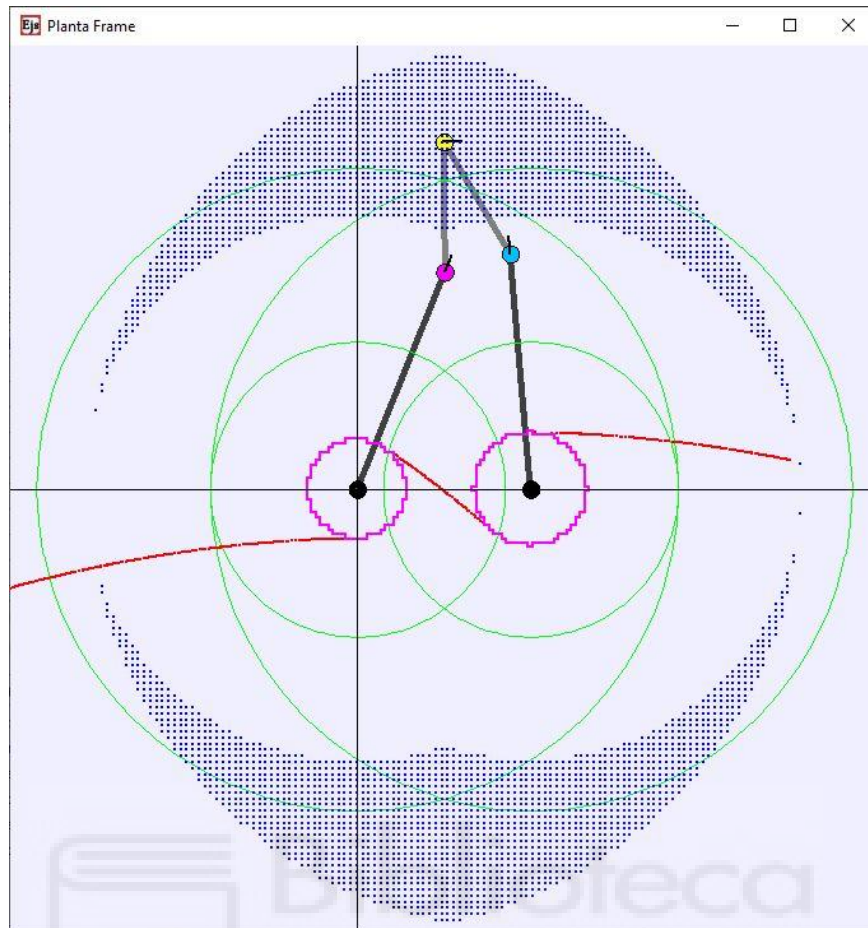


Figura 2.11: Ventana Planta de la interfaz creada con EJS

En la primera ventana 3D, mostrada en la Figura 2.10, se incluyen también las variables más importantes como las sigmas ($\sigma, \sigma_1, \sigma_2$), de las cuales depende la solución de la cinemática directa (*FK*) e inversa (*IK*), las de posición y orientación (x, y, z, φ) , y todos los parámetros de diseño del robot $(d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4, q_{3_min}, q_{3_max}, q_{4_min}, q_{4_max})$. De esta forma, se pueden ir variando sus valores y ver la influencia de éstas sobre el funcionamiento del robot.

En la segunda ventana, mostrada en la Figura 2.11, se incluyen el espacio de trabajo para z y φ constantes, representado con la nube de puntos azules, y las curvas de singularidades, diferenciando las de tipo serie en color magenta y las de tipo paralelo en color rojo. En color verde se pueden apreciar los límites $q_{3_min}, q_{3_max}, q_{4_min}, q_{4_max}$, donde los valores mínimos coinciden y los máximos también.

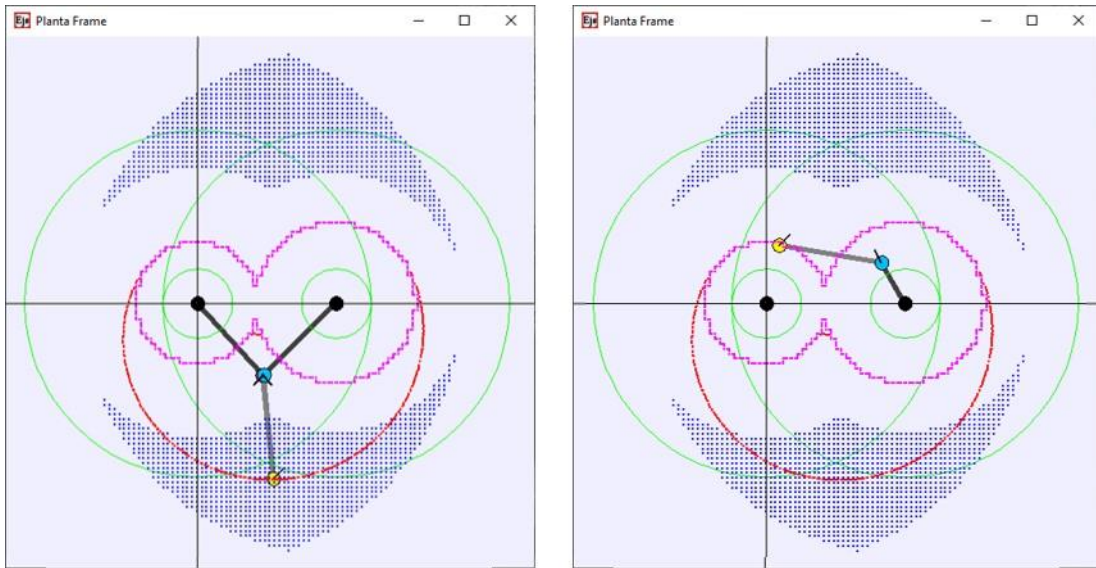


Figura 2.12: Ejemplos de configuraciones singulares del robot



3. CONSTRUCCIÓN DE PROTOTIPO CON IMPRESIÓN 3D

3.1. TECNOLOGÍAS Y MATERIALES UTILIZADOS

- Impresión 3D por método de extrusión de material

Técnica de manufactura aditiva donde se genera una pieza con volumen mediante sucesivas deposiciones ordenadas y controladas de material termoplástico. Este material es extruido a través de la boquilla de una impresora 3D y solidifica al poco tiempo de salir de la boquilla, apoyándose y adhiriéndose a la capa de material anterior. Finalizada una capa, se aumenta la altura y se comienza a realizar el camino prefijado de la capa siguiente. La sucesión de capas va formando la pieza deseada.

- Impresora 3D

Máquina que utiliza la impresión 3D para crear piezas a partir de un modelo 3D hecho por una herramienta de diseño asistido por computador (CAD). En este proyecto, la herramienta CAD usada es Autodesk Inventor (versión estudiante) y los modelos de impresora 3D han sido "CTC Dual" y "Creality CR-10s".

- Filamento bobinado 3D

Material termoplástico utilizado para la creación de una pieza por impresión 3D. En este caso, se ha utilizado el material PLA (ácido poliláctico) con un diámetro de 1.75 mm.

- Tornillería

Tornillos, arandelas, tuercas de freno, varillas roscadas y pletinas metálicas para el montaje y sujeción de las partes del robot.

- Base Metálica

Estructura metálica que hace de soporte o base del robot.

- Actuator eléctrico

Dispositivo mecánico que transforma la electricidad en fuerza, la cual permite accionar cualquier dispositivo para llevar a cabo determinado movimiento o acción.

- Lineal (4). Aquel que realiza una fuerza en línea recta. Para este prototipo, se han empleado actuadores lineales modelo ACTUONIX P16-200-64-12-P que utilizan un motor de corriente continua internamente. Los números en el nombre del modelo significan: milímetros de carrera, es decir, la diferencia entre el actuador totalmente retraído y totalmente extendido (“200”); relación de reducción de la reductora colocada entre el motor de corriente continua y el husillo (“64”); voltios de tensión nominal del actuador (“12”); y, la existencia del potenciómetro que suministra un voltaje proporcional a la extensión del actuador, lo que permite realizar un control en bucle cerrado del robot (P).
- Rotatorio (2). Aquel que realiza una fuerza rotatoria. En este caso, se utilizan motores de corriente continua de modelo Maxon 110164 con reductora modelo 144002.

3.2. MEDIDAS INICIALES

Como paso previo al diseño óptimo del robot, se realiza el análisis de sensibilidad. El objetivo de este estudio es averiguar cuánta influencia tiene cada parámetro de diseño del robot ($d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4, q_{3_min}, q_{3_max}, q_{4_min}, q_{4_max}$) sobre el espacio de trabajo. Según los resultados obtenidos, se establecerán los valores finales para la construcción del prototipo del robot. Este prototipo servirá a su vez para poder prevenir los posibles fallos que no son perceptibles únicamente con la herramienta EJS, como las colisiones entre articulaciones. De esta manera, se podrá mejorar la configuración entre piezas del robot optimizado final.

Para esto, partiendo de un diseño inicial, se variará un único parámetro geométrico con un diferencial pequeño, tanto por encima del valor actual como

por debajo de este. A continuación, se volverá a dejar el parámetro en su valor inicial y se seguirá el mismo procedimiento con el resto de parámetros. Para poder comparar y apreciar las diferencias con el diseño inicial, se realizan capturas de pantalla a todas las variaciones efectuadas, las cuales se pueden visualizar en el Anexo 2. Esta tarea se puede hacer más fácilmente creando un menú con deslizadores y cajas numéricas en la ventana 3D de la herramienta EJS, tal y como aparece en la Figura 2.10.

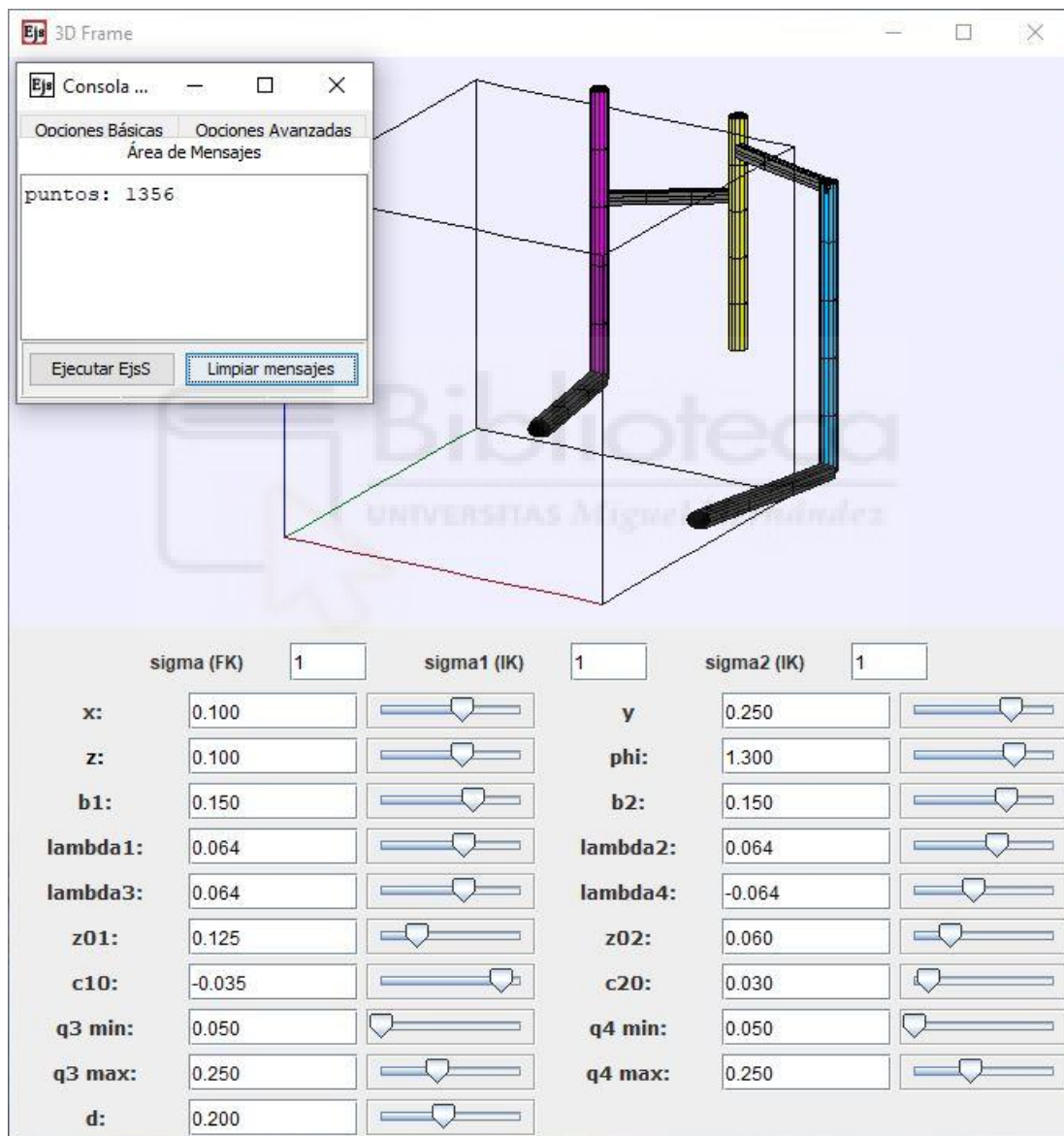


Figura 3.1: Ventana 3D de EJS con los valores del diseño inicial para el análisis de sensibilidad

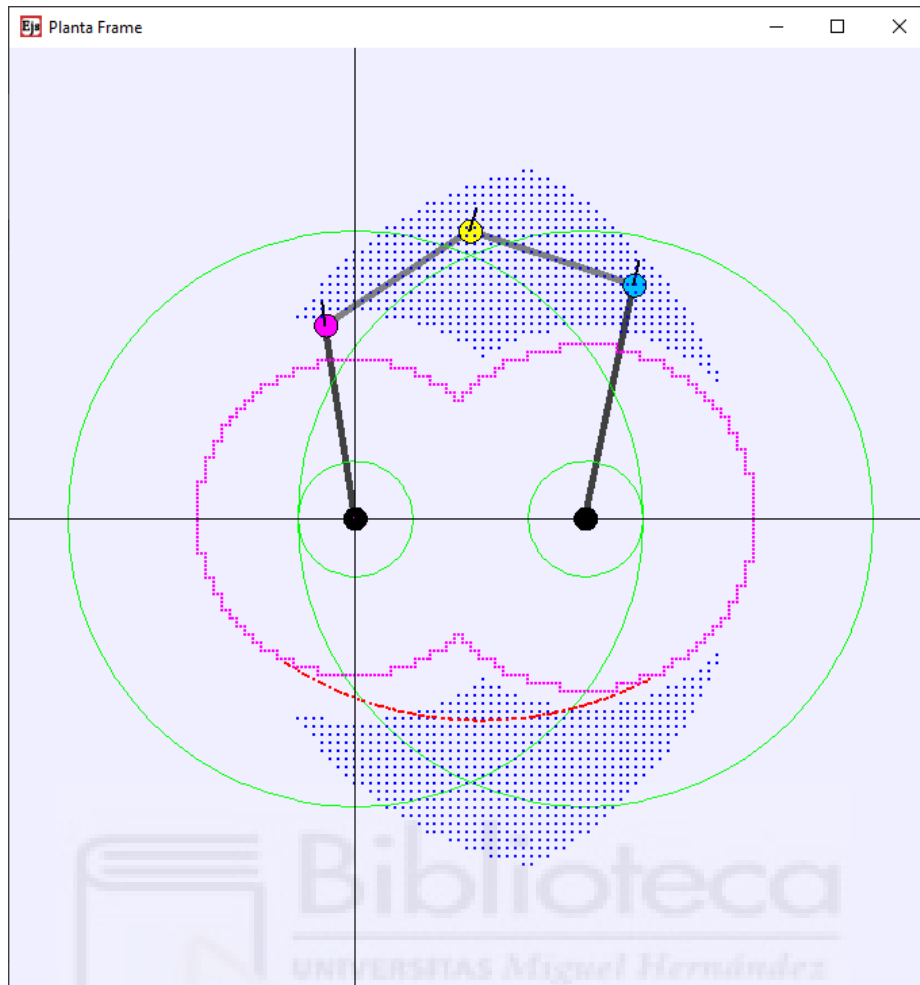


Figura 3.2: Ventana Planta de EJS con el espacio de trabajo generado por los valores del diseño inicial para el análisis de sensibilidad donde z y φ son constantes

El análisis de sensibilidad realizado en el Anexo 2 revela dos tipos de parámetros geométricos: aquellos que cuanto menor son en valor, mayor espacio de trabajo generan, como los parámetros $d, b_2, \lambda_3, z_{01}, c_{10}, q_{3_min}$ y q_{4_min} ; y, aquellos que cuanto más grandes son en valor, mayor espacio de trabajo consiguen, como $b_1, \lambda_1, \lambda_2, \lambda_4, z_{02}, c_{20}, q_{3_max}$ y q_{4_max} .

Tras analizar los 15 parámetros citados anteriormente de forma individual, los valores finales de los parámetros geométricos establecidos son los siguientes:

$$d = 0.2 \text{ m}$$

$$b_1 = b_2 = 0.15 \text{ m}$$

$$\lambda_1 = \lambda_2 = \lambda_3 = 0.15 \frac{\text{m}}{\text{rev}} = \frac{0.15 \text{ m}}{2\pi \text{ rad}} \approx 0.024 \frac{\text{m}}{\text{rad}}$$

$$\lambda_4 = -0.15 \frac{m}{rev} = -\frac{0.15}{2\pi} \frac{m}{rad} \approx -0.024 \frac{m}{rad}$$

$$z_{01} = 0.125 \text{ m}$$

$$z_{02} = 0.06 \text{ m}$$

$$c_{10} = -0.035 \text{ m}$$

$$c_{20} = 0.03 \text{ m}$$

$$q_{3_min} = q_{4_min} = 0.17 \text{ m}$$

$$q_{3_max} = q_{4_max} = 0.37 \text{ m}$$

Aunque el análisis de sensibilidad aporta mucha información sobre cómo poder mejorar el espacio de trabajo, se opta por tomar un término medio en cuanto a los valores de los parámetros geométricos. Es por esto por lo que varios de los valores iniciales no cambian para realizar el prototipo del robot. Hay que tener en cuenta que los valores iniciales de los parámetros no han sido estudiados escrupulosamente y que el resultado obtenido en el análisis de sensibilidad se ve condicionado por éstos. Por tanto, si se eligen otros valores iniciales, no necesariamente el resultado debe ser el mismo.

Además, también se han tenido en cuenta otros aspectos. El primero es establecer la diferencia entre mínimos y máximos de las variables articulares q_3 y q_4 , que viene condicionada por la carrera de los actuadores utilizados. Como cuánto más grande es esta diferencia, mayor espacio de trabajo se consigue, se eligen los actuadores con mayor carrera (20 cm). Pero si se construye el prototipo tal y como se había diseñado el robot en un primer momento, el centro de gravedad de éste no estará equilibrado y el prototipo flectará cuando los actuadores lineales se extiendan, siendo necesario corregir y mejorar el diseño inicial. Lo mismo ocurre para los valores de las varillas pasivas b_1 y b_2 . Para los valores iniciales escogidos, el análisis revela la mejora del espacio de trabajo cuando b_1 aumenta y cuando b_2 disminuye en valor. Sin embargo, se decide dejar lo más equilibrado posible al prototipo y se eligen los mismos valores iniciales. Con los valores z_{01} y z_{02} o c_{10} y c_{20} , que también varían de forma inversa entre ellos para poder aumentar el espacio de trabajo, se determina emplear los mismos valores iniciales. Por

otra parte, el análisis evidencia la importancia del parámetro d , que representa la distancia entre apoyos de los actuadores lineales, llegando a ser $d = 0$ el mejor valor posible para el robot. Es decir, cuando los ejes z de cada actuador lineal son coaxiales, se consigue mejorar el espacio de trabajo indiscutiblemente, consiguiendo que el prototipo pueda llegar a zonas nuevas, no alcanzables al mismo tiempo por ningún otro parámetro.

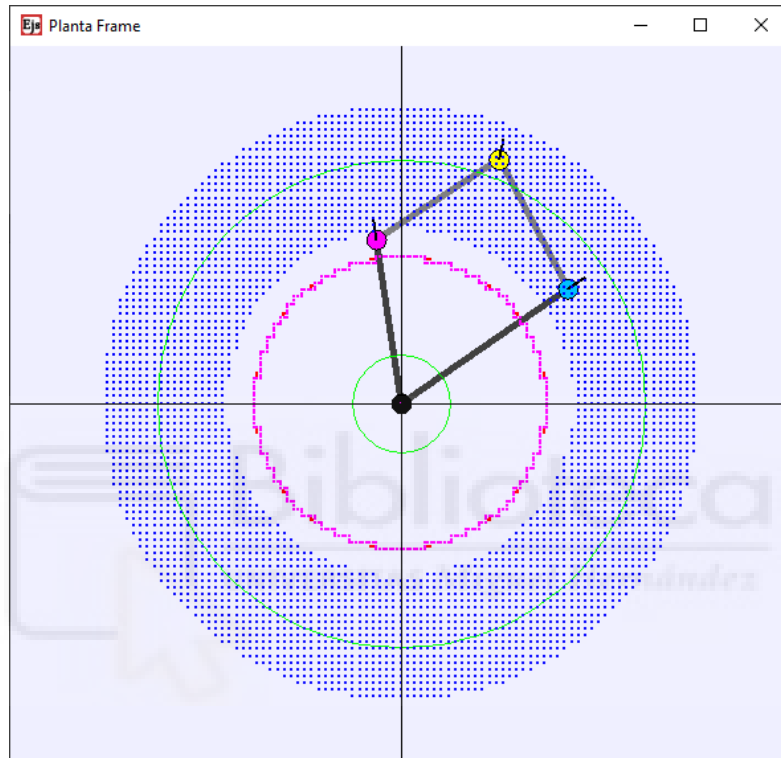


Figura 3.3: Espacio de trabajo alcanzable cuando $d = 0$ y el resto de parámetros tienen los valores del diseño inicial

Asimismo, los valores de los pasos de las articulaciones helicoidales $\lambda_1, \lambda_2, \lambda_3$ y λ_4 son de gran relevancia. Inicialmente, éstos toman valores absolutos bastante grandes ($0.064 \frac{m}{rad} \approx 0.4 \frac{m}{rev}$) para conseguir un avance mayor con un giro de ϕ pequeño, lo que implica una trayectoria de la hélice de la articulación helicoidal más alargada. No obstante, en términos prácticos, cuanto más alargada es la trayectoria de la hélice, más rectilínea se convierte y más difícil será efectuar el movimiento del robot completo. Por ello, se decide disminuir el valor absoluto de estos parámetros y se determina que un avance de 15 cm por cada vuelta es una buena opción para el prototipo. El valor de λ_4 debe ser diferente en signo o valor que λ_3 ya que, de no ser así, el efector

final se atornillaría o desatornillaría libremente a través de las tuercas, aun bloqueando todos los actuadores. En otras palabras, el efector final se movería sin control y el robot se encontraría siempre en una configuración singular. Esto se puede ver claramente en la ecuación (2.38), donde $\det(J_X) = 0$ si $\lambda_3 = \lambda_4$, independientemente del resto de valores. Por tanto, para evitar esta singularidad se cambia el signo de λ_4 , por ejemplo, dejándolo con el mismo valor absoluto que el resto de pasos de las articulaciones helicoidales $\lambda_1, \lambda_2, \lambda_3$.

Por otra parte, se establecen las medidas de las articulaciones helicoidales. H_1 tendrá 16 cm de alto, H_2 , 10 cm, y la altura total de H_3+H_4 será también de 16 cm, repartidos a partes iguales a cada articulación.

Finalmente, el espacio de trabajo conseguido con los valores finales elegidos se muestra en la [Figura 2.11](#), que está compuesto por 3806 puntos cuando las variables de posición y orientación son las que se visualizan en el menú de la [Figura 2.10](#). Si se compara con la [Figura 3.2](#), que muestra el espacio de trabajo generado antes del análisis compuesto por 1356 puntos, la mejoría es visible.

No obstante, estos valores elegidos no son los mejores. Como se ha comentado anteriormente, este análisis es un paso previo a la obtención del robot optimizado, el cual estará basado en un estudio más exhaustivo para cada valor de los parámetros. Este diseño óptimo del robot se estudiará en el capítulo 4.

3.3. DESARROLLO

3.3.1. MODELADO 3D Y MONTAJE

Una vez ya se tienen las medidas iniciales determinadas para la construcción del prototipo, se decide hacer uso de la herramienta CAD Autodesk Inventor para crear las piezas necesarias del prototipo.

Lo primero a generar son las articulaciones helicoidales. Éstas están compuestas cada una por un cilindro y una tuerca que enlazan entre sí a través de una hélice cilíndrica de paso $\lambda_1, \lambda_2, \lambda_3$ y λ_4 , según la articulación.

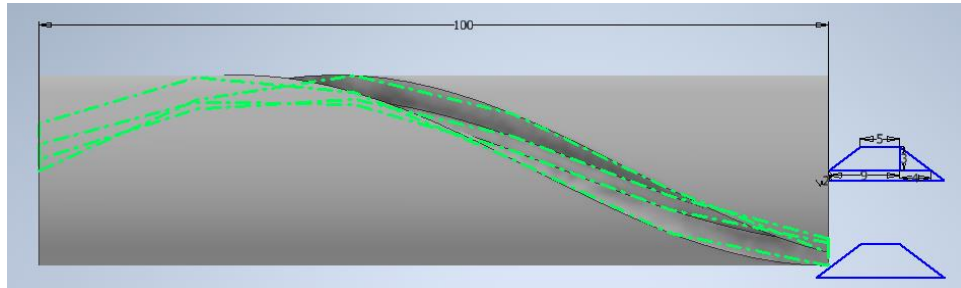


Figura 3.4: Visualización de la trayectoria de la hélice cilíndrica restada al cilindro para generar una articulación helicoidal

En la creación de los cilindros se utiliza la forma de un trapecio isósceles para generar la hélice cilíndrica que hay que restarle. El trapecio se genera con la base bastante más grande que la de la base menor, de manera que los ángulos formados sean más inclinados. De este modo, durante la impresión de los cilindros e independientemente del valor del paso utilizado para generar la hélice, cada capa impresa serviría de soporte para la capa siguiente. Esto es así debido a que cada capa únicamente sobresaldría ligeramente sobre la anterior, cosa que no ocurriría en caso de que la trayectoria de la hélice se realizara con una forma cuadrada, ya que se formaría un voladizo que vencería por la propia gravedad. Este efecto se puede entender más fácilmente con la Figura 3.5.

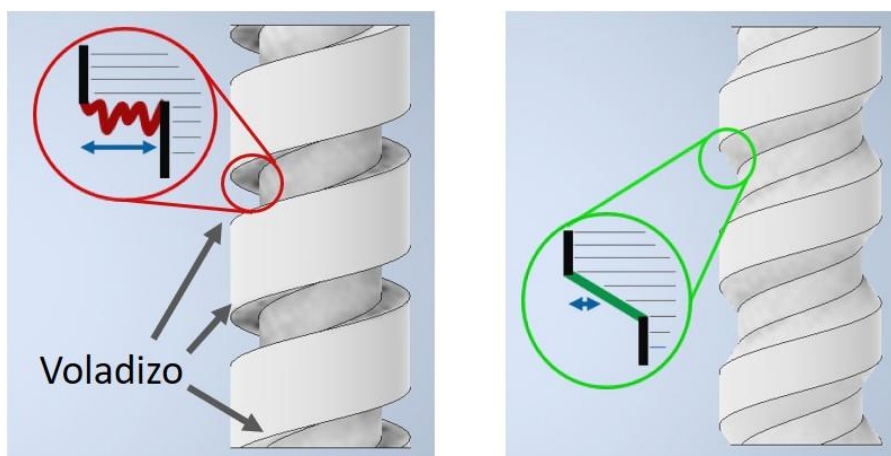


Figura 3.5: Comparativa entre la impresión de los cilindros usando una forma cuadrada (izquierda) o trapezoidal (derecha) para la trayectoria de la hélice cilíndrica

Siguiendo este procedimiento y teniendo en cuenta las alturas para cada articulación establecidas en el capítulo de Medidas iniciales, se obtienen los cuatro cilindros de cada articulación helicoidal H_1 , H_2 , H_3 y H_4 con sus respectivos pasos $\lambda_1, \lambda_2, \lambda_3$ y λ_4 .

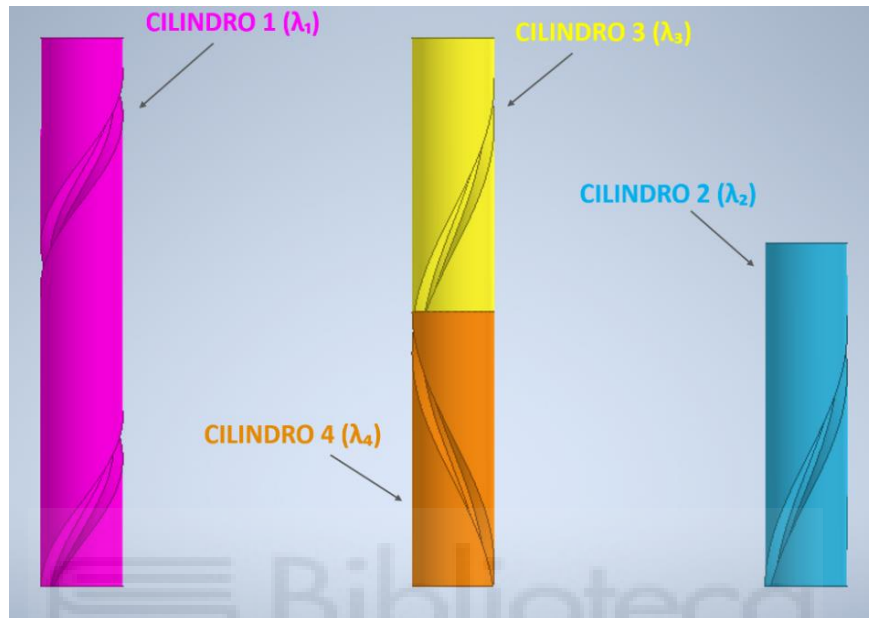


Figura 3.6: Visualización de los cilindros de cada articulación helicoidal

Para cada tuerca de las articulaciones helicoidales se emplea también, incluyendo cierta holgura, el mismo trapecio isósceles anterior para la generación de la hélice cilíndrica que debe ser unida a la tuerca base.

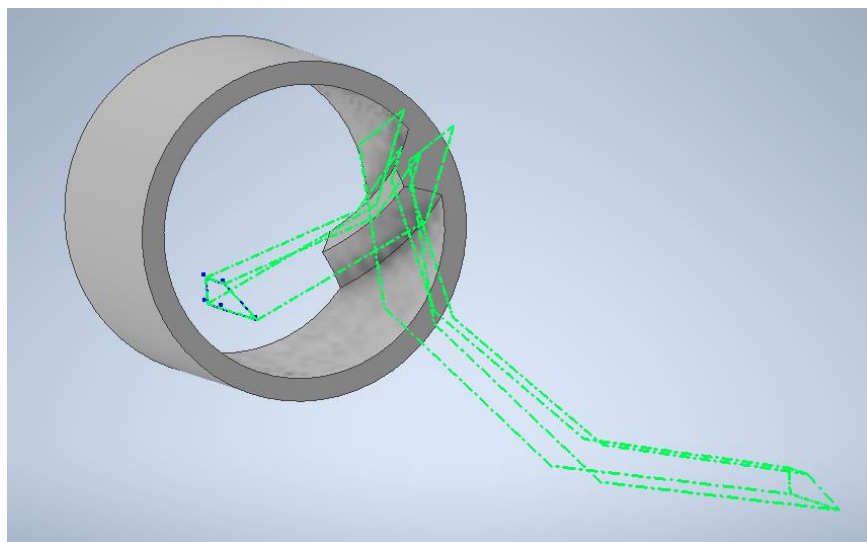


Figura 3.7: Representación de la trayectoria de la hélice cilíndrica y del resultado de unir ésta con la tuerca base

Después se le añaden dos orejetas para que sirvan de soporte de las varillas pasivas b_1 y b_2 y las tuercas quedan terminadas.

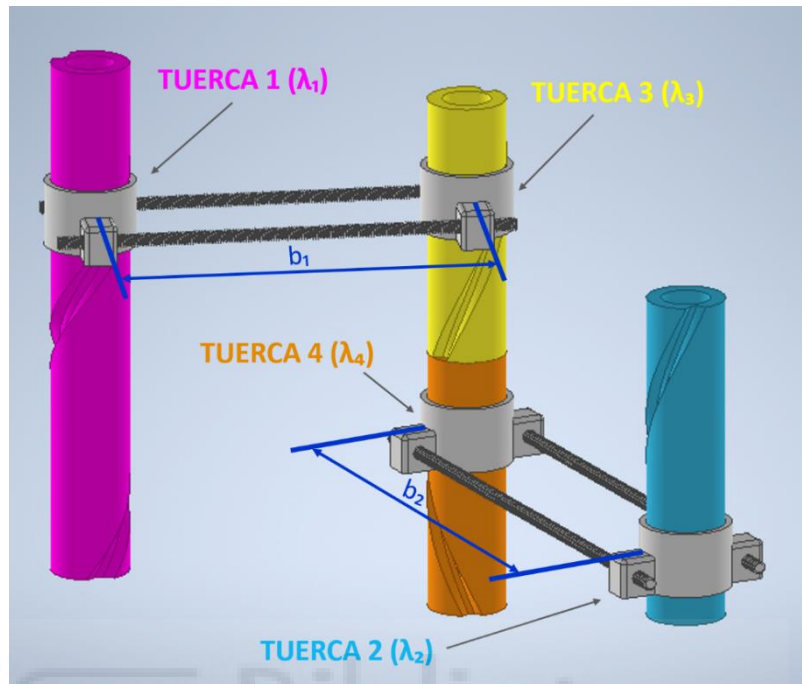


Figura 3.8: Ensamblaje de las articulaciones helicoidales y las varillas pasivas b_1 y b_2

Además, las articulaciones H_1 y H_2 se deben acoplar a los actuadores lineales, por lo que se necesita desarrollar otras dos piezas auxiliares. La idea es crear una pieza que encaje en el extremo del actuador (“plug”) y que también quede sujeta a la articulación. Para ello, se crea una pieza base (“aux_act_cil”) a la cual se le resta una pieza sólida generada a partir de la pieza “plug” (Figura 3.10). De esta forma, se consigue que el actuador lineal físico se ajuste perfectamente a la pieza auxiliar nueva. También se añaden los agujeros en los que se introducirá el tornillo para dejarlo todo solidario.

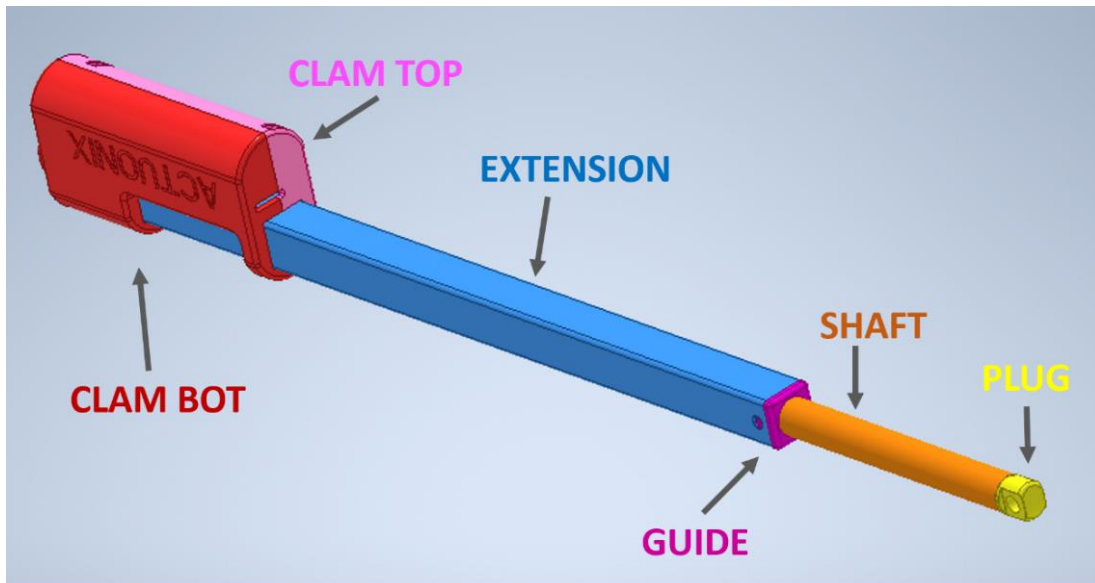


Figura 3.9: Partes que conforman el modelo 3D de un actuador lineal

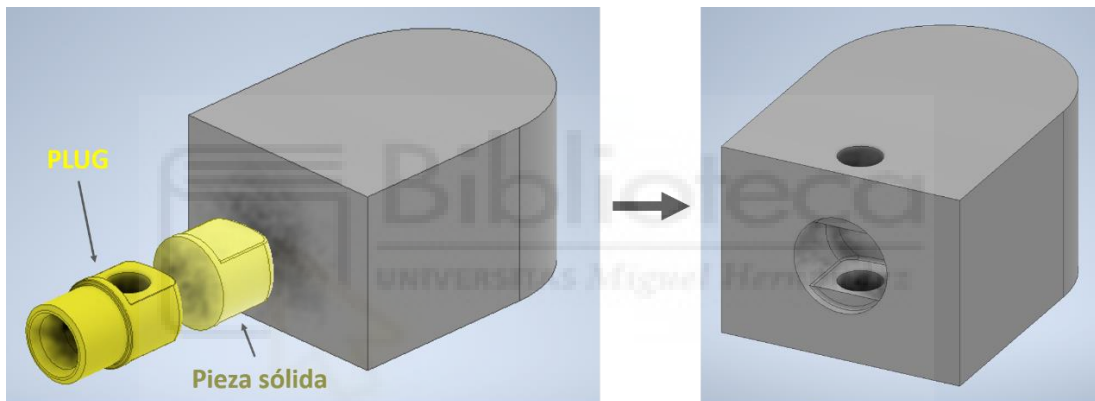


Figura 3.10: Transición de la pieza aux_act_cil

Ahora falta modificar la pieza auxiliar para poder ajustar también la articulación helicoidal. Para esto se piensa añadir un cilindro a la pieza base que se introduzca en la articulación helicoidal y que se ajusten mediante tornillos. Hay que modificar también la articulación helicoidal para ello, dejándola con cierto espesor y eliminando la parte central de la misma para poder introducir el cilindro de la pieza auxiliar (Figura 3.11). Además, se añaden los agujeros para introducir los tornillos que acoplarán las dos piezas (Figura 3.12).

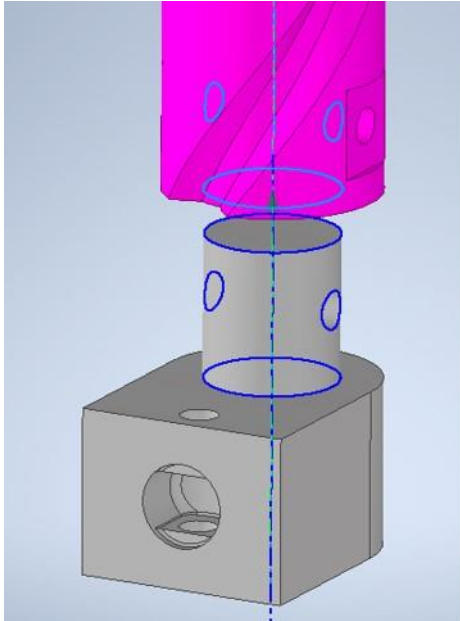


Figura 3.11: Representación de la coaxialidad entre el cilindro de la pieza aux_act_cil y el orificio del cilindro de la articulación helicoidal

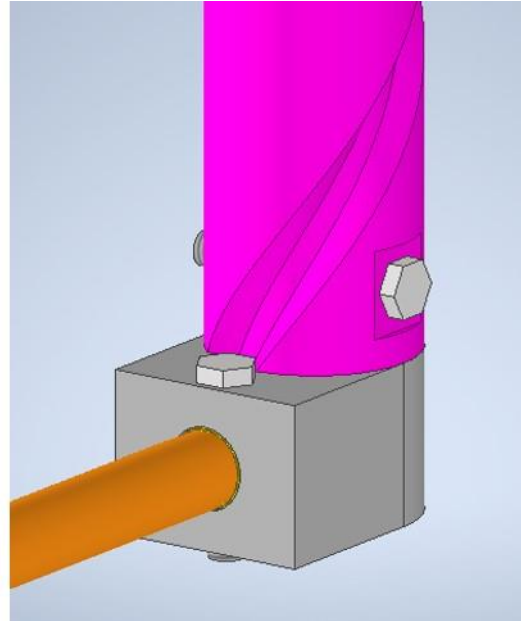


Figura 3.12: Pieza aux_act_cil ensamblada con el actuador lineal y el cilindro de la articulación helicoidal

Por último, falta por incluir los motores de corriente continua al prototipo, que van apoyados sobre una pletina adjunta a la base metálica que soporta todo el peso del prototipo. Éstos se posicionan al principio de cada cadena del robot, es decir, en una posición anterior a los actuadores lineales. Para esto también se necesita una pieza auxiliar, bastante parecida en cuanto al proceso de creación a la pieza aux_act_cil, que haga de enlace entre cada actuador lineal y rotatorio. Partiendo de una pieza base (“aux_lin_rot”), se resta una pieza sólida generada a partir de las piezas “clam_bot” y “clam_top” del actuador lineal (Figura 3.13).

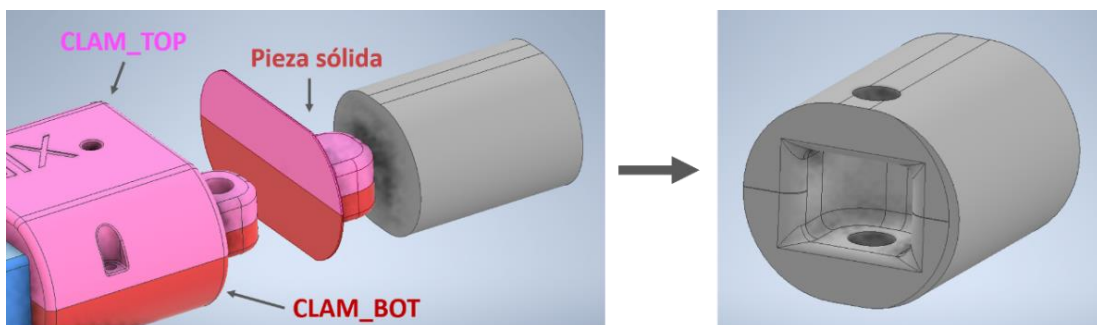


Figura 3.13: Transición de la pieza aux_lin_rot

Para que la pieza aux_lin_rot haga de enlace entre el motor de corriente continua, se le incluye otro cilindro en el que se inserta el eje del motor, el cual estará fijado por un tornillo (Figura 3.14). Esta primera idea no es la más acertada debido a que no hay suficiente espacio para que el tornillo introducido haga un buen apriete contra el eje del motor, por lo que se agranda este espacio (Figura 3.15).

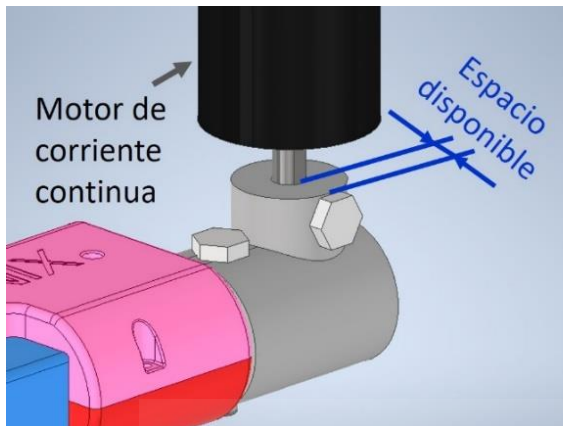


Figura 3.14: aux_lin_rot original con insuficiente espacio para el tornillo de ajuste del eje del motor

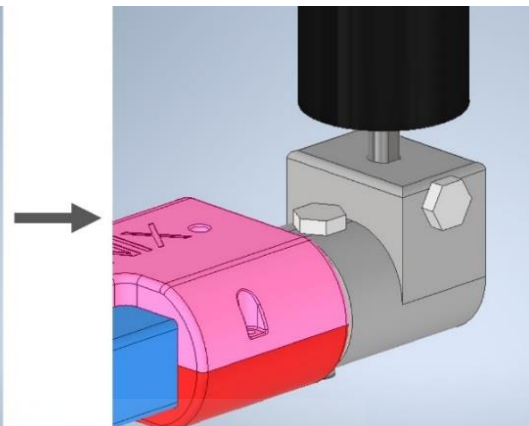


Figura 3.15: aux_lin_rot con más espacio para el tornillo de ajuste del eje del motor

No obstante, a pesar de haber realizado esta mejora, la pieza auxiliar impresa en 3D se erosionaba fácilmente en torno al eje del motor, llegando a agrandarse tanto el orificio en el que se posicionaba dicho eje, que la pieza se desprendía y se desmontaba así todo el prototipo. Se decidió entonces incluir un hexágono metálico para evitar cualquier erosión entorno al eje, el cual sería fijado por un tornillo que apretaría contra la cara plana del mismo (Figura 3.16).

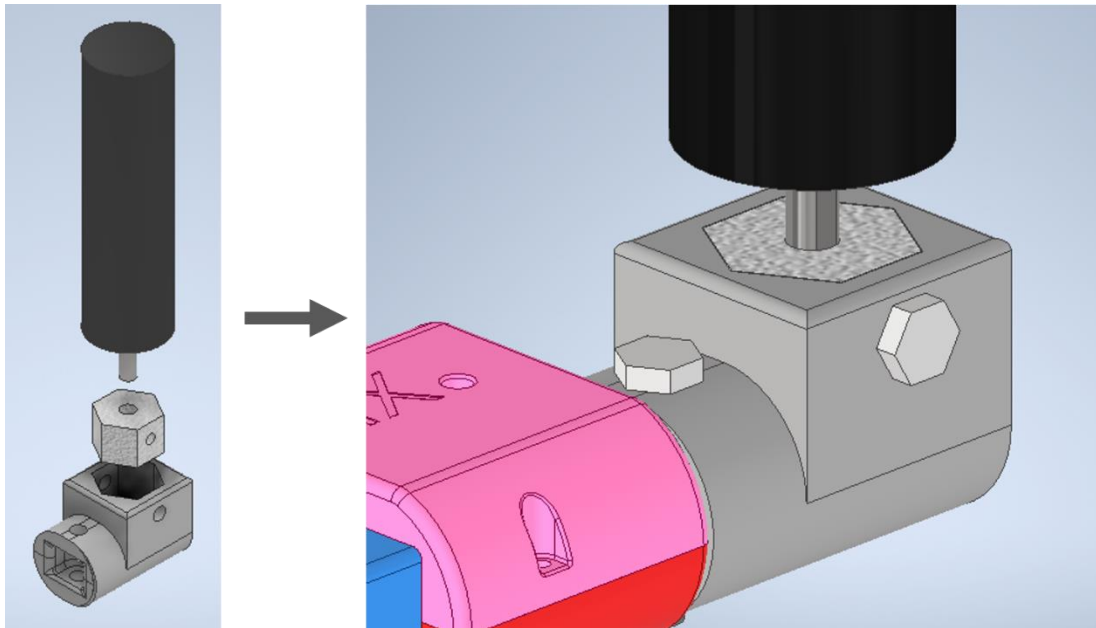


Figura 3.16: aux_lin_rot modificada para la inserción del hexágono metálico como soporte del eje del motor de corriente continua

De esta forma, el motor de corriente continua está estable y sin holguras alrededor del eje.

Teniendo todas las piezas, ya se puede ensamblar todo el prototipo. Empezando por el montaje entre los cilindros 3 y 4, se introduce una varilla roscada por el interior de los dos cilindros y, ejerciendo fuerzas opuestas en cada extremo mediante par de apriete de tuercas, se incrementa la fuerza de rozamiento entre los dos cilindros y se intenta impedir el movimiento de los mismos. No obstante, este método no funcionó como se esperaba y sí se producía un pequeño movimiento entre los cilindros que alteraba también la orientación de las hélices cilíndricas de los mismos (Figura 3.17). Para impedir este movimiento entre cilindros, se crea una pieza auxiliar (“aux_central”) intermedia entre los dos cilindros con una cavidad en forma de hexágono. También se genera un hexágono saliente en cada cilindro, los cuales se introducirán en la cavidad de la pieza auxiliar y, de esta forma, se obstruye cualquier movimiento de los cilindros en el plano XY (Figura 3.18). Aunque la orientación entre las hélices de los cilindros 3 y 4 ya no podrá variar 360° , sí se podrá variar cada 60° cambiando de posición el hexágono saliente de uno de los cilindros respecto a la pieza intermedia. Probando varias

configuraciones, se llega a la conclusión de que la opción con un mayor espacio de trabajo se produce al ubicar el final de la ranura de la hélice 3 y el comienzo de la hélice 4 de forma opuesta, como ya se muestra en la Figura 3.17. Además, para asegurar la posición central de la varilla roscada en cada cilindro, se empequeñecen los orificios centrales en los cuales se introducía la varilla roscada, pero dejando igualmente los cilindros huecos para ahorrar material y tiempo a la hora de imprimir en 3D.

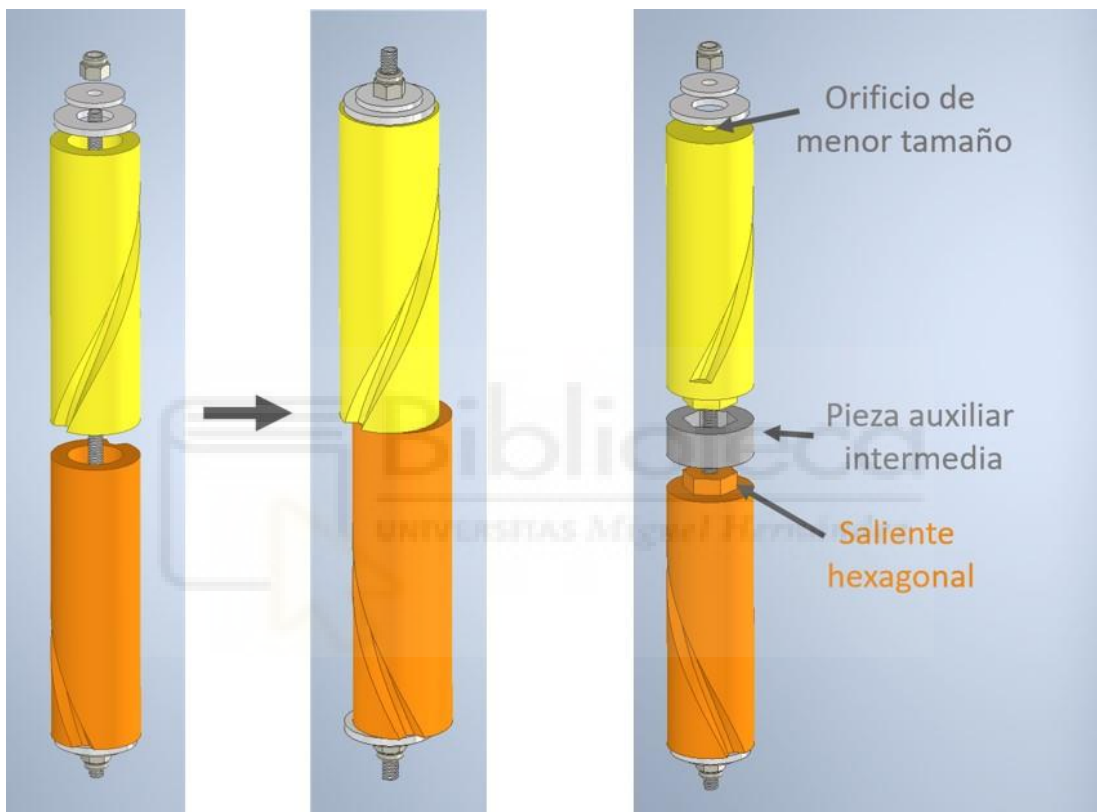


Figura 3.17: Transición de los cilindros 3 y 4 del montaje ideal (izquierda) al fallo por movimiento del cilindro 4 (derecha)

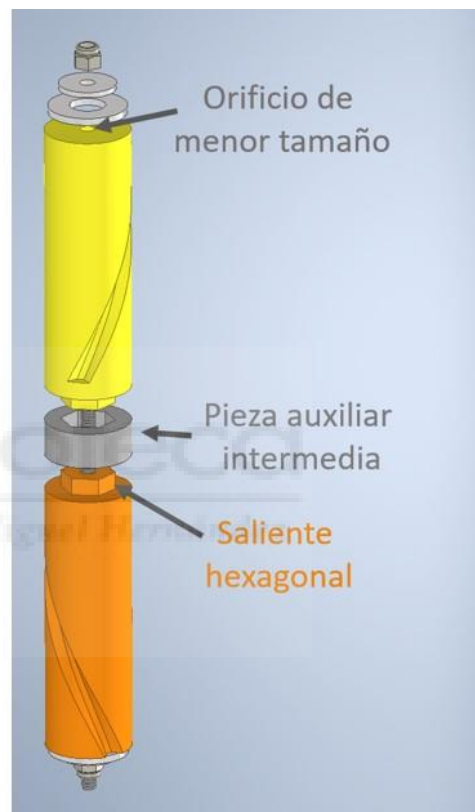


Figura 3.18: Montaje de los cilindros 3 y 4 mediante la pieza aux_central para impedir el movimiento entre cilindros

Prosiguiendo con el ensamblaje, se consigue mover manualmente el prototipo completo. Sin embargo, se producen algunas colisiones entre los actuadores (Figura 3.19) y no se puede mover todo el prototipo como se deseaba. Para evitar estas situaciones, se decide posicionar el actuador lineal 1 por encima de la articulación helicoidal 1 como se muestra en la Figura 3.20.

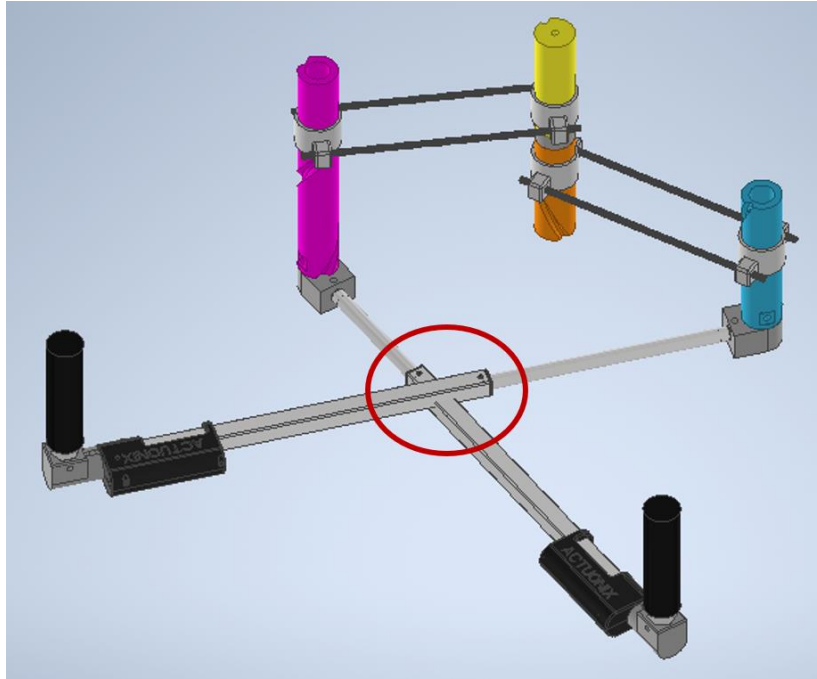


Figura 3.19: Representación de una colisión entre actuadores lineales

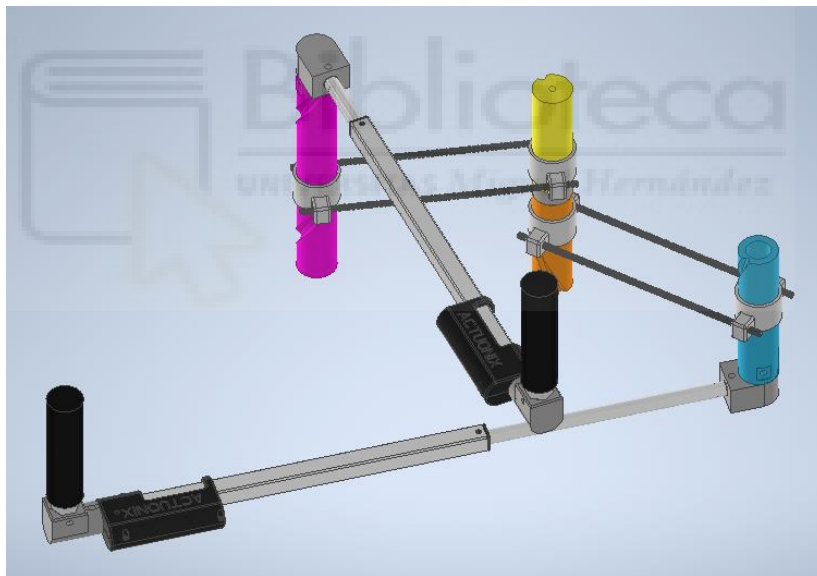


Figura 3.20: Configuración solución para las colisiones de la configuración inicial

Además, como se había predicho, el centro de gravedad del prototipo físico no está equilibrado y flexa en consecuencia, haciendo imposible el movimiento del prototipo sin una fuerza externa que lo posicione correctamente (Figura 3.21).

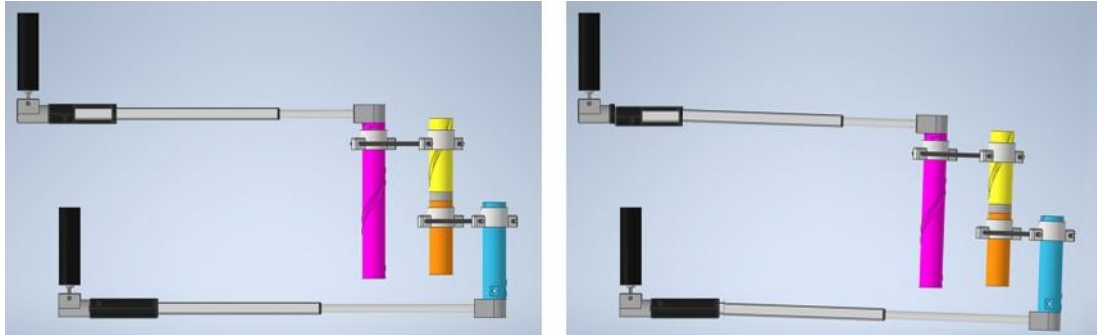


Figura 3:21: Diferencia entre el prototipo idealmente diseñado y el real

Para solucionarlo, se decide variar el diseño inicial y cambiar por completo la pieza que hace de enlace entre los actuadores lineales y rotatorios. La idea es realizar un modelo (“abrazadera”) que abrace a la pieza “extension” del actuador lineal y que también contenga al hexágono necesario para el eje del actuador rotatorio (Figura 3.22). De esta forma, la mayoría del peso del actuador lineal se ubica por detrás del apoyo del actuador rotatorio, compensándose el peso del prototipo (Figura 3.23).

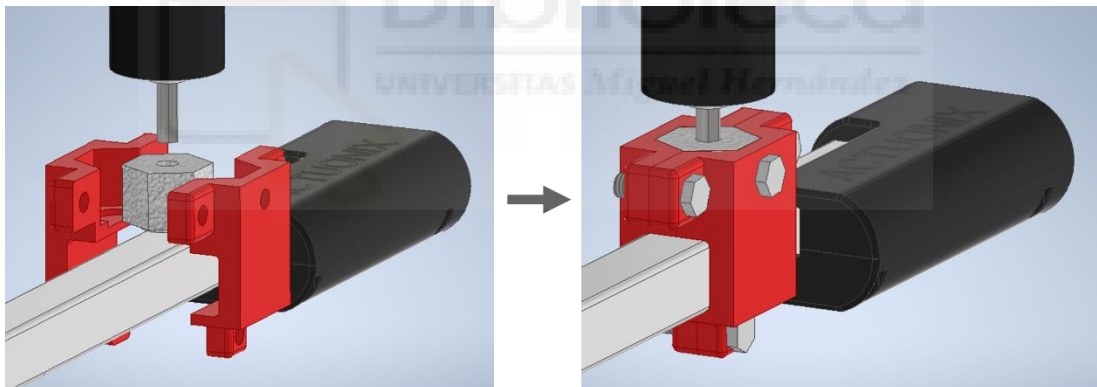


Figura 3.22: Posición en el prototipo de la pieza abrazadera

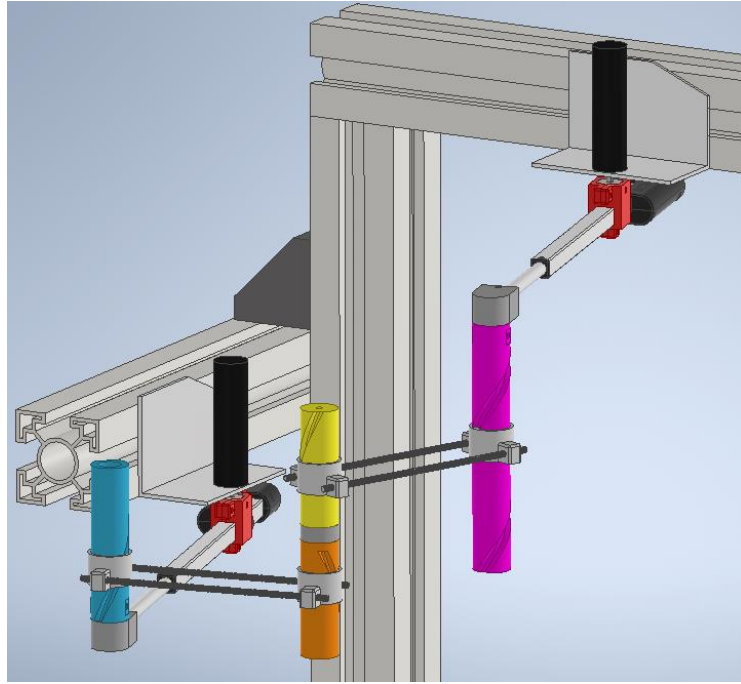


Figura 3.23: Prototipo ensamblado y equilibrado

Por otro lado, durante el ensamblaje se descubrió que los actuadores no están pensados para aguantar esfuerzos de torsión alrededor del eje del vástago (pieza “shaft”). Sin embargo, este tipo de esfuerzos sí se podrían producir durante el funcionamiento del prototipo implementado, provocando que las piezas “shaft” o “plug” se desenrosquen.

La posición final de las piezas “plug” tampoco es paralela al plano XY del prototipo (Figura 3.24), lo que genera una inclinación en las articulaciones helicoidales que dificulta el movimiento fluido del prototipo (Figura 3.25).

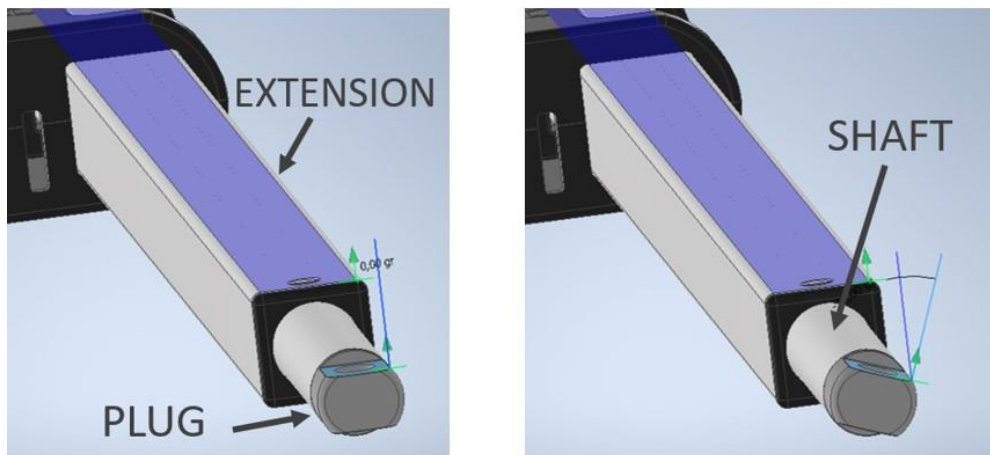


Figura 3.24: Diferencia entre la posición del plug ideal (izquierda) y real (derecha)

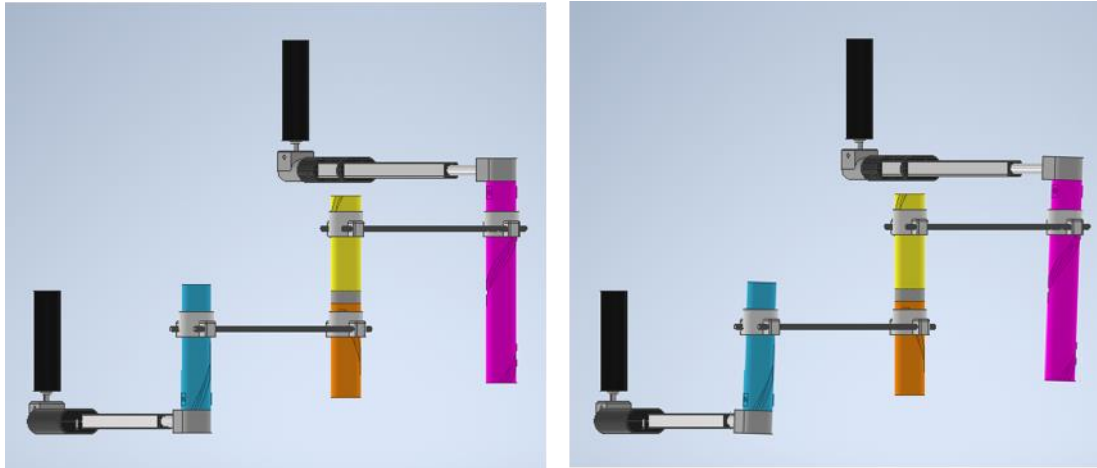


Figura 3.25: Diferencia entre la posición correcta del prototipo (izquierda) y la posición real (derecha)

Como solución, se decide usar dos actuadores lineales en cada cadena del prototipo. De esta forma, mediante otras piezas auxiliares, se garantiza que la posición y orientación de cada par de piezas “plug” sea la misma, reforzando también el resto de piezas para evitar el desacople de los actuadores lineales debido a cualquier par generado por el prototipo. Para esto se hace necesario la modificación de todas las piezas auxiliares hasta ahora realizadas.

El enlace entre cada par de actuadores con el motor de corriente continua se realiza a partir de dos piezas. Una de ellas (“aux_origen”) abraza a los dos actuadores lineales para acoplarlos y también hace de soporte del hexágono necesario para el eje del motor de corriente continua. La otra pieza (“aux_hex”) encaja en la anterior y es la que contiene al hexágono (Figura 3.26).

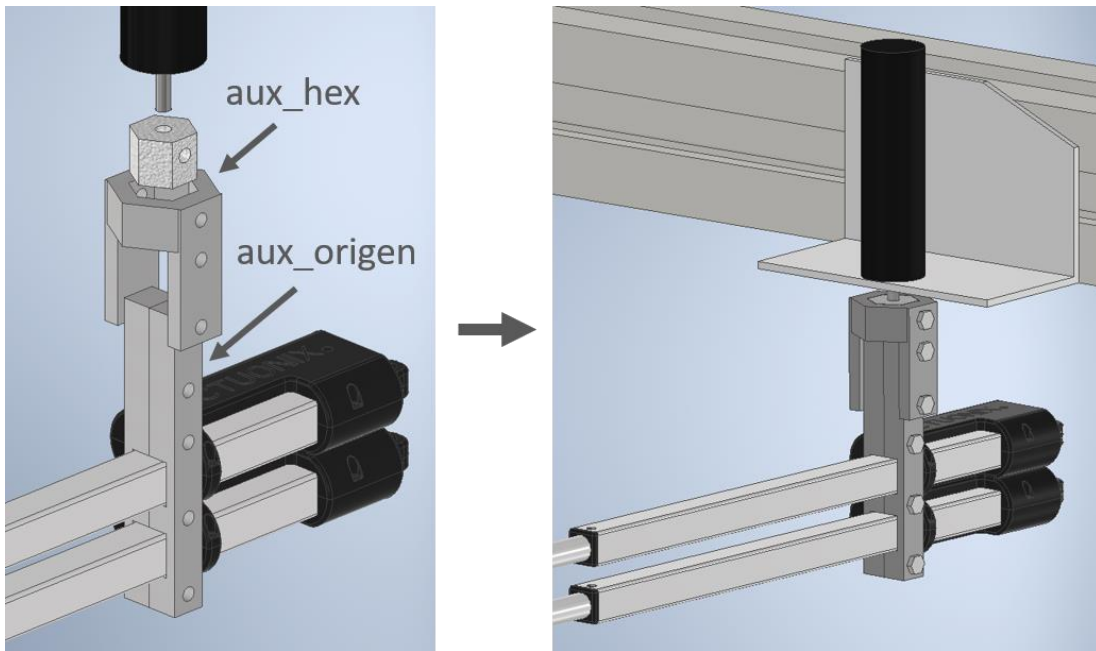


Figura 3.26: Montaje de las piezas aux_origen y aux_hex para enlazar los actuadores

Con la pieza “aux_t” se asegura que los extremos de cada par de actuadores lineales no se muevan entre sí en ningún eje. Además, esta pieza también hace la función de enlazar con los cilindros de las articulaciones helicoidales una vez se modifican éstos. Como refuerzo, se agrega otra pieza (“aux_mitad”) que abraza a los dos actuadores lineales justo al extremo de las piezas “extensión”.

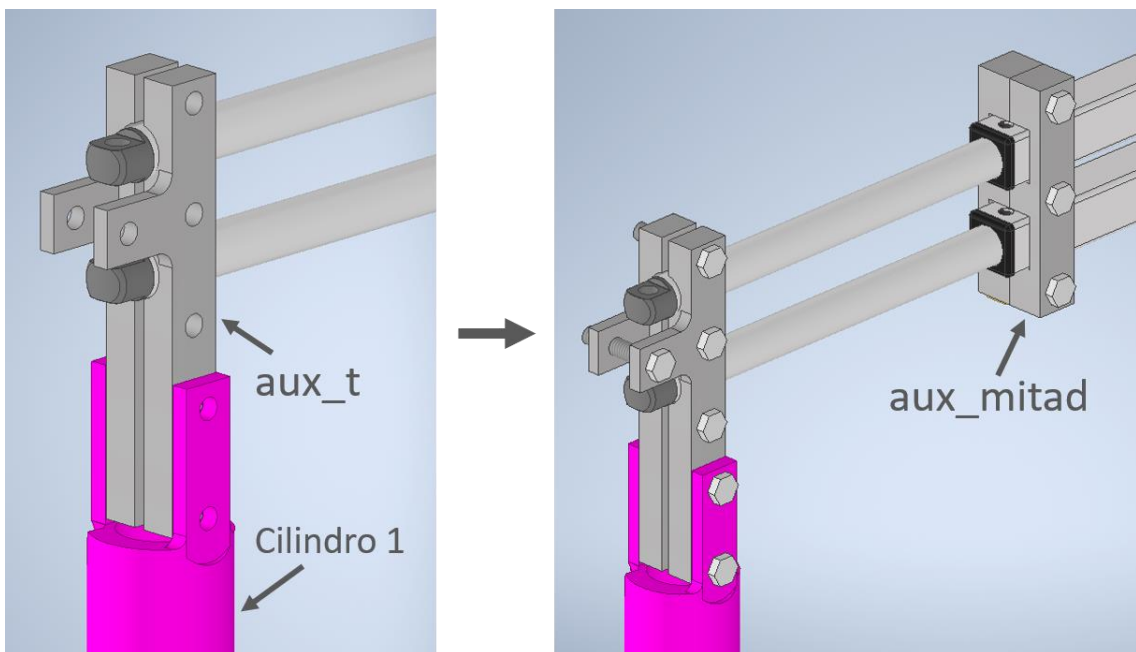


Figura 3.27: Montaje de las piezas aux_t y aux_mitad

Finalmente, ensamblando el resto de piezas, se consigue un prototipo físico más equilibrado y robusto como se muestra en la Figura 3.28.

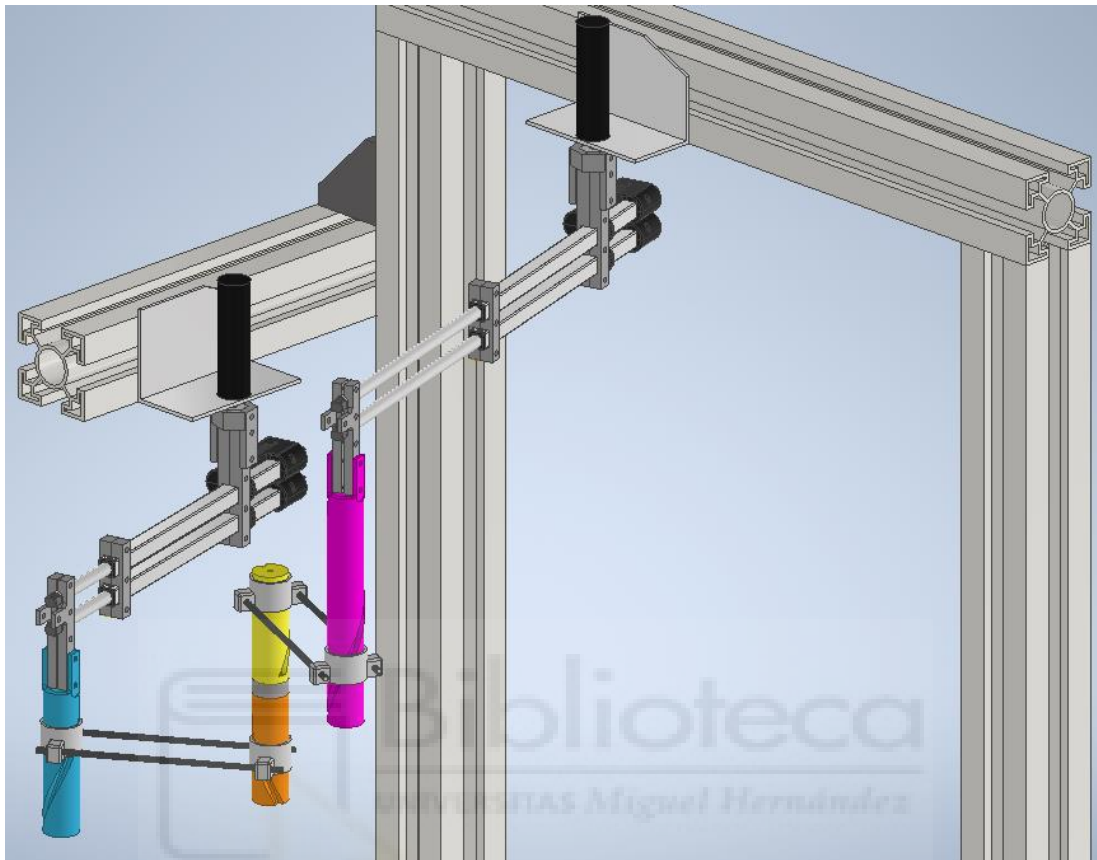


Figura 3.28: Prototipo final ensamblado

Aunque se intentó animar el prototipo ensamblado en Autodesk Inventor, el funcionamiento correcto de las articulaciones helicoidales no fue posible implementarlo. La forma más directa de efectuar este movimiento helicoidal es generando tangencia entre las hélices cilíndricas de las tuercas y cilindros. No obstante, debido al gran número de restricciones que se deben cumplir, la herramienta CAD acababa bloqueándose. Otra forma de simular este mismo movimiento sería empleando el movimiento de traslación-rotación con una pieza auxiliar. Sin embargo, el robot quedaba sobre-restringido al intentar realizar los cuatro movimientos helicoidales al mismo tiempo. Pese a todo esto, el movimiento correcto del robot se puede estudiar en el simulador de EJS que se ha descrito en capítulo 2.6.

3.3.2. IMPRESIÓN 3D

Una vez se han modelado las piezas necesarias, se imprimen en 3D. No obstante, este paso está condicionado por muchas variables.



Figura 3.29: Impresiones fallidas al imprimir un cilindro de las articulaciones helicoidales

Como se puede apreciar en la Figura 3.29, en las primeras impresiones aparecen varios errores que provocan una falta de solidez en las piezas, llegando incluso a producirse la rotura de éstas.

Lo primero a reajustar es la calibración de la placa de impresión. Este es el procedimiento que se emplea para dejar la base de la impresora a la distancia idónea de la boquilla de la impresora, por donde se extruye el material. En caso de realizarse a una distancia mayor, el material extruido no se adhiere a la placa de impresión. Si la distancia es menor, el trazo del camino realizado se deforma y la altura de cada capa disminuye. Además, dado que no se extruye todo el material que la impresora había calculado extruir, se pueden ocasionar bolas de material en las capas o atascos en el interior de la boquilla.

La forma más fácil de realizarlo es utilizando una hoja de papel convencional, las cuales rondan un grosor de 1 mm. Este método consiste en desplazar la boquilla de la impresora por los puntos más importantes de la placa de impresión, como los puntos cercanos a los bordes, y hacer pasar la hoja entre la placa y la boquilla, variando la presión en esos puntos hasta conseguir notar que se roza la hoja y se produce una pequeña dificultad al pasar. Sin embargo, este método no mejoró los problemas al probarlo. Otra forma es utilizar un

reloj comparador centesimal para que mida con precisión las cotas entre la placa y la boquilla en cada esquina de la placa. Con este método tampoco se mejoraron los resultados, de hecho, empeoraron. Sin embargo, resultaba ser que la placa de vidrio utilizada como base de impresión no era completamente plana, estaba deformada y formaba un valle en el centro que a simple vista no se apreciaba. Sin embargo, una tercera forma de realizar la calibración era ajustando las cotas durante la impresión. Imprimiendo una sola capa del diseño que se muestra en la Figura 3.30 y variando manualmente la distancia entre la base de impresión y la boquilla según los resultados, se consigue una buena calibración.

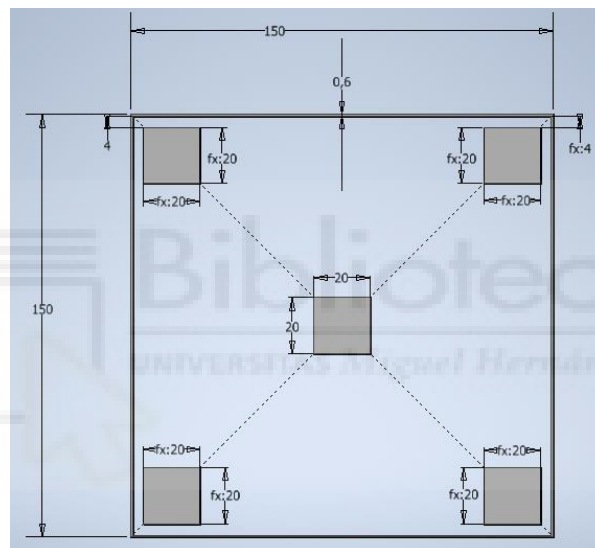


Figura 3.30: Diseño con cuadrados y borde exterior impreso en la zona central de la base para una correcta nivelación

Si los cuadrados generados quedan con las líneas de relleno separadas, significa que la posición de la boquilla respecto a la base es muy alta. En este caso, puede ser que el cuadrado pierda su forma y el relleno se rompa con facilidad al manipularlos.

Si los cuadrados quedan con una superficie rugosa, significa que la boquilla está muy cerca de la placa y, al extruir una línea, alcanza a tocar la línea anterior, calentándola de nuevo y causando una elevación del plástico entre líneas. En este caso, los cuadrados no pierden la forma y quedan resistentes, pero el acabado superficial no es el adecuado.

Si los cuadrados quedan con una superficie suave y mantienen su forma sin romperse, la nivelación es correcta.

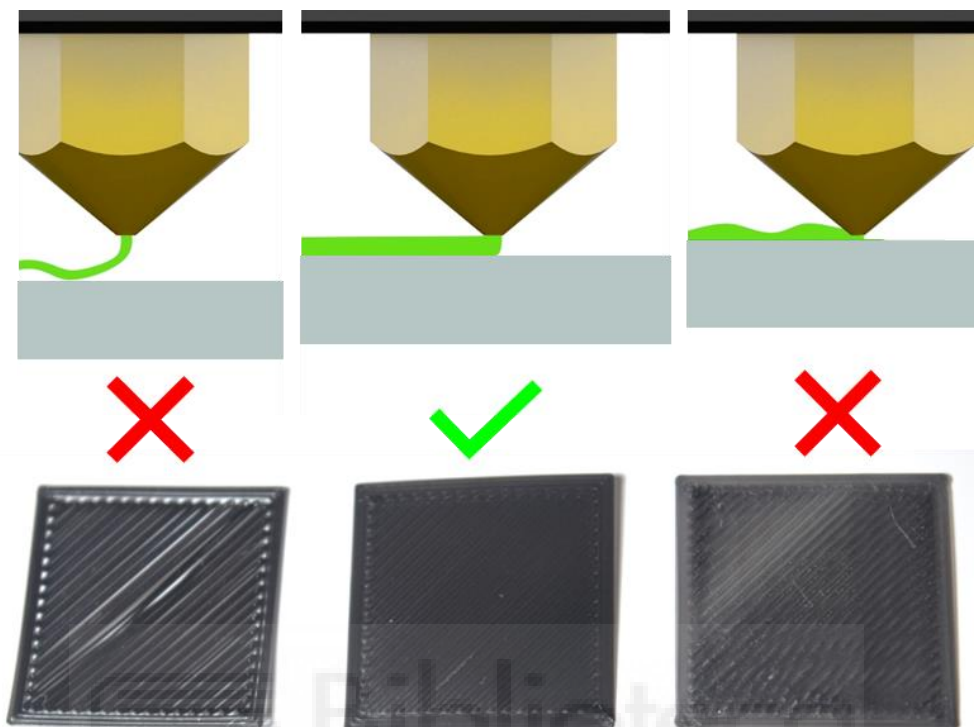


Figura 3.31: Comparativa entre cuadrados según la cota de la boquilla a la placa

Este último método es el ideal ya que, al tener solo en cuenta la zona central de la base al imprimir, que es la zona en la que se empezarán a imprimir las piezas, no resulta tan relevante la deformación de la base.

Tras conseguir que todos los cuadrados tuvieran la apariencia y resistencia correctas, se probó a imprimir de nuevo una pieza del prototipo, pero los resultados no eran los esperados (Figura 3.32).



Figura 3.32: Impresión de la tuerca de una articulación helicoidal después de la calibración

En la Figura 3.32 se aprecia cómo la unión entre las capas no es la adecuada, causando incluso la rotura de la pieza. Sin embargo, dado que la calibración manual estaba bien hecha, el problema tendría que originarse en los valores de los parámetros de impresión utilizados en el laminador o *slicer* utilizado (Ultimaker Cura [14]). En un principio, los parámetros de impresión variados fueron:

- Temperatura

- Temperatura de impresión

Es la temperatura de la boquilla durante la impresión o extrusión de material. El fabricante del filamento utilizado aconsejaba emplear temperaturas entre 200°C y 220°C, sin embargo, tras varias pruebas, el único cambio entre temperaturas bajas y altas era el brillo final de las piezas. A 217°C la pieza impresa era más brillante que a 200°C. El valor utilizado para imprimir fue 190°C.

- Temperatura de la placa de impresión

Se define como la temperatura de la cama caliente o base de la impresora, que facilita la adherencia de las piezas a la misma, evitando así que las piezas se despeguen o se curven.

El valor más utilizado es 60°C, pero se averiguó que la pieza quedaba perfectamente adherida a la superficie de impresión con 65°C.

- Velocidad de impresión

Las velocidades recomendadas para la impresora utilizada son entre 50 mm/s y 60 mm/s, por lo que se estableció el valor medio a 55 mm/s.

- Velocidad de impresión en la primera capa

Disminuir la velocidad en la primera capa es otra forma de que el material esté más tiempo de contacto en la boquilla, llegue a la temperatura adecuada y tenga mejor fluidez. Lo recomendable es establecer este parámetro al 50% del valor de la velocidad de impresión, es decir, a 27.5 mm/s.

- Relleno

La densidad de relleno define la cantidad de plástico utilizado en el interior de la impresión. Una mayor densidad de relleno significa que hay más plástico en el interior de la impresión, lo que conduce a un objeto más pesado y fuerte. Una densidad de relleno baja o nula se utiliza para modelos con un propósito visual.

- Patrón

Existen varios patrones de relleno en Cura y cada uno es preferible usarlo en comparación con otro dependiendo de la resistencia que se desee. Dado que se necesitan unas piezas no demasiado pesadas pero robustas, se establece el patrón de relleno cúbico con una densidad de entre un 20% y un 40%, dependiendo de la pieza impresa.

- Diámetro de boquilla

Cuanto mayor es el diámetro de la boquilla que extruye el material, menor es el detalle final de las piezas y el tiempo empleado. El valor utilizado era 0.4 mm.

- Altura de capa

Es el alto de cada capa impresa en el eje Z. El valor utilizado era de 0.4 mm.

Tras realizar pruebas con distintos valores de temperatura sin conseguir mejorar los acabados, se realizaron pruebas cambiando el diámetro de la boquilla y la altura de capa con los valores de 0.6 mm y 0.2 mm. Como no se obtuvieron buenos resultados, se volvió al valor de 0.4 mm para generar una precisión media. Sin embargo, el error estuvo en pensar que el diámetro de la boquilla y la altura de capa debían coincidir en valor. Hay que tener en cuenta que parte del material se emplea para la unión entre capas, por lo que lo recomendable es establecer la altura de capa al 50% del diámetro de la boquilla. De esta manera se evitan las estrías en la impresión causadas por la mala adherencia entre capas. Por tanto, los valores correctos son 0.4 mm para el diámetro de boquilla y 0.2 mm para la altura de capa.

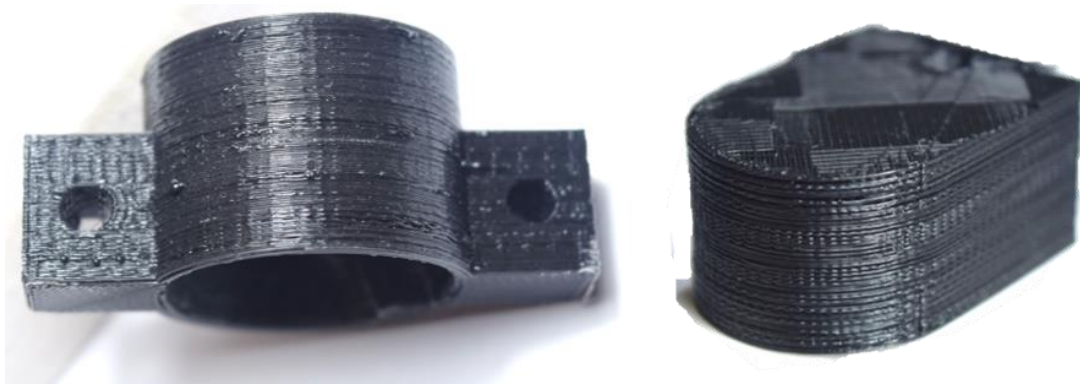


Figura 3.33: Impresiones de una tuerca y la pieza aux_act_cil tras variar los parámetros de temperatura, velocidad, relleno, diámetro de la boquilla y altura de capa

Aunque se mejoró la robustez de las piezas, se seguía sin tener un buen acabado superficial. Se modificaron otros parámetros:

- Flujo

Es la cantidad de material que se extruye durante un período de tiempo específico y se basa en el diámetro del filamento y la velocidad de impresión. El valor por defecto es del 100%. Para ajustar este parámetro se realiza la prueba del cubo. Esta prueba consiste en imprimir un cubo hueco, únicamente con dos paredes, y sin la cara superior, para comprobar que el espesor de las caras laterales se corresponde con las del modelo 3D diseñado (Figura 3.34). Utilizar un diámetro de boquilla de 0.4 mm generará líneas con un grosor de 0.4 mm y 0.2 mm de altura, por lo que las dos paredes deberán tener un grosor de 0.8 mm. Sin embargo, una vez impreso el cubo hueco y habiendo medido el grosor de las dos paredes con un calibre digital, se comprueba que el grosor es de 0.94 mm, mayor al esperado. En consecuencia, el valor del flujo debe ser disminuido para conseguir precisión en las medidas.

Flujo correcto

$$= \frac{\text{Flujo inicial} * \text{Grosor de la cara lateral del cubo en el diseño}}{\text{Grosor de la cara lateral del cubo impreso}}$$

$$\text{Flujo correcto} = \frac{100\% * 0.8 \text{ mm}}{0.94 \text{ mm}} \rightarrow \text{Flujo correcto} = 85\%$$

Al volver a imprimir el cubo con el nuevo valor del flujo (85%), se comprueba como la precisión ha mejorado, siendo el nuevo grosor de las paredes 0.82 mm (Figura 3.35).

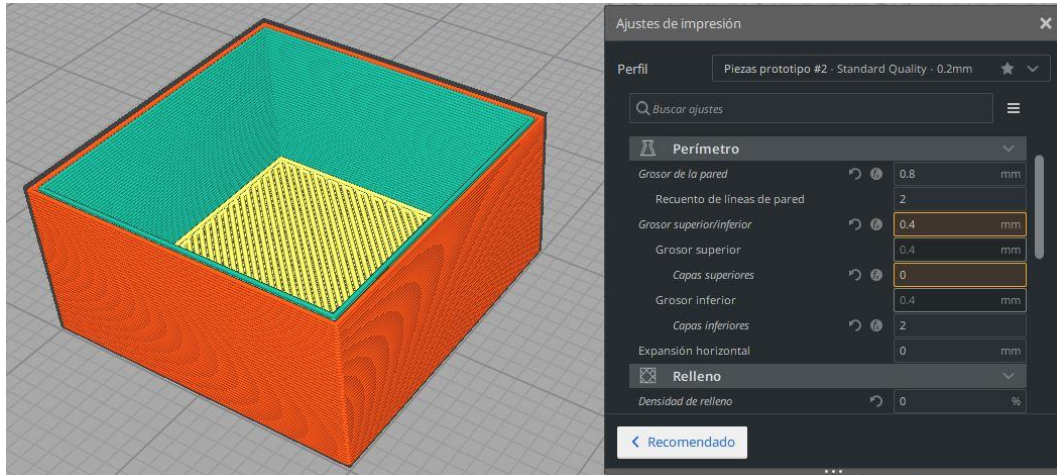


Figura 3.34: Diseño de impresión para la prueba del cubo



Figura 3.35: Comprobación de los grosores antes (izquierda) y después del ajuste del flujo de material a extruir (derecha)

- Alineación de costuras en Z

Esta configuración permite elegir dónde comienza cada nueva capa en la dirección Z y afecta dónde estará la costura del modelo. Esto es útil para modelos con capas iguales consecutivas ya que la costura puede ser visible. Hay varias opciones de configuración, pero la seleccionada es la denominada “esquina más pronunciada”, que posiciona la costura en la esquina más nítida (Figura 3.36).



Figura 3.36: Visualización de las costuras en las piezas (rojo) y la posición nueva de las costuras (azul) tras la configuración de la alineación de costuras en Z

Modificando estos últimos parámetros se consiguió suavizar y homogeneizar las paredes exteriores de las piezas (Figura 3.37). Además, las medidas de las piezas se correspondían mucho más a los valores del diseño 3D.



Figura 3.37: Impresión del cilindro de una articulación helicoidal tras modificar los parámetros del flujo y las costuras

No obstante, se presentó otro problema que también se muestra en la Figura 3.37. El encordado o *stringing* es la aparición de pequeños hilos de material que quedan cuando se imprimen partes de la pieza que están separadas, por ejemplo, huecos o agujeros. Cuando la boquilla va de un lado a otro y, aunque el motor del extrusor de la impresora no empuje al filamento, se va drenando material y dejando rastro del desplazamiento efectuado debido a que la

boquilla sigue estando caliente. Para solventar esta cuestión se analizan los elementos que intervienen al realizar la retracción del filamento, que se refiere al acto de retirar o retroceder el filamento en los momentos en que la boquilla se mueve sobre una zona que no debe ser impresa.

Distancia de retracción (mm)	Velocidad retracción (mm/s)	Velocidad de desplazamiento (mm/s)
5.2 – 5.6	40-70	120-180

Tabla 3.1: Valores de los parámetros recomendados para la impresora CR-10s

- **Distancia de retracción**

Es la distancia en milímetros que el material se retrae de la boquilla para contrarrestar la presión positiva que se produce en la misma. Una retracción prolongada crea más tensión en el material, por lo que aumenta el tiempo empleado y disminuye la cantidad de material de salida, pudiendo incluso llegar a deformar el flujo y disminuir el ancho de extrusión en el lugar donde verdaderamente se tendría que posicionar material. En cambio, una retracción corta tiene una mayor probabilidad de exudación, pero mantiene el material seguro y el tiempo de impresión es menor. El valor más adecuado resultó ser 5.8 mm.

- **Velocidad de retracción**

Velocidad a la que se retrae el filamento y se prepara para un movimiento de retracción. El valor establecido es 60 mm/s.

- **Velocidad de desplazamiento**

Velocidad a la que se producen los movimientos de desplazamiento. El valor establecido es 120 mm/s

- **Retracción en el cambio de capa**

Se marca la casilla para obligar a la impresora a retraer el filamento antes de comenzar a imprimir la siguiente capa.

Estos valores son los que mejor resultado han dado haciendo pruebas con la pieza mostrada en la Figura 3.38, diseñada para destacar este efecto de los hilos con mayor claridad.

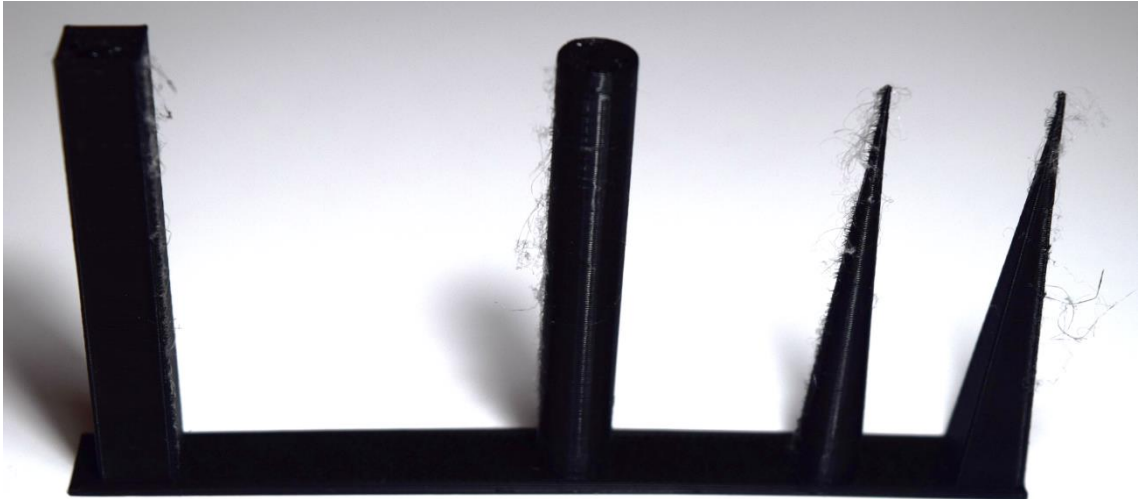


Figura 3.38: Pieza final en la que los hilos son de un diámetro muy pequeño y fácilmente eliminables de las piezas



Figura 3.39: Acabado superficial de una tuerca después de realizar la calibración y de variar todos los parámetros explicados

Finalmente, como se puede ver en la Figura 3.39, las piezas impresas con la impresora CR-10S consiguen una buena apariencia y robustez.

Todo lo anterior es extrapolable a la impresora CTC Dual. Sin embargo, dado que esta impresora utiliza su propio programa laminador, hay varios

parámetros que no son modificables. Esto no es inconveniente dado que los valores de fabricación son correctos y solo se ha tenido que realizar la calibración de la placa de impresión y ajustar la temperatura, la velocidad, el diámetro de la boquilla y la altura de capa, utilizando los mismos o similares valores a los de la CR-10S.

3.4. PROTOTIPO FINAL

El prototipo físico con las piezas finales se muestra en las Figuras 3.40 y 3.41.



Figura 3.40: Prototipo físico ensamblado



Figura 3.41: Prototipo físico ensamblado desde una vista más cercana

3.5. FUNCIONAMIENTO

Para comprobar el funcionamiento del prototipo, se realiza el control en bucle abierto, aplicando tensión a cada actuador.

El movimiento es el esperado, pero se advierten dos problemas:

- I) El prototipo aún flexa. Esto se debe a que el centro de gravedad se encuentra bastante alejado de los puntos de apoyo de los motores rotatorios.
- II) El movimiento está muy limitado. Aunque se ha mejorado el espacio de trabajo, éste sigue siendo muy reducido y restringido en ciertas zonas.

En cuanto al primer problema, debido al diseño funcional de las piezas auxiliares utilizadas, este se puede arreglar fácilmente desplazando las piezas “aux_origen” hacia adelante para equilibrar aún mejor el centro de masas y reducir la flexión de las piezas. Manteniendo constante la carrera de los actuadores, con esta acción lo que se hace es disminuir q_{3_min} y q_{4_min} . Y, como se ha visto en el análisis de sensibilidad, al disminuir estos parámetros geométricos, el espacio de trabajo del robot aumenta. Por tanto, esta maniobra es necesaria y beneficiosa para el funcionamiento del prototipo.

Para el segundo problema, en cambio, es necesario un estudio más exhaustivo que el análisis de sensibilidad realizado. Este nuevo análisis se explica en el siguiente capítulo, que busca optimizar el diseño del robot para maximizar su espacio de trabajo.



4. OPTIMIZACIÓN DEL DISEÑO

En el capítulo 3 se ha construido un prototipo inicial para validar el concepto del robot diseñado. Sin embargo, dicho prototipo no tiene una gran movilidad debido a que sus dimensiones no se eligieron en base a cálculos óptimos sino en base a simulaciones preliminares hechas con EJS. Por tanto, es necesario optimizar el diseño para maximizar su espacio de trabajo.

Esto se hará utilizando el algoritmo matemático de optimización “simulannealbnd” o recocido simulado que implementa la herramienta de cómputo numérico Matlab [15]. Este algoritmo buscará los valores óptimos para los parámetros geométricos $(d, b_1, b_2, z_{01}, z_{02}, c_{10}, c_{20}, \lambda_1, \lambda_2, \lambda_3, \lambda_4)$ que optimizan el espacio de trabajo. No obstante, como todos los comandos de optimización de Matlab, este algoritmo no maximiza, sino que minimiza. Pero maximizar el espacio de trabajo compuesto por n puntos es equivalente a minimizar $-n$ puntos, por lo que sólo habrá que pasarle este valor en negativo. Este comando se ejecutará de la siguiente manera:

```
diseño_optimo = simulannealbnd(@esp_trabajo, diseño_inicial, minimos, maximos);
```

Donde:

- *diseño_optimo* es el vector con los valores óptimos de los parámetros geométricos obtenidos como resultado de la optimización.
- *esp_trabajo* se refiere al nombre de la función que retorna $-n$ puntos que componen el espacio de trabajo estudiado.
- *diseño_inicial* es el vector de parámetros geométricos de un diseño considerado inicial. Los valores de este vector influirán sobre el resultado óptimo, siendo posible incluso obtener distintos valores óptimos para un mismo diseño inicial.
- *minimos* y *maximos* son los vectores que contienen los valores mínimos y máximos admisibles de los parámetros geométricos. Estas restricciones se imponen para evitar diseños extremadamente grandes o pequeños, que puedan ser incluso imposibles de realizar. No se incluyen los parámetros de diseño q_{3_min} , q_{3_max} , q_{4_min} , q_{4_max} en estos vectores ya que sus valores

están condicionados por las medidas de los actuadores lineales utilizados y sus carreras. Los valores establecidos son:

	b_1 (m)	b_2 (m)	λ_1 ($\frac{m}{rad}$)	λ_2 ($\frac{m}{rad}$)	λ_3 ($\frac{m}{rad}$)	λ_4 ($\frac{m}{rad}$)	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
<i>minimos</i>	0.1	0.1	-0.4	-0.4	-0.4	-0.4	-0.083	-0.22	-0.14	0.03	0
<i>maximos</i>	0.4	0.4	0.4	0.4	0.4	0.4	0.057	-0.14	-0.03	0.14	0.4

Tabla 4.1: Valores elegidos para los vectores *minimos* y *maximos*

Estos valores se eligieron teniendo en cuenta varios aspectos.

- Longitud de las varillas pasivas (b_1, b_2)

Al representar distancias, el valor mínimo admisible no podrá ser negativo. Sin embargo, tampoco podrán tener un valor nulo ya que enlazan articulaciones helicoidales con un diámetro preestablecido. Por tanto, teniendo en cuenta el análisis de sensibilidad que refleja comportamientos opuestos al intentar mejorar el espacio de trabajo, se fija un valor mínimo de 0.1 m. Para limitar el largo del robot, como máximo se aceptan 0.4 m.

- Pasos de cada articulación helicoidal ($\lambda_1, \lambda_2, \lambda_3, \lambda_4$)

Cuanto mayor es el paso establecido en valor absoluto, mayor es el avance en el eje Z con un giro pequeño de la articulación. Por el contrario, cuanto menor sea el paso en valor absoluto, menor será el avance en el eje Z y mayor giro será necesario efectuar para variar este avance. Sin embargo, como se ha podido comprobar con el prototipo construido, fijar valores absolutos grandes para todos los pasos de las articulaciones helicoidales no asegura un avance rápido en el eje Z del efector final con giros pequeños en las articulaciones. Por otro lado, como ya se había comentado, cuanto más grande sea el paso, más rectilínea se vuelve la hélice cilíndrica de la articulación y más complicado de realizar se vuelve el movimiento del robot. Para averiguar los valores mínimo y máximo se llevan a cabo algunas pruebas con el

modelo 3D del cilindro de una articulación helicoidal y se decide el valor absoluto de 0.4 m/rad.

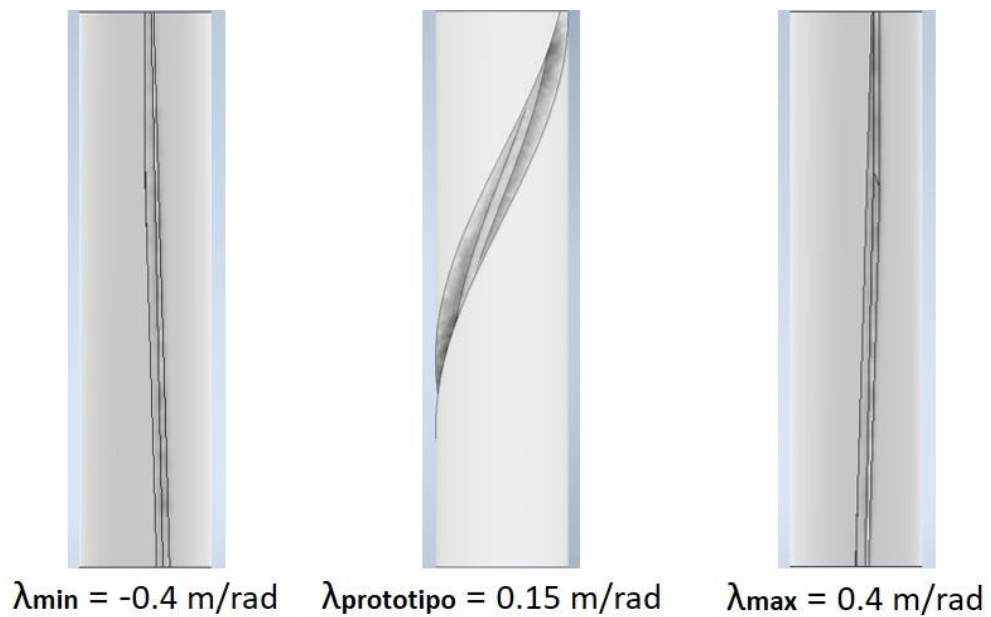


Figura 4.1: Comparativa entre los valores de los pasos de las articulaciones helicoidales mínimo, el utilizado para el prototipo construido y el máximo

- Alturas iniciales de las tuercas 1 y 2 (z_{01}, z_{02})

Para la elección de estos valores se decide fijar las alturas de las articulaciones helicoidales H_1 y H_2 con las mismas medidas del prototipo (0.16 m). De esa forma, dado que se emplearían las mismas piezas auxiliares para el robot optimizado, los valores máximos y mínimos de estos parámetros geométricos son los que se muestran en la Figura 4.2.

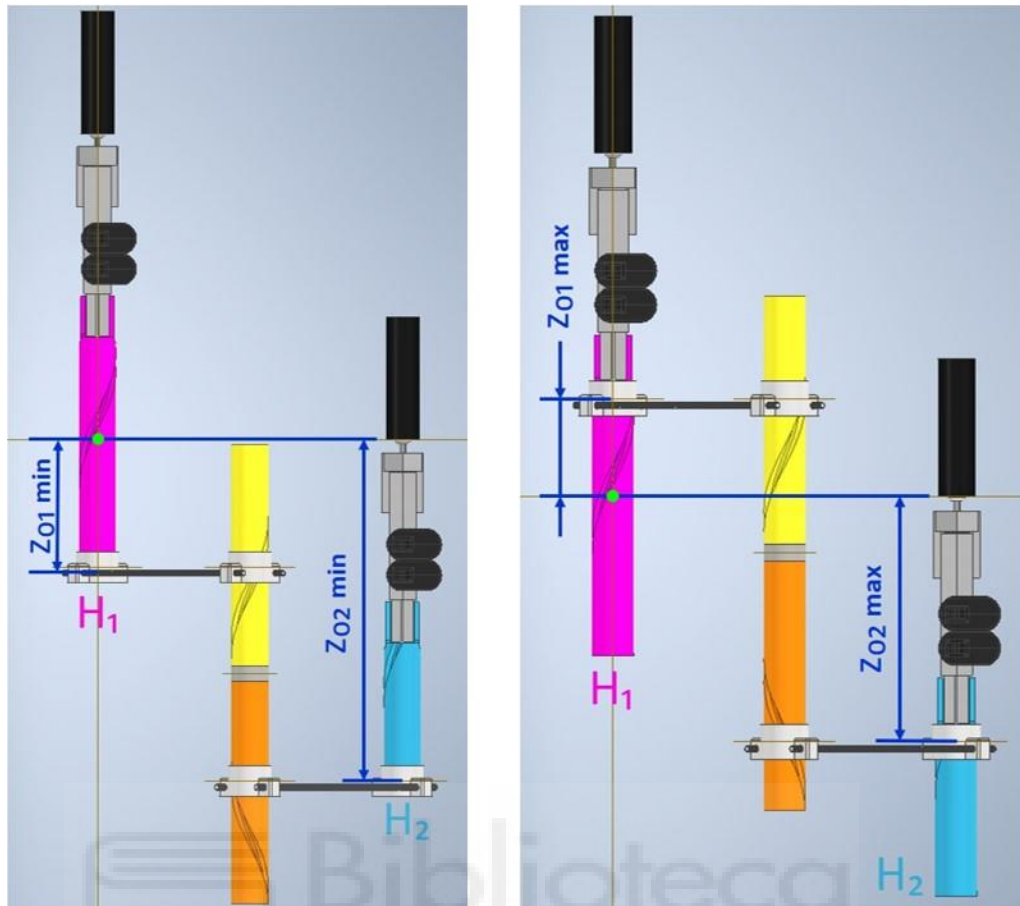


Figura 4.2: Valores mínimos y máximos de z_{01} y z_{02} tomando como origen el punto verde de acuerdo con el esquema de la Figura 2.1

- Alturas iniciales de las tuercas 3 y 4 respecto al efector final (c_{10}, c_{20})
 Estos valores están limitados por las distancias mínima y máxima que puede haber entre las tuercas 3 y 4, las cuales se producen en los límites de z_{01} y z_{02} . La distancia mínima (0.06 m) se produce entre z_{01_min} y z_{02_max} , mientras la distancia máxima (0.28 m) se produce entre z_{01_max} y z_{02_min} , como se muestra en las Figuras 4.3 y 4.4. De esta manera, estas alturas mínimas y máximas, utilizando la misma pieza auxiliar intermedia entre los cilindros, son las que se muestran también en las Figuras 4.3 y 4.4.

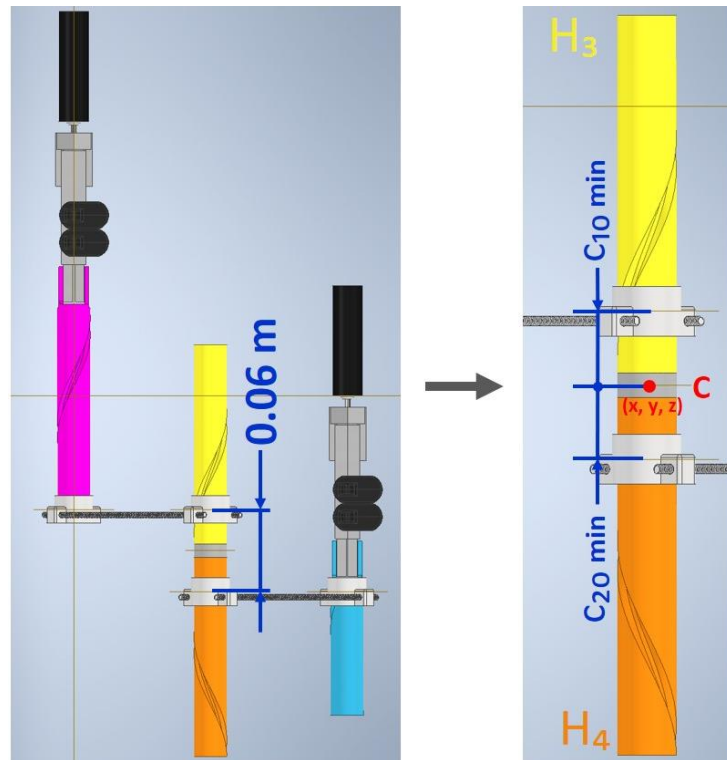


Figura 4.3: Valores mínimos de c_{10} y c_{20} respecto del efector final (punto C rojo)

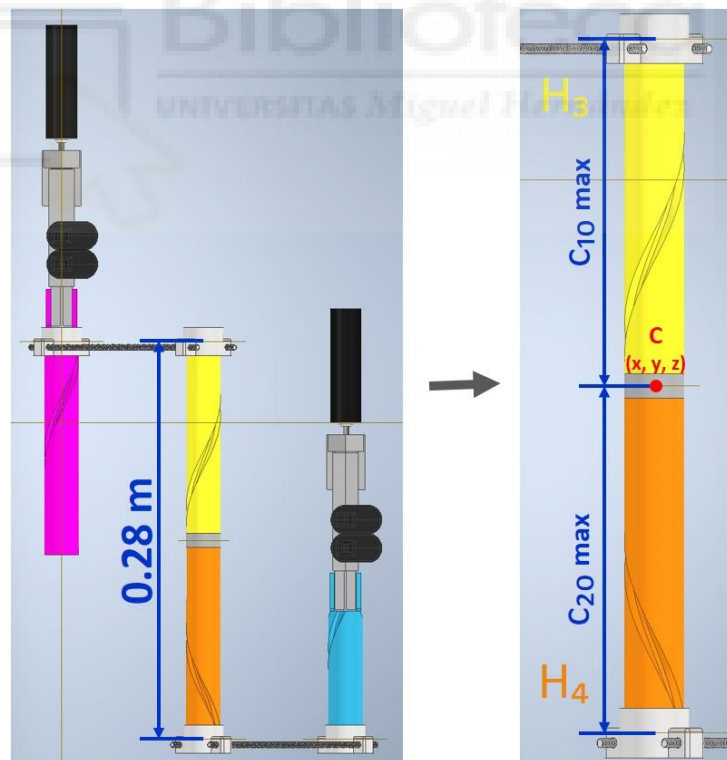


Figura 4.4: Valores máximos de c_{10} y c_{20} respecto del efector final (punto C rojo)

- Distancia entre apoyos de los actuadores lineales en el eje X (d)
Al ser una distancia, su valor no debe ser negativo, por lo que se establece un mínimo nulo. Como medida máxima se establece 0.4 m.

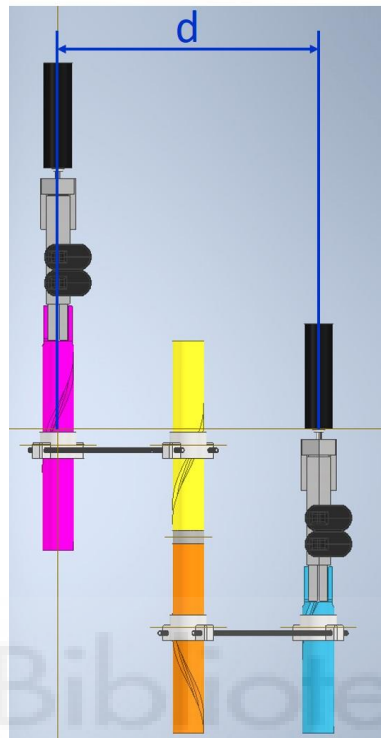


Figura 4.5: Representación de la distancia d

Teniendo la información anterior en cuenta, las líneas que se ejecutarán en la ventana Command Window de Matlab antes de ejecutar el algoritmo de optimización recocido simulado serán:

```
minimos = [0.1, 0.1, -0.4, -0.4, -0.4, -0.4, -0.083, -0.22, -0.14, 0.03, 0];
```

```
maximos = [0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.057, -0.14, -0.03, 0.14, 0.4];
```

De esta forma, solo habrá que pasarle los nombres de los parámetros de entrada del algoritmo *simulannealbnd* cuando se ejecute.

En este estudio se realizan seis pruebas con los mismos seis diseños iniciales para cada espacio 2D y 4D. Los valores utilizados son:

	b_1 (m)	b_2 (m)	λ_1 $(\frac{m}{rad})$	λ_2 $(\frac{m}{rad})$	λ_3 $(\frac{m}{rad})$	λ_4 $(\frac{m}{rad})$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
1	0.344	0.372	-0.298	0.331	0.106	-0.322	-0.080	-0.176	-0.024	0.117	0.071
2	0.391	0.387	-0.012	0.240	-0.287	-0.063	-0.020	-0.157	-0.124	0.086	0.016
3	0.400	0.400	0.143	-0.206	0.395	0.186	-0.056	-0.146	-0.049	0.023	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Tabla 4.2: Valores elegidos de los diseños iniciales para la optimización del espacio de trabajo 2D y 4D

Los códigos implementados para la optimización del espacio de trabajo se encuentran en el [Anexo 3](#).

4.1. MAXIMIZACIÓN DEL ESPACIO DE TRABAJO 2D

Como paso previo a la maximización del espacio de trabajo 4D, se genera el código que maximiza el espacio de trabajo en 2D. En el [Algoritmo 1](#) del capítulo 2.4 se muestra la forma de hallar el espacio de trabajo 4D. Sin embargo, al igual que se hizo en EJS, en este código sólo se emplearán las líneas que varían (x, y) , manteniendo z y φ constantes. De esta manera, se podrá comprobar si el código implementado en Matlab genera el mismo espacio de trabajo que el código en EJS. También será más intuitiva la interpretación gráfica del resultado obtenido visualizando un espacio 2D que uno 4D. Además, la ejecución del algoritmo de optimización emplea bastante menos tiempo ya que solamente debe recorrer los ejes XY. Esto permite realizar las ejecuciones más rápidamente y poder perfeccionar el estudio antes de pasar a optimizar el espacio 4D; que es más exigente computacionalmente.

Una vez se ha conseguido transcribir las líneas necesarias del [Algoritmo 1](#) a Matlab y constatar el correcto funcionamiento, se prosigue a verificar la maximización del espacio de trabajo con el algoritmo de optimización *simulannealbnd*.

Se realizan las seis pruebas de maximización del espacio de trabajo en 2D con los diseños iniciales comentados anteriormente, y se estudia un único caso en el que $z = \varphi = 0$. Los resultados óptimos se muestran en la Tabla 4.3.

diseño_optimo2D = simulannealbnd(@esp_trabajo2d,diseño_inicial,minimos,maximos);

	b_1 (m)	b_2 (m)	λ_1 $\left(\frac{m}{rad}\right)$	λ_2 $\left(\frac{m}{rad}\right)$	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
1	0.356	0.355	-0.390	0.325	0.396	-0.368	0.015	-0.166	-0.066	0.137	0.000
2	0.357	0.357	0.335	0.282	-0.306	-0.291	0.043	-0.184	-0.115	0.122	0.000
3	0.356	0.356	0.314	0.325	0.247	-0.135	0.046	-0.153	-0.082	0.112	0.000
4	0.363	0.361	-0.352	-0.352	-0.033	0.333	0.008	-0.192	-0.094	0.124	0.000
5	0.357	0.355	-0.379	0.315	0.343	-0.353	-0.018	-0.144	-0.054	0.119	0.000
6	0.355	0.355	-0.201	-0.389	0.394	-0.127	0.038	-0.206	-0.082	0.116	0.000

Tabla 4.3: Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 2D

Como se puede ver en la Tabla 4.3, las varillas pasivas optimizan el espacio de trabajo cuando tienen la misma longitud ($b_1 = b_2$) y sus valores son altos. Por otro lado, en términos generales se advierte que los pasos de las articulaciones helicoidales tienden a tomar valores absolutos grandes ($|\lambda_1| \approx |\lambda_2| \approx |\lambda_3| \approx |\lambda_4|$). En cuanto a las alturas iniciales de las tuercas, z_{01} suele tomar valores pequeños, pero z_{02} no toma un rango más concreto de valores. La distancia entre la tuerca 3 c_{10} y el efector final tampoco es concluyente. En cambio, la distancia entre la tuerca 4 y el efector final sí toma un rango menor de valores. Por último, tal como se advertía en el análisis de sensibilidad realizado, se mejora considerablemente el espacio de trabajo cuando los apoyos de los actuadores son coaxiales ($d = 0$).

Para tener una buena resolución del espacio de trabajo 2D en la maximización, se ha utilizado una malla de 201x201 puntos en la discretización de cada eje X e Y. Teniendo esto en cuenta, los puntos y porcentajes de malla alcanzada son:

	Puntos totales	Porcentaje de malla alcanzada
1	15620	38.66%
2	15556	38.50%
3	15640	38.71%
4	15500	38,36%
5	15588	38.58%
6	15536	38.45%

Tabla 4.4: Puntos y porcentaje de malla alcanzada empleando $n = 201$ puntos en la discretización de cada eje X e Y cuando $z = \varphi = 0$

4.2. MAXIMIZACIÓN DEL ESPACIO DE TRABAJO 4D

Tras haber comprobado el buen funcionamiento del código de Matlab para el espacio de trabajo 2D y que el algoritmo matemático mayoriza este espacio correctamente, se procede a extender el anterior código al espacio 4D y maximizarlo.

Sin embargo, el número de puntos utilizado en la discretización de cada eje X e Y para la maximización del espacio 2D es demasiado alto para utilizarlo en el espacio 4D. Mientras en el espacio 2D la cinemática se resolvía $n^2 = 201^2 = 40401$ veces, utilizando este mismo número de puntos, en el espacio de trabajo 4D se resolvería $n^4 = 201^4 = 1632240801$ veces. Esto emplearía demasiado tiempo en cálculos, por lo que es necesario disminuir el número de puntos. Finalmente, se establece $n = 17$ puntos para cada eje del espacio 4D, llegando a ejecutarse $n^4 = 17^4 = 83521$ veces el código de la cinemática inversa cada vez que se ejecuta el código del espacio de trabajo 4D implementado para la maximización del mismo.

Se empiezan las pruebas de maximización del espacio de trabajo en 4D con los diseños iniciales comentados, pero se obtienen los diseños que se muestran en la Tabla 4.5.

`diseno_optimo4D = simulannealbnd(@esp_trabajo4d,diseno_inicial,minimos,maximos);`

	b_1 (m)	b_2 (m)	λ_1 ($\frac{m}{rad}$)	λ_2 ($\frac{m}{rad}$)	λ_3 ($\frac{m}{rad}$)	λ_4 ($\frac{m}{rad}$)	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
1	0.397	0.397	0.400	-0.399	0.001	0.000	-0.006	-0.208	-0.137	0.064	0.000
2	0.399	0.398	0.384	0.385	0.000	0.000	0.038	-0.165	-0.125	0.079	0.000
3	0.383	0.383	-0.393	0.394	0.000	0.000	-0.003	-0.182	-0.082	0.092	0.000

Tabla 4.5: Análisis 1. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D en un primer intento

Los puntos y porcentajes de malla alcanzada son:

	Puntos totales	Porcentaje de malla alcanzada
1	22236	26.62%
2	22304	26.70%
3	22916	27.44%

Tabla 4.6: Puntos y porcentaje de malla alcanzada en un primer intento empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 1.

Como se puede ver en la Tabla 4.6, el espacio de trabajo ha aumentado. Sin embargo, en las tres pruebas realizadas que se muestran en la Tabla 4.5, los valores de los pasos de las articulaciones 3 y 4 son nulos. Estos valores no son válidos por dos razones. Primero, esto implicaría el uso de articulaciones rotacionales en lugar de las helicoidales. Es decir, el valor nulo en las articulaciones helicoidales elimina el avance traslacional de este tipo de articulaciones, por lo que únicamente habría un movimiento de rotación. En segundo lugar, el robot se encontraría en una configuración singular en todo momento. Al igual que se tuvo en cuenta para la construcción del prototipo, el valor de los pasos de las articulaciones 3 y 4 debe ser distinto en el robot óptimo ($\lambda_3 \neq \lambda_4$) para que sea posible controlar el efector final.

Debido a esta situación, es necesario limitar el rango de valores admisibles de estos parámetros geométricos. Esto se conseguiría restringiendo los valores de λ_3 y λ_4 para que fueran distintos y no nulos, obteniendo así, dos rangos (positivo y negativo) para cada parámetro. De esta forma, se ejecutaría el algoritmo de optimización en cuatro casos (Tabla 4.7).

	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$
1	(+)	(+)
2	(+)	(-)
3	(-)	(+)
4	(-)	(-)

Tabla 4.7: Ejecuciones necesarias para el estudio de un único diseño inicial

Tras finalizar cada ejecución de los casos 1 y 4, en los que se tiene el mismo signo, habría que asegurar que los resultados no tuvieran el mismo valor.

Sin embargo, para evitar excesivos tiempos de cálculo y simplificar, se decide emplear el caso 2 de la Tabla 4.7 en el que λ_3 toma valores positivos y, λ_4 , negativos. El valor absoluto mínimo decidido para cada parámetro es $|\lambda_3| = |\lambda_4| = 0.03 m$.

	b_1 (m)	b_2 (m)	λ_1 $\left(\frac{m}{rad}\right)$	λ_2 $\left(\frac{m}{rad}\right)$	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
<i>minimos</i>	0.1	0.1	-0.4	-0.4	0.03	-0.4	-0.083	-0.22	-0.14	0.03	0
<i>maximos</i>	0.4	0.4	0.4	0.4	0.4	-0.03	0.057	-0.14	-0.03	0.14	0.4

Tabla 4.8: Valores actualizados de los vectores *minimos* y *maximos* para evitar valores no válidos de las articulaciones helicoidales 3 y 4 en la ejecución del algoritmo de optimización

Realizando de nuevo las seis pruebas de optimización con los vectores *minimos* y *maximos* actualizados (Tabla 4.8), los diseños óptimos a los que se llegan son:

	b_1 (m)	b_2 (m)	λ_1 $\left(\frac{m}{rad}\right)$	λ_2 $\left(\frac{m}{rad}\right)$	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
1	0.390	0.390	0.400	0.400	0.030	-0.030	0.014	-0.196	-0.074	0.136	0.000
2	0.383	0.383	0.400	0.400	0.030	-0.030	0.023	-0.159	-0.047	0.135	0.000
3	0.390	0.390	0.400	0.400	0.030	-0.030	0.052	-0.188	-0.102	0.138	0.000
4	0.390	0.390	-0.400	-0.398	0.030	-0.030	-0.006	-0.151	-0.060	0.069	0.000

5	0.389	0.390	0.400	0.400	0.030	-0.030	0.055	-0.149	-0.048	0.140	0.000
6	0.390	0.389	0.399	0.400	0.030	-0.030	0.019	-0.185	-0.133	0.060	0.000

Tabla 4.9: Análisis 2. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D tras reajustar los valores absolutos mínimos de λ_3 y λ_4

Los puntos y porcentajes de malla alcanzada son:

	Puntos totales	Porcentaje de malla alcanzada
1	18184	21.77%
2	18248	21.85%
3	18208	21.80%
4	18056	21.62%
5	18168	21.75%
6	17904	21.44%

Tabla 4.10: Puntos y porcentaje de malla alcanzada empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 2

Como se puede ver en la Tabla 4.9, los valores de $b_1, b_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ y d , toman los mismos valores óptimos en todas las pruebas. Se confirma también que las varillas pasivas optimizan el espacio de trabajo 4D cuando tienen la misma longitud ($b_1 = b_2$) y que se consigue un mayor espacio con $d = 0$. Los valores de los pasos también coinciden entre ellos, siendo $\lambda_1 = \lambda_2$ y $|\lambda_3| = |\lambda_4|$. Por el contrario, los valores de las alturas iniciales de las tuercas z_{01} y z_{02} son los que más varían junto con los valores de las alturas de las tuercas 3 y 4 con respecto al efector final c_{10} y c_{20} .

Dado que no se puede representar el espacio de trabajo 4D, es decir, el espacio XYZPHI, sin perder información en una dimensión; el espacio de trabajo alcanzable (*reachable workspace*) del robot que genera el mejor diseño del Análisis 2 independientemente de los valores del eje PHI, se visualiza en las Figuras 4.6, 4.7 y 4.8.

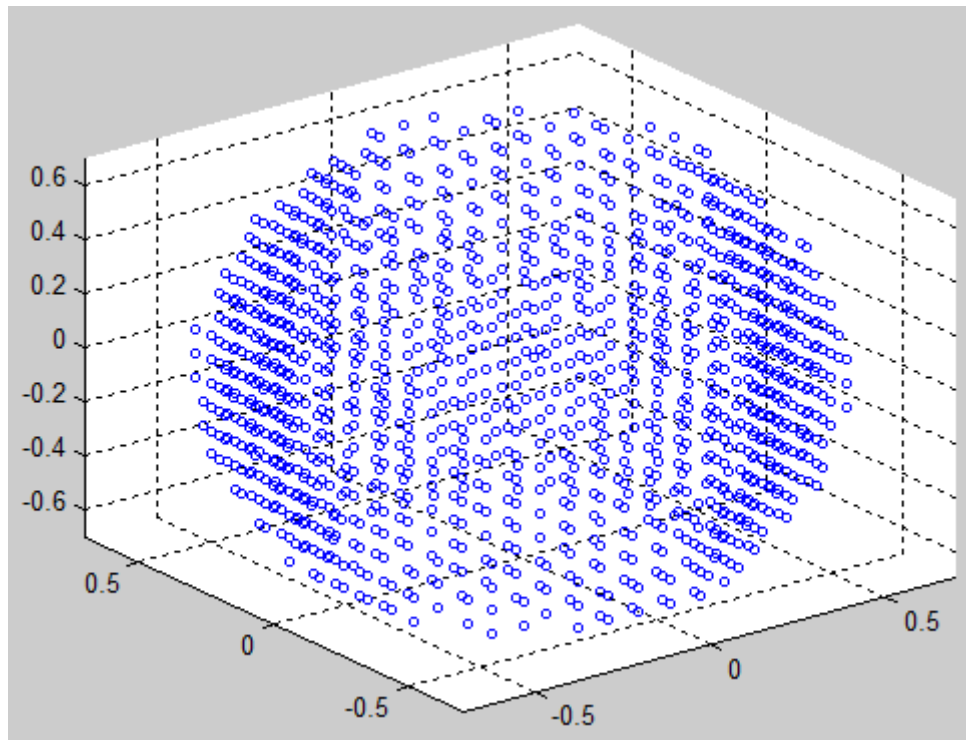


Figura 4.6: Nube de puntos del espacio de trabajo 3D (XYZ) empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2

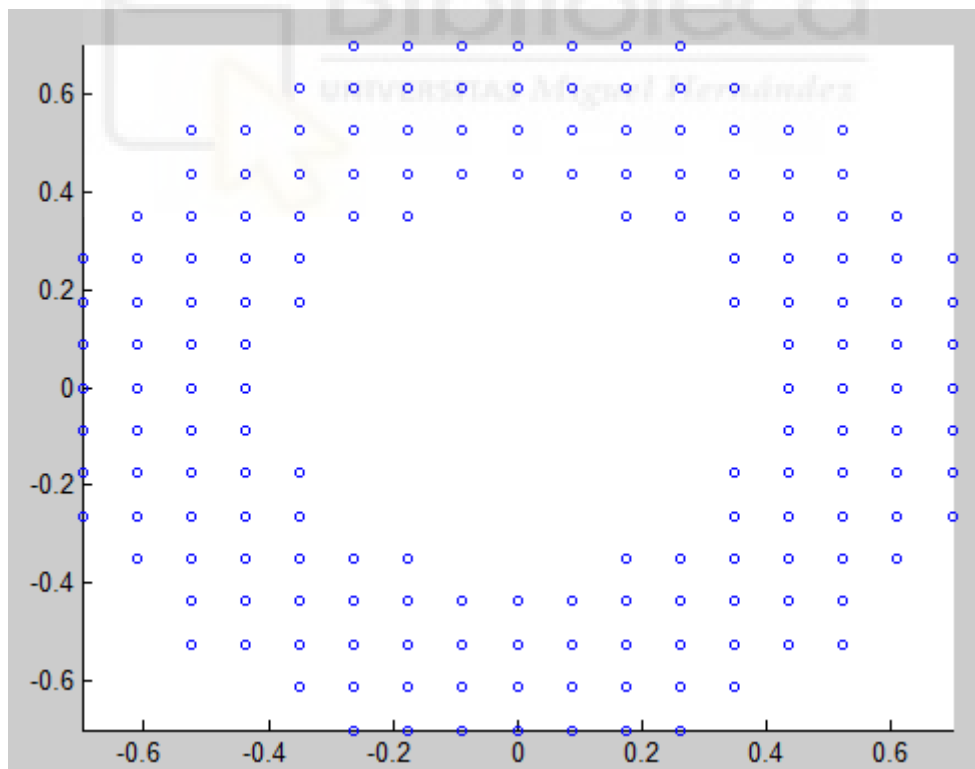


Figura 4.7: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2

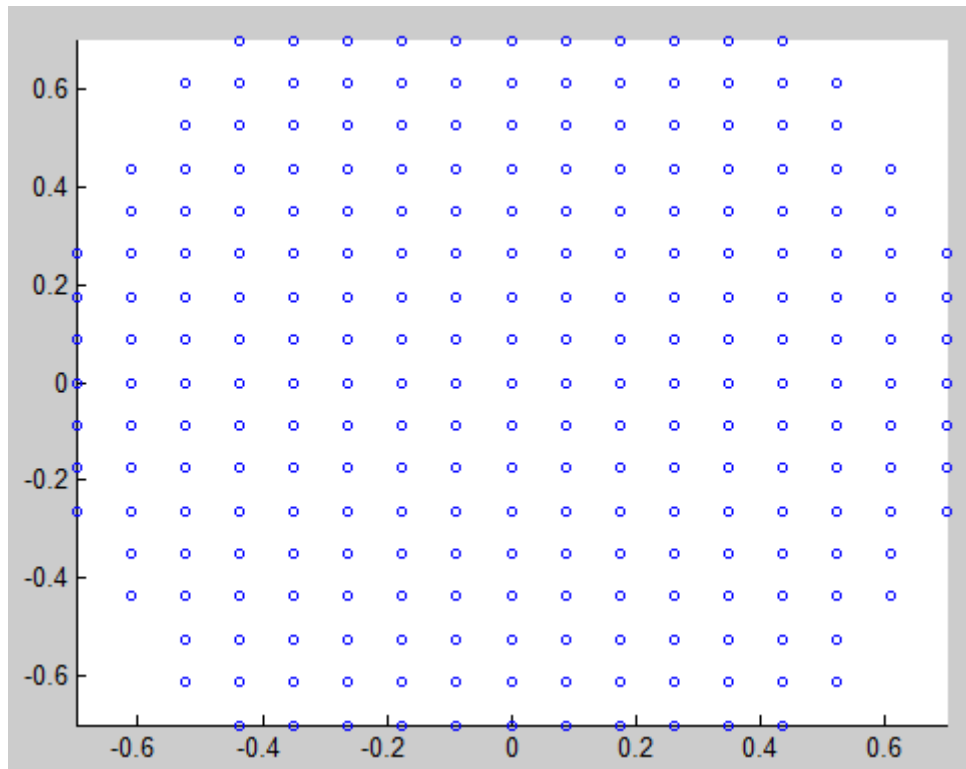


Figura 4.8: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos para el diseño 2 del Análisis 2

El porcentaje de malla al que se llega en la Tabla 4.4 se produce tomando el 100% del espacio 2D cuando $z = \varphi = 0$. En cambio, la Tabla 4.10 toma el 100% del espacio 4D. Por esto, para comparar los valores, es necesario ejecutar los diseños obtenidos en la optimización del espacio 2D (Tabla 4.4) con el código de la optimización 4D. Haciendo esto, los puntos y porcentajes de malla alcanzada son:

	Puntos totales	Porcentaje de malla alcanzada
1	3500	4.19%
2	10916	13.07%
3	4624	5.54%
4	5864	7.02%
5	3788	4.54%
6	6916	8.28%

Tabla 4.11: Puntos y porcentaje de malla alcanzada empleando $n=17$ puntos en la discretización de cada eje X, Y, Z y PHI con los diseños óptimos a los que se llega con la optimización del espacio 2D cuando $z = \varphi = 0$

Comparando las tablas 4.10 y 4.11, se puede ver la mejoría notoria del espacio de trabajo del robot al maximizar el espacio 4D.

Sin embargo, los valores de los parámetros geométricos λ_1 y λ_2 toman los valores límite de los vectores *minimos* y *maximos* ($|\lambda_1| = |\lambda_2| = 0.4 \text{ m/rad}$). Estos valores se eligieron en un principio para abarcar un gran rango de posibles valores. Pero como ya se ha comentado, unos valores grandes en los pasos de las articulaciones helicoidales pueden generar la eliminación de la forma helicoidal de las mismas, lo que convierte el movimiento helicoidal en traslacional. Es decir, desaparecen las articulaciones helicoidales y degeneran en articulaciones prismáticas prácticamente puras. Para evitar este comportamiento se modifican los valores de los pasos elegidos y se establece que, por cada revolución de la articulación, se avanzará como máximo 40 cm (0.0637 m/rad). Además, para evitar valores nulos en las articulaciones λ_3 y λ_4 , se establece un valor absoluto mínimo con el que, por cada revolución, mínimo se avance 3cm (0.0048 m/rad).

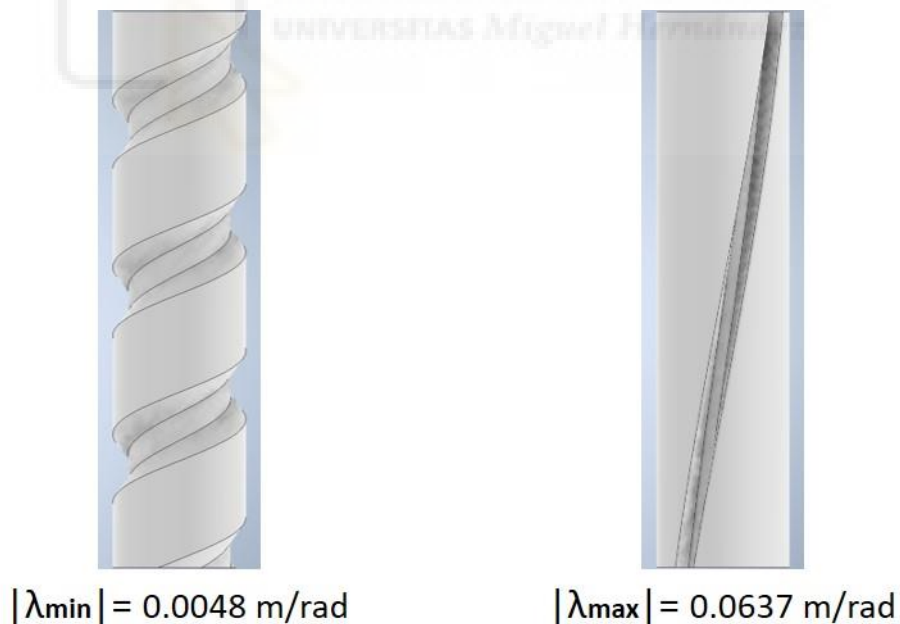


Figura 4.9: Valores absolutos mínimo y máximo para los pasos de las articulaciones helicoidales en

	b_1 (m)	b_2 (m)	λ_1 $\left(\frac{m}{rad}\right)$	λ_2 $\left(\frac{m}{rad}\right)$	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
<i>minimos</i>	0.1	0.1	-0.0637	-0.0637	0.0048	-0.0637	-0.083	-0.22	-0.14	0.03	0
<i>maximos</i>	0.4	0.4	0.0637	0.0637	0.0637	-0.0048	0.057	-0.14	-0.03	0.14	0.4

Tabla 4.12: Valores actualizados de los vectores *minimos* y *maximos* para asegurar la existencia de las articulaciones helicoidales y que no degeneren en una rotación pura o una traslación pura

Los resultados óptimos conseguidos con los nuevos vectores *minimos* y *maximos* que se muestran en la Tabla 4.12 son:

	b_1 (m)	b_2 (m)	λ_1 $\left(\frac{m}{rad}\right)$	λ_2 $\left(\frac{m}{rad}\right)$	λ_3 $\left(\frac{m}{rad}\right)$	λ_4 $\left(\frac{m}{rad}\right)$	z_{01} (m)	z_{02} (m)	c_{10} (m)	c_{20} (m)	d (m)
1	0.106	0.106	0.063	- 0.063	0.005	- 0.005	-0.006	-0.202	-0.130	0.066	0.000
2	0.104	0.103	-0.022	-0.039	0.010	-0.014	0.042	-0.153	-0.116	0.045	0.000
3	0.103	0.105	0.063	- 0.063	0.005	-0.005	0.039	-0.140	-0.129	0.044	0.000
4	0.104	0.104	0.062	0.063	0.005	- 0.005	-0.080	-0.213	-0.048	0.077	0.000
5	0.107	0.104	-0.057	0.057	0.005	-0.005	0.011	-0.147	-0.033	0.130	0.000
6	0.106	0.106	0.063	0.063	0.005	-0.005	-0.082	-0.169	-0.053	0.033	0.000

Tabla 4.13: Análisis 3. Valores de los diseños óptimos obtenidos en la maximización del espacio de trabajo 4D tras ajustar adecuadamente los valores de los pasos para las articulaciones helicoidales

Los puntos y porcentajes de malla alcanzada son:

	Puntos totales	Porcentaje de malla alcanzada
1	12548	15.02%
2	11660	13.96%
3	12184	14.59%
4	12376	14,82%
5	12128	14.52%
6	12544	15.02%

Tabla 4.14: Puntos y porcentaje de malla alcanzada empleando $n = 17$ puntos en la discretización de cada eje X, Y, Z y PHI para el Análisis 3

Aunque el porcentaje de malla alcanzada en el Análisis 3 (Tabla 4.14) ha disminuido en comparación con la alcanzada en el Análisis 2 anterior (Tabla 4.10), esto no significa que no se haya mejorado el espacio de trabajo. Como se ha comentado, los valores anteriores de los pasos de las articulaciones helicoidales no eran realmente adecuados, por lo que comparar los resultados de este último análisis con los del anterior no sería correcto.

Sin embargo, en términos generales, las conclusiones son las mismas aunque los valores óptimos hayan variado. Los valores de $b_1, b_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ y d , siguen tomando los mismos valores óptimos en todas las pruebas. Las varillas pasivas siguen optimizando el espacio de trabajo 4D cuando tienen la misma longitud ($b_1 = b_2$). Y también se consigue un mayor espacio de trabajo con $d = 0$. Los valores de los pasos también coinciden entre ellos, siendo $|\lambda_1| = |\lambda_2|$ y $|\lambda_3| = |\lambda_4|$. Por otro lado, los valores de las alturas iniciales de las tuercas (z_{01}, z_{02}) y los valores de las alturas de las tuercas 3 y 4 con respecto al efector final (c_{10}, c_{20}) vuelven a ser los parámetros que más varían y que prácticamente determinan la diferencia de puntos entre pruebas.

Tras el análisis de seis pruebas de optimización en las que se obtienen valores bastante concluyentes, el diseño que mayor espacio de trabajo produce es el caso 1 de la tabla 4.14. Por tanto, estos valores son los que se tomarán para el diseño del robot óptimo (Figura 4.10).

$diseño_optimo = [0.106, 0.105, 0.063, -0.063, 0.005, -0.005, -0.006, -0.202, -0.130, 0.066, 0.000]$

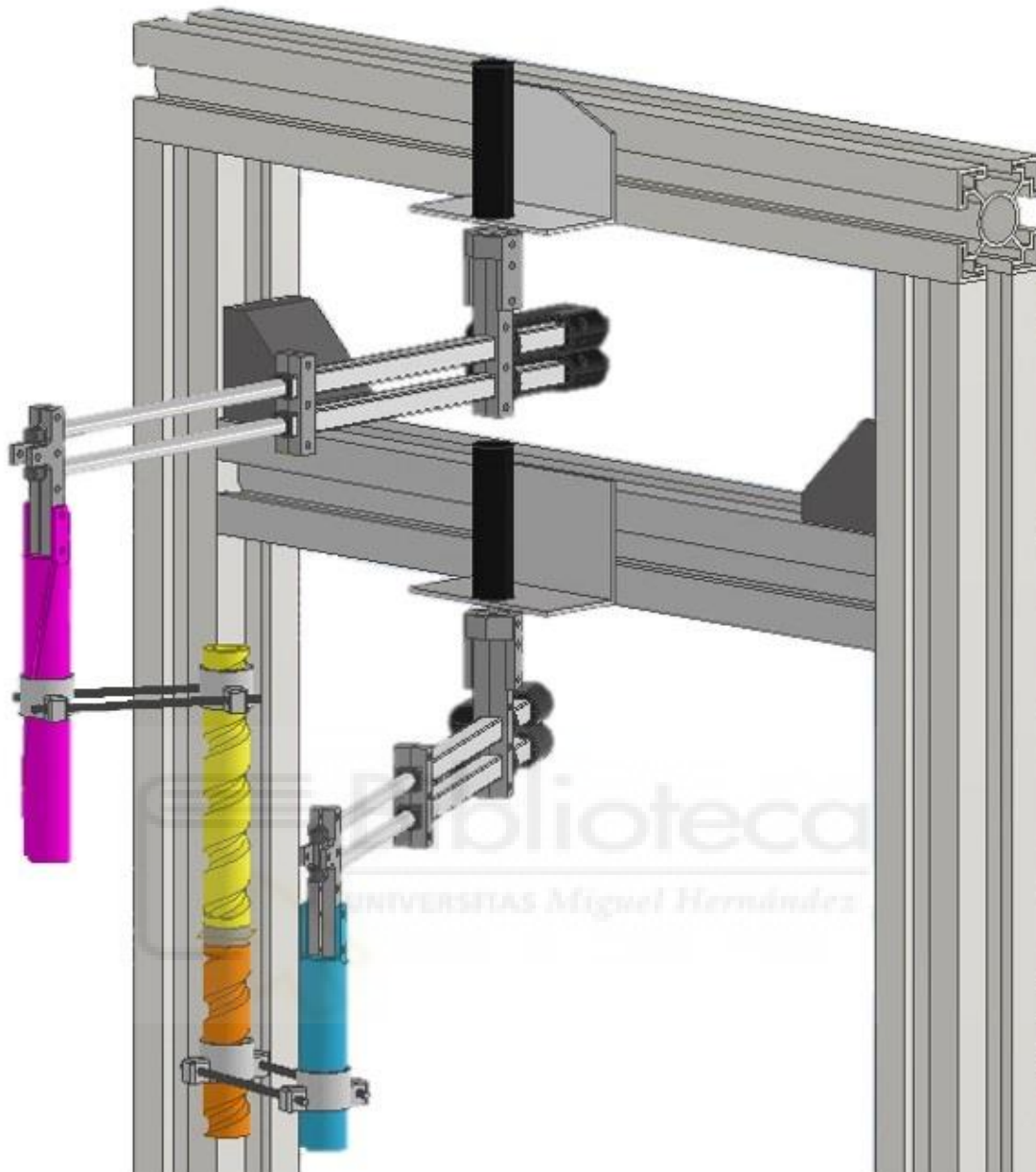


Figura 4.10: Robot optimizado final

Como se ha comentado anteriormente, no se puede representar el espacio de trabajo 4D sin perder información en una dimensión. El espacio de trabajo alcanzable o máximo 3D (*reachable workspace*) que genera el robot optimizado independientemente de los valores del eje PHI, se visualiza en las Figuras 4.11, 4.12 y 4.13.

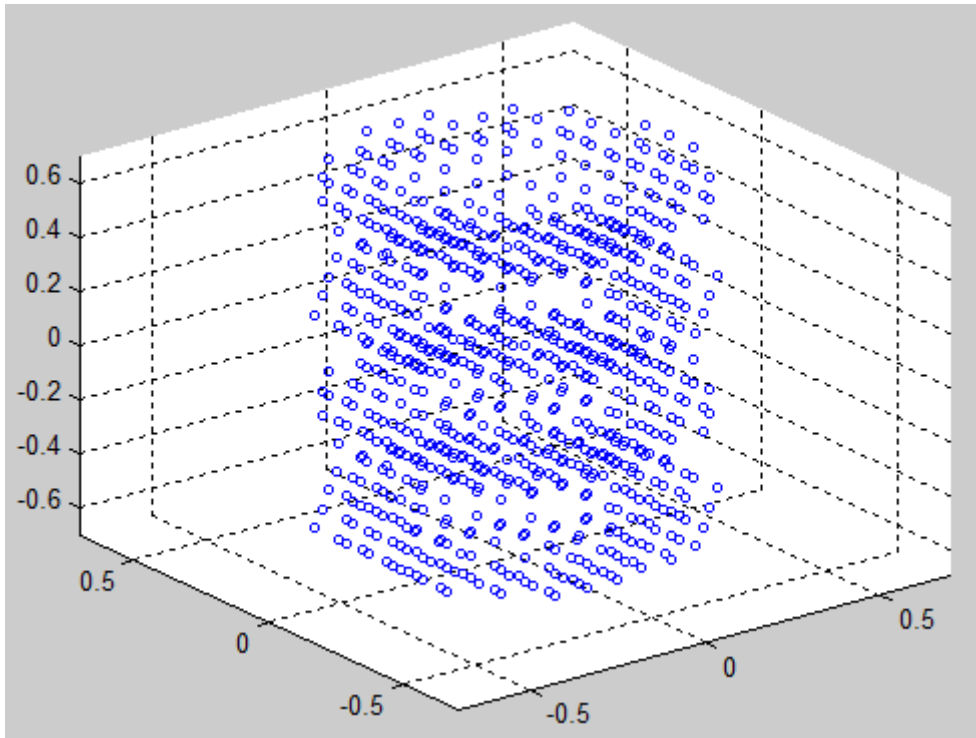


Figura 4.11: Nube de puntos del espacio de trabajo 3D (XYZ) empleando una malla de $n=17$ puntos

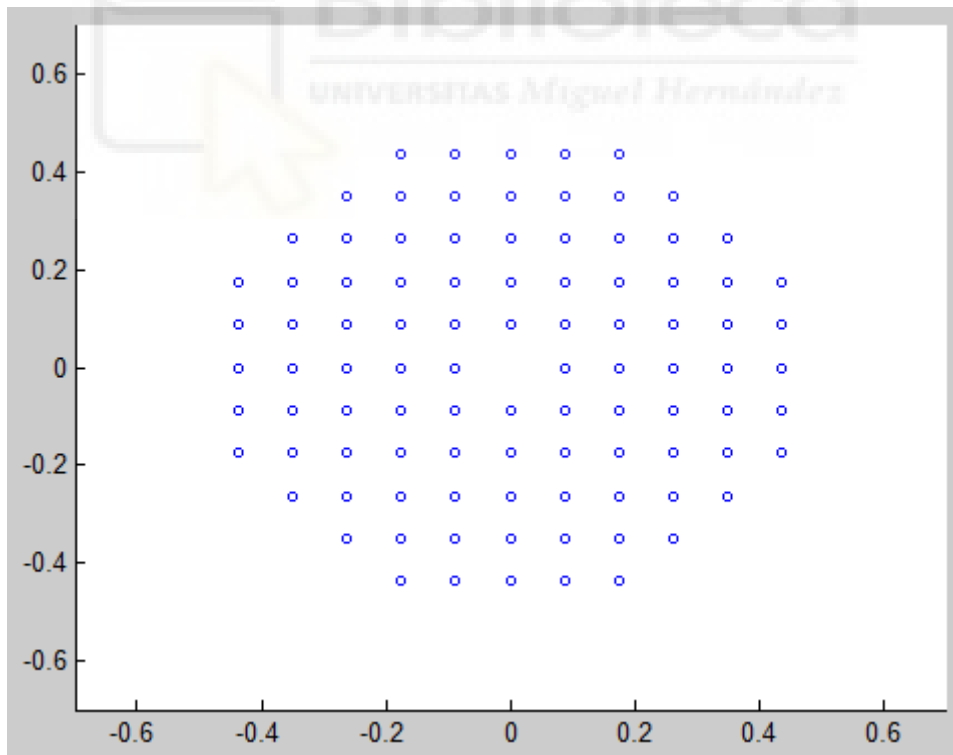


Figura 4.12: Nube de puntos del espacio de trabajo en el plano XY empleando una malla de $n=17$ puntos

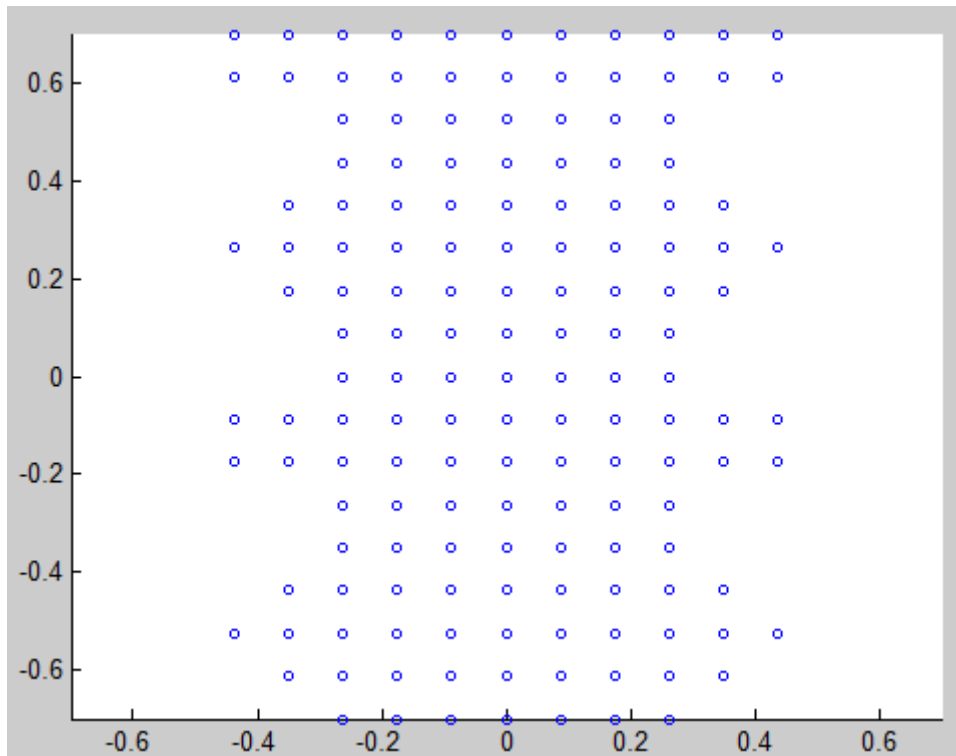


Figura 4.13: Nube de puntos del espacio de trabajo en el plano YZ empleando una malla de $n=17$ puntos

Dependiendo de la posición en el eje Z y la orientación del efector final, el robot consigue un mayor o menor alcance en el espacio de trabajo. Como se puede ver en las Figuras 4.11, 4.12 y 4.13, los máximos valores alcanzados en el espacio de trabajo se producen tanto en valores de alturas positivos como negativos. Además, se visualiza un patrón circular en torno al robot debido al propio movimiento helicoidal del robot.

Para visualizar más fácilmente el espacio de trabajo, se emplea la interfaz creada con EJS. Ejemplos de los espacios mínimos y máximos conseguidos se muestran en las Figuras 4.14 y 4.15.

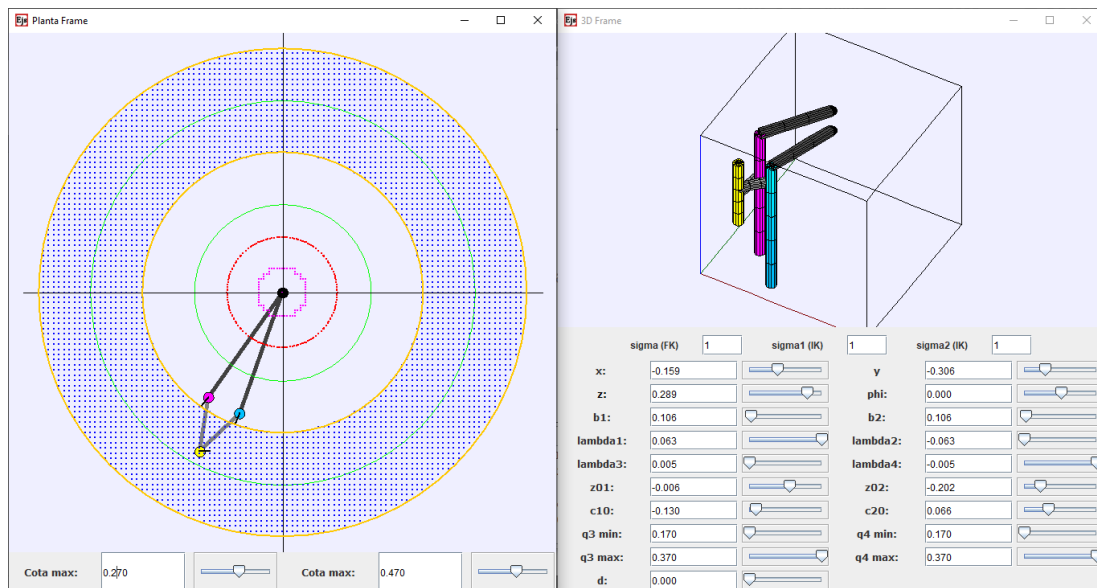


Figura 4.14: Ejemplo de configuración que consigue uno de los espacios de trabajo máximos con valores aproximados entre 0.47 m y 0.27 m

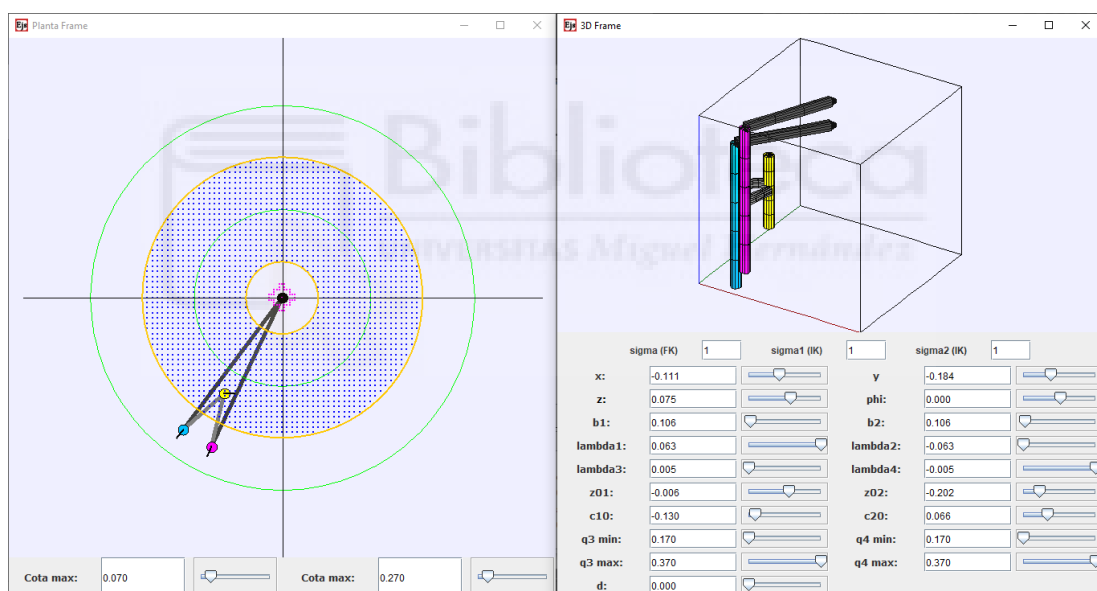


Figura 4.15: Ejemplo de configuración que consigue uno de los espacios de trabajo mínimos con valores aproximados entre 0.27 m y 0.08 m

Esta optimización del diseño se ha realizado con los valores de 0.17m para las variables q_{3_min} y q_{4_min} . Sin embargo, como se averiguó durante la comprobación del funcionamiento del prototipo construido, este diseño óptimo deberá ser equilibrado para evitar la flexión de las piezas. Para solucionarlo se deben mover los apoyos de los actuadores lineales, lo que significa la disminución de los valores q_{3_min} y q_{4_min} y, por tanto, de la mejora del espacio de trabajo, en cualquier caso.

5. CONCLUSIONES

Se ha diseñado y optimizado un robot paralelo de topología novedosa, compuesto por dos brazos en configuración RPHH cada uno, para lograr el máximo espacio de trabajo posible.

El cálculo de la cinemática directa e inversa del robot se realizó mediante un análisis geométrico. El espacio de trabajo del robot se averiguó basando su cálculo en la cinemática inversa. También se calcularon las configuraciones singulares del robot mediante un análisis Jacobiano. Después, se simuló el movimiento del robot con la herramienta EJS y se realizó un análisis de sensibilidad de los parámetros geométricos para hallar unas medidas iniciales adecuadas con las que poder construir un prototipo. Tras ensamblar el prototipo habiendo modelado e impreso en 3D las piezas necesarias, se realizó una prueba de movilidad en bucle abierto para constatar el buen funcionamiento del robot. No obstante, el espacio de trabajo seguía siendo reducido y se prosiguió a realizar un análisis de optimización del robot utilizando el algoritmo de recocido simulado mediante la herramienta de cómputo numérico Matlab. De esta forma, se consiguió llegar a un diseño del robot que maximiza el espacio de trabajo del mismo y cumplir el objetivo del trabajo de fin grado.

El diseño del robot propuesto es simple en comparación con otros robots con el mismo tipo de aplicación, grados de libertad, movimiento de Schönflies y articulaciones empleadas. Por un lado, se evitan complicaciones de diseño al no utilizar articulaciones esféricas y, por otro lado, tampoco se utilizan brazos compuestos por paralelogramos como suele ser más normal en robots paralelos. Además, la simetría del robot facilita su análisis cinemático.

El diseño óptimo del robot genera también un buen espacio de trabajo que varía en función de la altura y orientación del efector final; y que perfectamente podría emplearse para operaciones de *pick and place* (recoger y colocar), como se había previsto.

5.1. TRABAJOS FUTUROS

A partir del diseño y análisis del robot propuesto, se podrían realizar otros trabajos.

- Análisis de las colisiones entre eslabones

Por simplicidad, en el diseño del robot propuesto se han omitido las posibles colisiones que puedan producirse en varios eslabones del robot. No obstante, el espacio de trabajo se verá limitado por estas colisiones, por lo que será necesario su análisis para una futura implementación práctica del robot.

- Control del robot en bucle cerrado

- Modelado cinemático de velocidades y aceleraciones

Para generar trayectorias y lograr que el efector final las ejecute con la velocidad requerida en cada momento, es necesario el análisis de las velocidades y aceleraciones.

- Modelado dinámico

Para hallar la relación entre el movimiento, estudiado con el análisis cinemático, y las fuerzas que lo producen, se necesita estudiar la dinámica del robot.

- Implementación del circuito controlador empleando una placa de Arduino

- Construir el robot óptimo diseñado y probar su funcionamiento con un efector final

El robot propuesto está pensado para realizar operaciones de *pick and place* (recoger y colocar), por lo que utilizar una pinza como efector final y realizar pruebas de movilidad, repetitividad y precisión sería otra de las tareas a realizar en el futuro.

Bibliografía

- [1] M. Díaz Rodríguez *et al.*, “Aplicación de los Robots Paralelos,” *HAL Arch. Ouvert.*, 2018, Accessed: Sep. 06, 2020. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01907282>.
- [2] R. Aracil, R. J. Saltarén, J. M. Sabater, and O. Reinoso, “Robots paralelos: Máquinas con un pasado para una robótica del futuro,” *Rev. Iberoam. Automática e Informática Ind.*, vol. 3, pp. 16–28, 2006, [Online]. Available: <https://158.42.9.104/index.php/RIAI/article/view/8105>.
- [3] R. L. Norton, *Diseño de Maquinaria, síntesis y análisis de máquinas y mecanismos*, 4th ed. Boston: McGraw Hill Higher Education, 2009.
- [4] K. Kishore Kumar, A. Srinath, and A. Srikanth, “Design analysis of parallel robot for surgical applications,” 2013. [Online]. Available: <https://pdfs.semanticscholar.org/0080/73e5d7d58f84a220ea1b92b5265309a54e26.pdf>.
- [5] E. V. Gough, “Contribution to discussion of papers on research in automobile stability, control and tyre performance,” *Proc. Inst. Mech. Eng.*, pp. 392–395, 1956.
- [6] D. Stewart, “A Platform with Six Degrees of Freedom,” *Proc. Inst. Mech. Eng.*, vol. 180, no. 1, pp. 371–386, Jun. 1965, doi: 10.1243/pime_proc_1965_180_029_02.
- [7] Y. Yu, Z. B. Xu, Q. W. Wu, P. Yu, S. He, and G. Q. Wang, “Kinematic analysis and testing of a 6-RR RP RR parallel manipulator,” *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 231, no. 13, pp. 2515–2527, 2017, doi: 10.1177/0954406216633034.
- [8] R. Clavel, “Une nouvelle structure de manipulateur parallèle pour la robotique légère,” *APII*, 23, vol. 6, pp. 501–519, 1989.
- [9] S. Krut, O. Company, V. Nabat, and F. Pierrot, “Heli4: A parallel robot for scara motions with a very compact traveling plate and a symmetrical design,” *IEEE Int. Conf. Intell. Robot. Syst.*, no. 1, pp. 1656–1661, 2006, doi: 10.1109/IROS.2006.282120.
- [10] T. Harada and J. Angeles, “Kinematics and singularity analysis of a CRRHHRRC parallel Schönflies motion generator,” *Trans. Can. Soc. Mech. Eng.*, vol. 38, no. 2, pp. 173–183, 2014, doi: 10.1139/tcsme-2014-

0012.

- [11] A. Moritz Schoenflies, “La géométrie du mouvement: exposé synthétique,” 1893.
- [12] P. C. Lee and J. J. Lee, “On the kinematics of a new parallel mechanism with Schoenflies motion,” *Robotica*, vol. 34, no. 9, pp. 2056–2070, 2016, doi: 10.1017/S0263574714002732.
- [13] R. Stawell Ball, *A Treatise on the Theory of Screws*. Cambridge University Press, 1900.
- [14] “Ultimaker Cura Soporte.” <https://support.ultimaker.com/hc/en-us/sections/360003548619-Print-settings>.
- [15] “Algoritmo Simulannealbnd de Matlab.” <https://es.mathworks.com/help/gads/simulannealbnd.html>.



Anexos

Anexo 1. Código implementado en EJS.....	108
Anexo 2. Análisis de sensibilidad	115
Anexo 3. Código implementado en Matlab.....	138



Anexo 1. Código implementado en EJS

Los algoritmos explicados en los capítulos 2.2, 2.3, 2.4 y 2.5, quedan implementados como se muestran a continuación.

▪ Cinemática directa (Capítulo 2.2)

```
public double[] cinematica_directa(double q1, double q3, double q2, double q4,
double d, double b1, double b2, double z01, double z02, double c10,
double c20, double lambda1, double lambda2, double lambda3,
double lambda4, double sigma) {
    double x1, y1, x2, y2;
    x1 = q3*Math.cos(q1);
    y1 = q3*Math.sin(q1);
    x2 = q4*Math.cos(q2) + d;
    y2 = q4*Math.sin(q2);
    double beta = Math.atan2(y2-y1, x2-x1);
    double R = Math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
    double theta1 = sigma*Math.acos((b1*b1 + R*R - b2*b2) / (2*b1*R)) / (2*b1*R);
    double phi1 = beta + theta1;
    double x = x1 + b1*Math.cos(phi1);
    double y = y1 + b1*Math.sin(phi1);
    double phi2 = Math.atan2(y-y2, x-x2);
    double psi1 = phi1 - q1;
    double psi2 = phi2 - q2;
    double phi = (z02 - z01 + lambda2*psi2 - lambda1*psi1 + c20 - c10) /
        (lambda3 - lambda4);
    double z = z01 + lambda1*psi1 + lambda3*phi + c10;

    return new double[] {x, y, z, phi, phi1, phi2};
}
```

▪ Cinemática inversa (Capítulo 2.3)

```
public double[] cinematica_inversa(double x, double y, double z, double phi,
double d, double b1, double b2, double z01, double z02, double c10, double c20,
double lambda1, double lambda2, double lambda3, double lambda4,
double sigma1, double sigma2) {
    double psi1 = (z - z01 - phi*lambda3 - c10) / (lambda1);
```

```

double psi2 = (z - z02 - phi*lambda4 - c20) / (lambda2);
// Resolucion de q1, q3
double q3 = - b1*Math.cos(psi1) +
    sigma1*Math.sqrt(- b1*b1*Math.sin(psi1)*Math.sin(psi1) + x*x + y*y);
double cos_q1 = (b1*x*Math.cos(psi1) + b1*y*Math.sin(psi1) + q3*x) /
    (2*b1*q3*Math.cos(psi1) + b1*b1 + q3*q3);
double sin_q1 = (b1*y*Math.cos(psi1) - b1*x*Math.sin(psi1) + q3*y) /
    (2*b1*q3*Math.cos(psi1) + b1*b1 + q3*q3);
double q1 = Math.atan2(sin_q1, cos_q1);
// Resolucion de q2, q4
double q4 = - b2*Math.cos(psi2) + sigma2*Math.sqrt(- b2*b2*Math.sin(psi2) *
    Math.sin(psi2) + x*x - 2*d*x + y*y + d*d);
double cos_q2 = (b2*(x-d) * Math.cos(psi2) + b2*y*Math.sin(psi2) + q4*(x-d)) /
    (2*b2*q4*Math.cos(psi2) + Math.pow(b2,2) + Math.pow(q4,2));
double sin_q2 = (b2*y*Math.cos(psi2) + b2*(d-x) * Math.sin(psi2) + q4*y) /
    (2*b2*q4*Math.cos(psi2) + Math.pow(b2,2) + Math.pow(q4,2));
double q2 = Math.atan2(sin_q2, cos_q2);
double phi1 = q1 + psi1;
double phi2 = q2 + psi2;

return new double[] {q1, q2, q3, q4, phi1, phi2};
}

```

- Espacio de trabajo (Capítulo 2.4)

```

public void espacio_trabajo() {

    // Limites espacio de trabajo estudiado
    double x_min = -0.70;
    double x_max = 0.70;
    double y_min = -0.70;
    double y_max = 0.70;

    // Tamaño mallas
    int Nx = 201;
    int Ny = 201;
}

```

```

// Diferenciales en cada eje
double dx = (x_max - x_min) / (Nx-1);
double dy = (y_max - y_min) / (Ny-1);

malla = new double[Nx*Ny][2];

double x, y;
int contador = 0;
x = x_min;
for(int i=0; i<Nx; i++){ // Bucle que varia x
    y = y_min;
    for(int j=0; j<Ny; j++){ // Bucle que varia y
        //Resolver cinematica_inversa. Se pasan las coordenadas locales (x, y) y el
        //resto de variables son los valores actuales globales
        double[] coords_articulares = cinematica_inversa(x, y, z, phi, d, b1, b2, z01,
        z02, c10, c20, lambda1, lambda2, lambda3, lambda4, sigma1, sigma2);
        //De la solucion obtenida (q1, q2, q3, q4), comprobar cuales son reales con la
        //funcion Double.isNaN(qi)
        if( !(Double.isNaN(coords_articulares[0]) ||
        Double.isNaN(coords_articulares[1]) || Double.isNaN(coords_articulares[2])
        || Double.isNaN(coords_articulares[3]) )){
            //Si son reales, comprobar si q3 y q4 estan entre limites articulares
            //q3_min, q3_max, q4_min y q4_max
            if(coords_articulares[2]>=q3_min && coords_articulares[2]<=q3_max &&
            coords_articulares[3]>=q4_min && coords_articulares[3]<=q4_max){
                //Si se cumple la anterior condicion, se almacena el punto en la malla
                malla[contador][0] = x;
                malla[contador][1] = y;
                contador++;
            }
        }
        y = y + dy;
    }
    x = x + dx;
}
}

```

- Singularidades (Capítulo 2.5)

```
public void singularidades() {  
  
    // Limites espacio de trabajo estudiado  
    double x_min = -0.7;  
    double x_max = 0.7;  
    double y_min = -0.7;  
    double y_max = 0.7;  
  
    // Tamaño mallas  
    int Nx = 201;  
    int Ny = 201;  
  
    // Diferenciales en cada eje  
    double dx = (x_max - x_min) / (Nx-1);  
    double dy = (y_max - y_min) / (Ny-1);  
  
    // Reserva de memoria  
    double[][] malla_x = new double[Nx][Ny];  
    double[][] malla_y = new double[Nx][Ny];  
    double[][] malla_curva = new double[Nx][Ny];  
  
    double q1, q2, q3, q4;  
    double x, y;  
    x = x_min;  
    for(int i=0;i<Nx;i++){  
        y = y_min;  
        for(int j=0;j<Ny;j++){  
            // Almacenar coordenadas de cada punto  
            malla_x[i][j] = x;  
            malla_y[i][j] = y;  
  
            // Resolver cinematica_inversa. Se pasan las coordenadas locales (x, y) y  
            // el resto de variables son los valores actuales globales  
            double[] coords = cinematica_inversa(x, y, z, phi, d, b1, b2, z01, z02, c10,  
            c20, ambda1, lambda2, lambda3, lambda4, sigma1, sigma2);  
            q1 = coords[0];  
        }  
    }  
}
```

```

q2 = coords[1];
q3 = coords[2];
q4 = coords[3];
// Los valores del determinante de la jacobiana Jx en cada punto se
almacenan en malla_curva
malla_curva[i][j] = (b1 * b2 * (lambda4-lambda3) *
Math.cos(q1 + (z-z01-phi*lambda3-c10)/(lambda1)) *
Math.sin(q2 + (z-z02-phi*lambda4-c20)/(lambda2))) /
(lambda1*lambda2) + (b1*b2*(lambda3-lambda4) *
Math.sin(q1 + (z-z01-phi*lambda3-c10) / (lambda1)) *
Math.cos(q2 + (z-z02-fi*lambda4-c20) / (lambda2))) / (lambda1*lambda2);
y += dy;
}
x += dx;
}

```

```

singularidades_tipo2 = new double[Nx*Ny][2];
// fronteras_workspace_NaN == singularidades_tipo1
fronteras_workspace_NaN = new double[Nx*Ny][2];
int contador=0;
int contador_fronteras_workspace_NaN = 0;
int i_vecino, j_vecino;
double x_vecino, y_vecino;
double curva, curva_vecino;

x = x_min;
for(int i=0;i<Nx;i++){
y = y_min;
for(int j=0;j<Ny;j++){
// Se comprueban los puntos vecinos
for(int di=-1;di<=1;di++){
for(int dj=-1;dj<=1;dj++){
// Se excluye el nodo central (no comparar con el mismo punto)
if(di!=0 || dj!=0){
i_vecino = i + di;
j_vecino = j + dj;

```



```

// Excluimos vecinos fuera de la malla
if(i_vecino>=0 && j_vecino >=0 && i_vecino<Nx && j_vecino<Ny){
    curva = malla_curva[i][j];
    curva_vecino = malla_curva[i_vecino][j_vecino];
    x = malla_x[i][j];
    y = malla_y[i][j];
    x_vecino = malla_x[i_vecino][j_vecino];
    y_vecino = malla_y[i_vecino][j_vecino];

    if( !Double.isNaN(malla_curva[i][j]) &&
    !Double.isNaN(malla_curva[i_vecino][j_vecino]) &&
    curva*curva_vecino < 0 ){
        // Interpolacion lineal en los dos ejes
        // determinante de la jacobiana Jx == f donde las
        // singularidades_tipo2 se alcanzan para f=0
        // x1, y1, f1 --> nodo actual
        // x2, y2, f2 --> nodo vecino
        // (x2-x1)/(f2-f1) = (x - x1) / (f - f1) -> (x - x1)/(x2-x1) = (0 - f1)/(f2-f1)
        // →x = -f1*(x2-x1)/(f2-f1) + x1
        // (y2-y1)/(f2-f1) = (y - y1) / (f - f1) -> (y - y1)/(y2-y1) = (0 - f1)/(f2-f1)
        // →y = -f1*(y2-y1)/(f2-f1) + y1
        singularidades_tipo2[contador][0] =
            - curva * (x_vecino - x) / (curva_vecino-curva) + x;
        singularidades_tipo2[contador][1] =
            - curva * (y_vecino - y) / (curva_vecino-curva) + y;
        contador++;
    }
    if( (!Double.isNaN(curva) && Double.isNaN(curva_vecino)) ||
    Double.isNaN(curva) && !Double.isNaN(curva_vecino) ){
        fronteras_workspace_NaN[contador_fronteras_workspace_NaN][0]
            = (x+x_vecino)/2;
        fronteras_workspace_NaN[contador_fronteras_workspace_NaN][1]
            = (y+y_vecino)/2;
        contador_fronteras_workspace_NaN++;
    }
}
}

```

```
    }  
  }  
}  
y += dy;  
}  
x += dx;  
}  
}
```



Anexo 2. Análisis de sensibilidad

El diseño inicial para el análisis de sensibilidad toma los siguientes valores para los parámetros geométricos:

$$\begin{aligned}d &= 0.2 \text{ m} \\b_1 &= b_2 = 0.15 \text{ m} \\ \lambda_1 = \lambda_2 = \lambda_3 &= 0.4 \frac{\text{m}}{\text{rev}} = \frac{0.4 \text{ m}}{2\pi \text{ rad}} \simeq 0.064 \frac{\text{m}}{\text{rad}} \\ \lambda_4 &= -0.4 \frac{\text{m}}{\text{rev}} = -\frac{0.4 \text{ m}}{2\pi \text{ rad}} \simeq -0.064 \frac{\text{m}}{\text{rad}} \\ z_{01} &= 0.125 \text{ m} \\ z_{02} &= 0.06 \text{ m} \\ c_{10} &= -0.035 \text{ m} \\ c_{20} &= 0.03 \text{ m} \\ q_{3_min} &= q_{4_min} = 0.05 \text{ m} \\ q_{3_max} &= q_{4_max} = 0.25 \text{ m}\end{aligned}$$

Las coordenadas operacionales X que se mantienen constantes para visualizar el espacio de trabajo en 2D tienen los siguientes valores:

$$\begin{aligned}z &= 0.1 \text{ m} \\ \varphi &= 1.3 \text{ rad}\end{aligned}$$

El espacio de trabajo que generan estos parámetros iniciales está compuesto por 1356 puntos utilizando una malla de 201x201 puntos y es el que se muestra en la Figura A2.1.

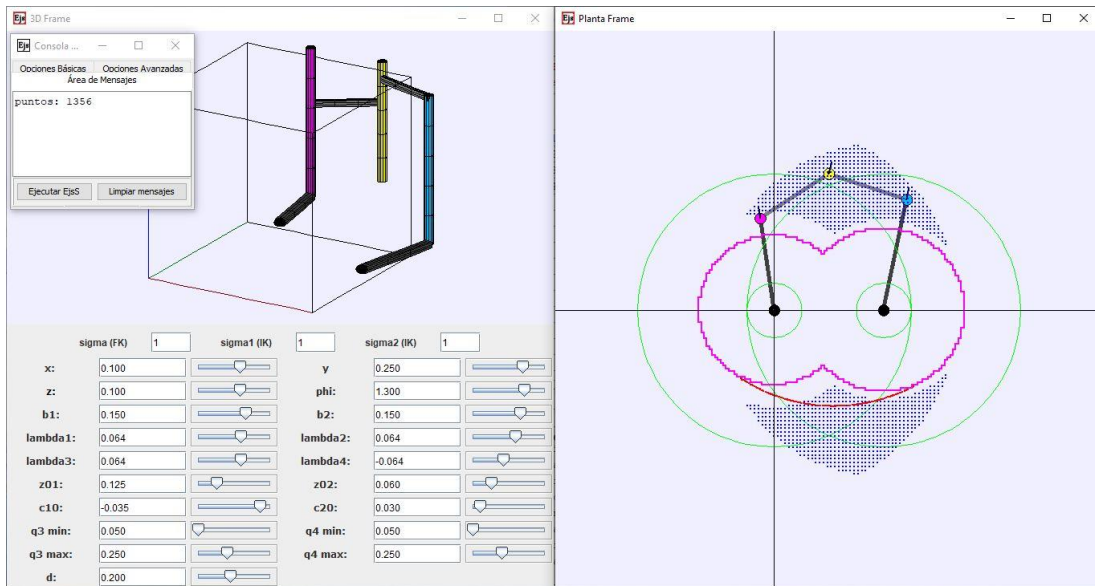
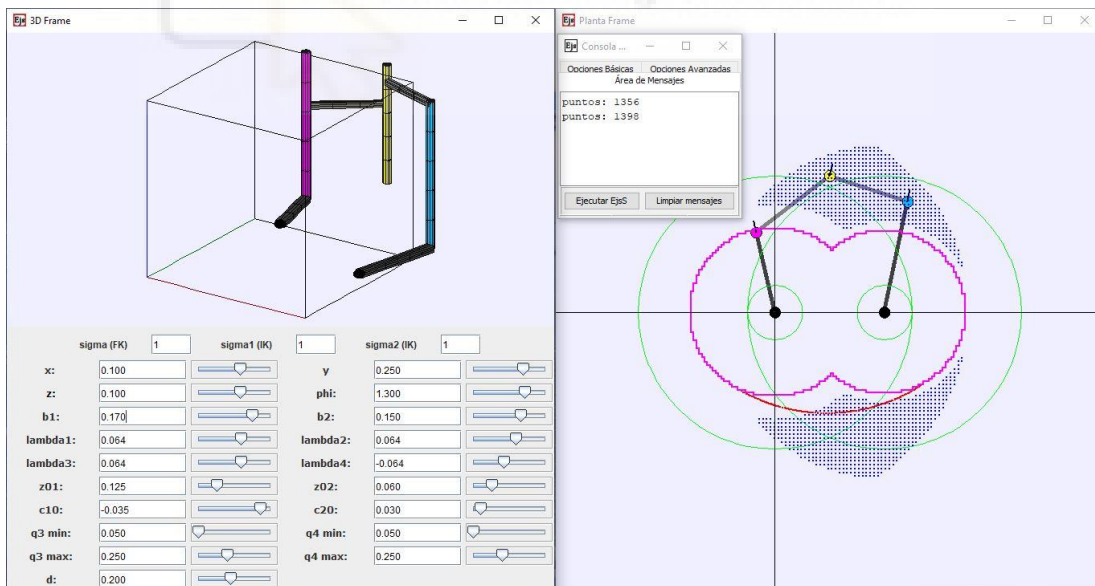


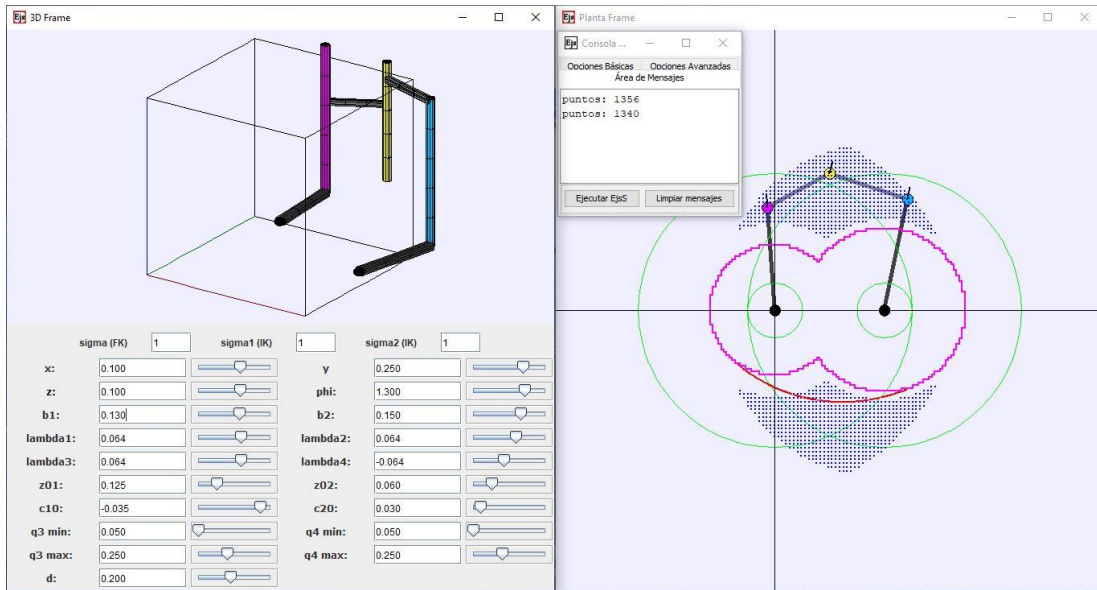
Figura A2.1: Espacio de trabajo obtenido con el diseño inicial para el análisis de sensibilidad

A continuación, se van variando los valores de los parámetros geométricos para averiguar el efecto de éstos sobre el espacio de trabajo en 2D.

- Parámetro $b_1 = 0.15 \text{ m}$ (1356 puntos)
 - $b_1 + 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1398 puntos)

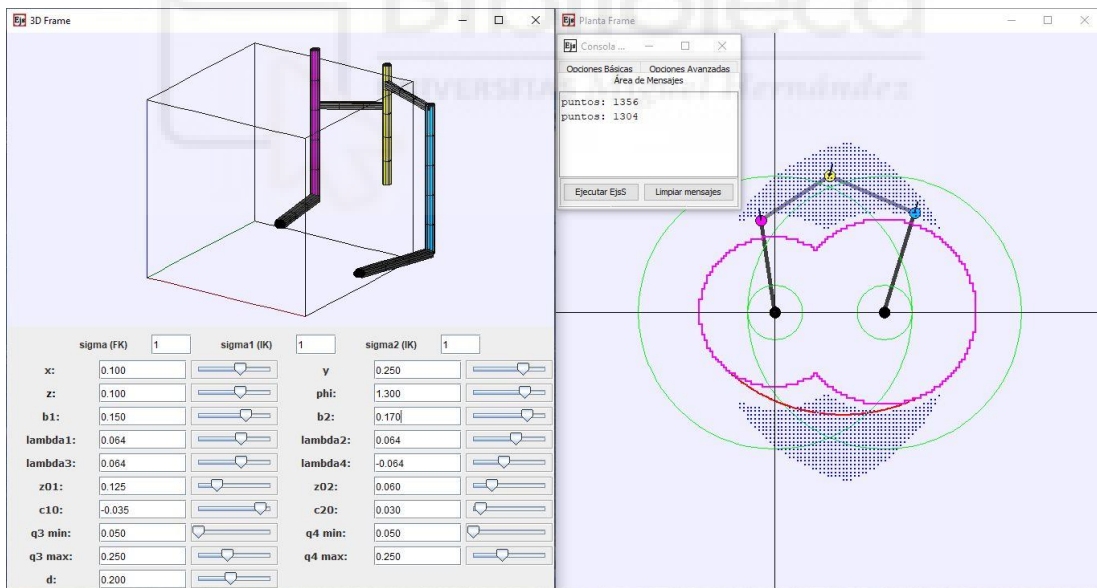


- $b_1 = 0.02m \rightarrow$ menor espacio de trabajo (1340 puntos)

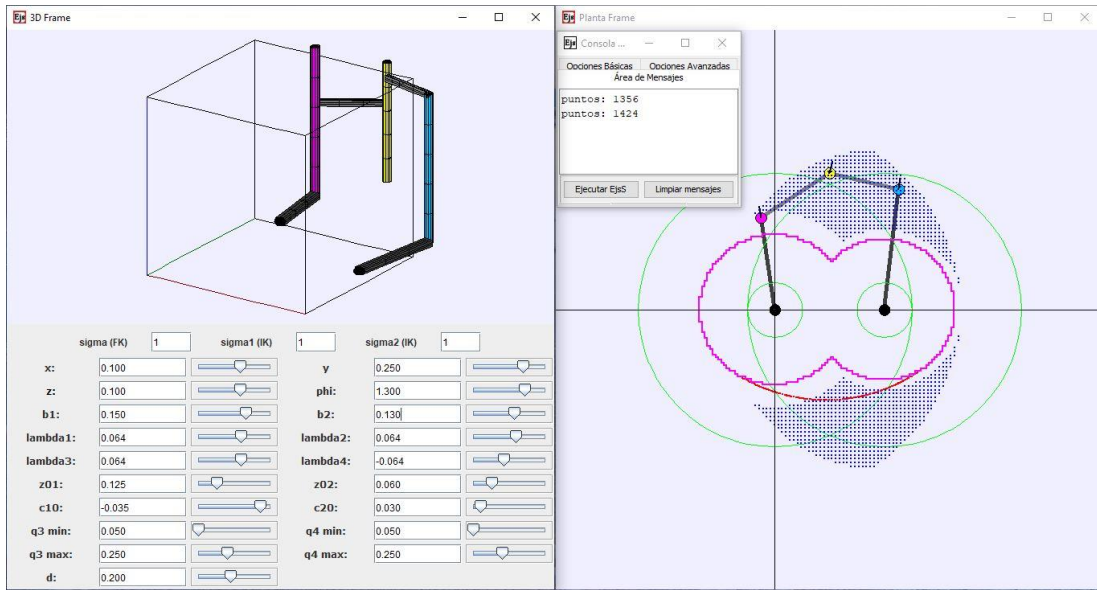


- Parámetro $b_2 = 0.15 m$ (1356 puntos)

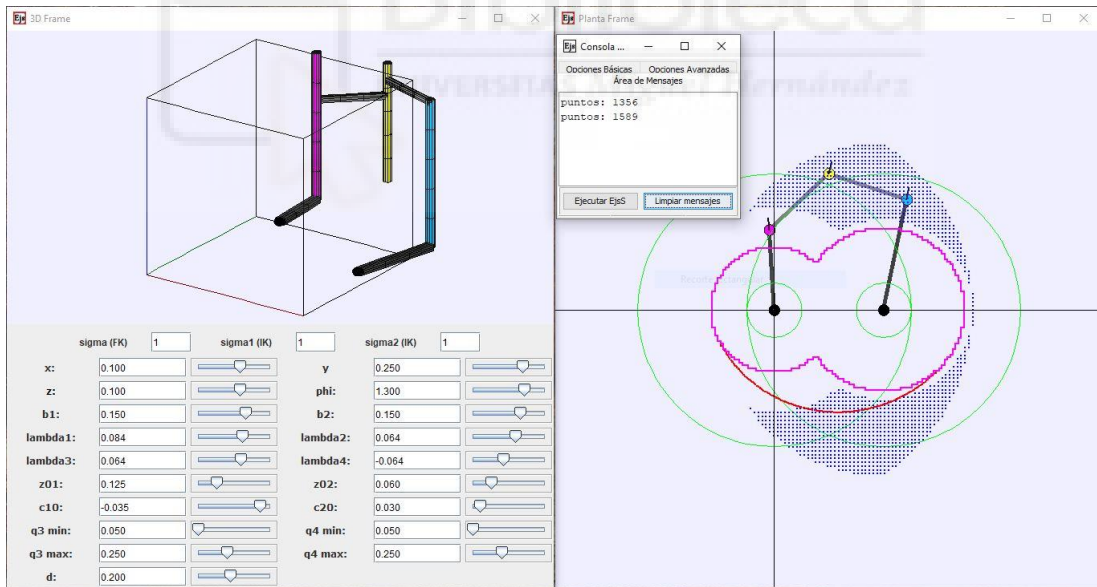
- $b_2 + 0.02m \rightarrow$ menor espacio de trabajo (1304 puntos)



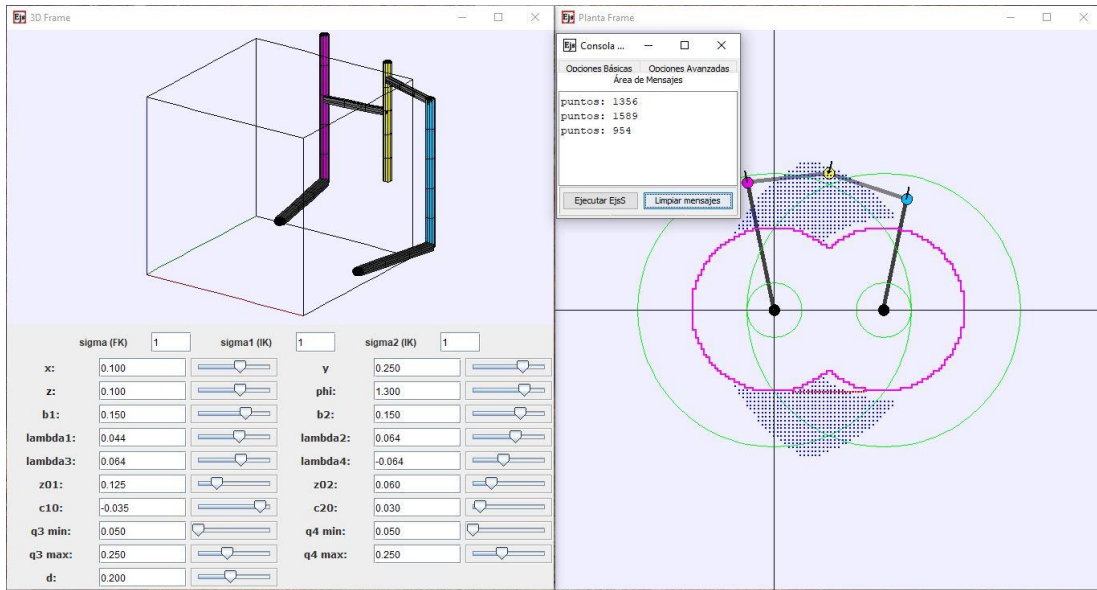
- $b_2 = 0.02m \rightarrow$ mayor espacio de trabajo (1424 puntos)



- Parámetro $\lambda_1 = 0.064 m/rad$ (1356 puntos)
- $\lambda_1 + 0.02 m/rad \rightarrow$ mayor espacio de trabajo (1589 puntos)

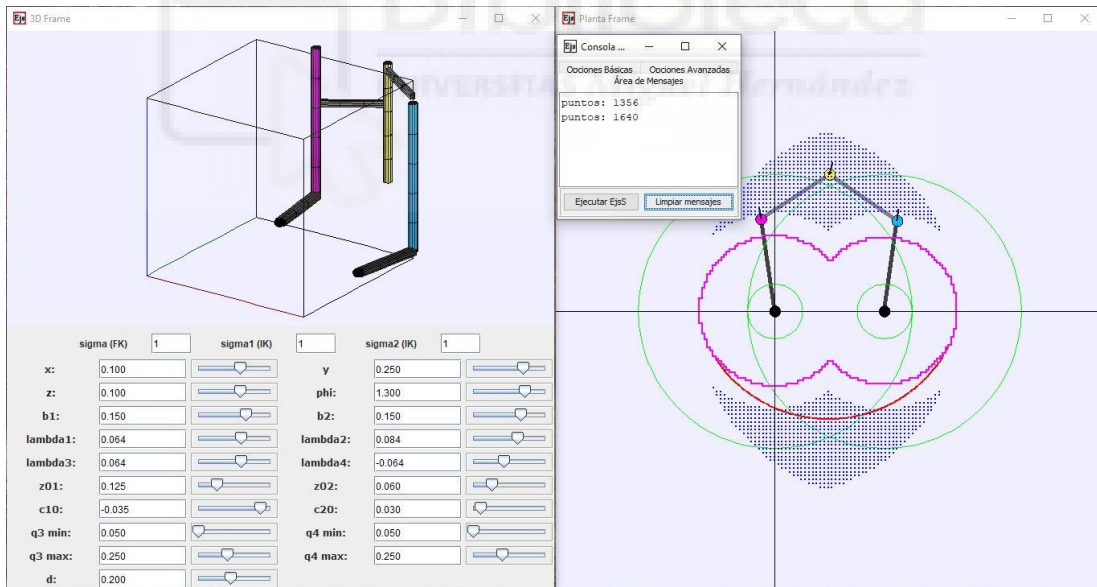


- $\lambda_1 - 0.02 \text{ m/rad} \rightarrow$ menor espacio de trabajo (954 puntos)

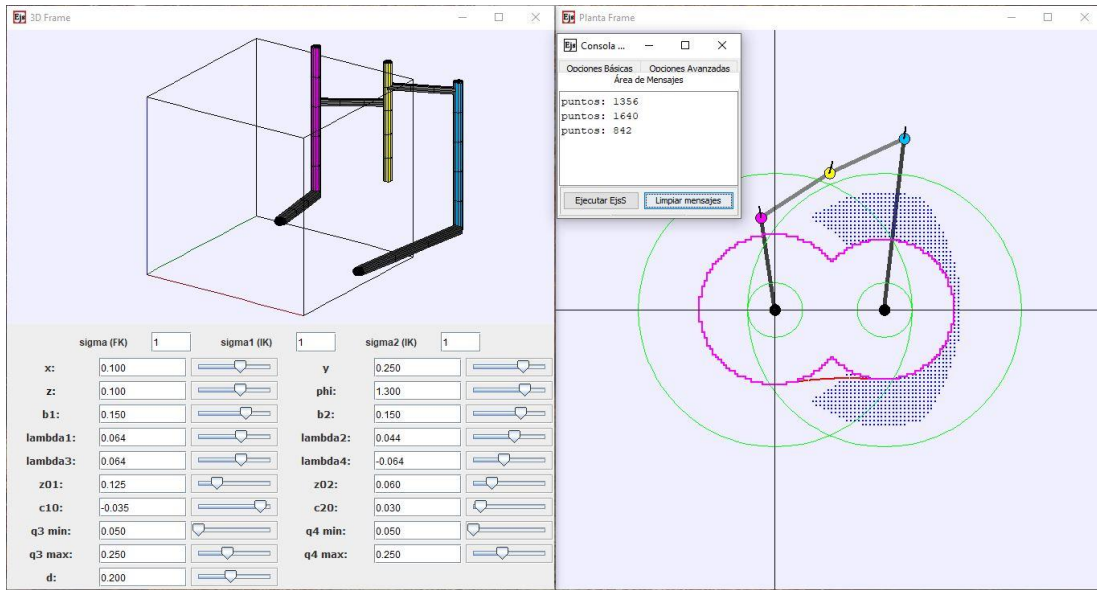


- Parámetro $\lambda_2 = 0.064 \text{ m/rad}$ (1356 puntos)

- $\lambda_2 + 0.02 \text{ m/rad} \rightarrow$ mayor espacio de trabajo (1640 puntos)

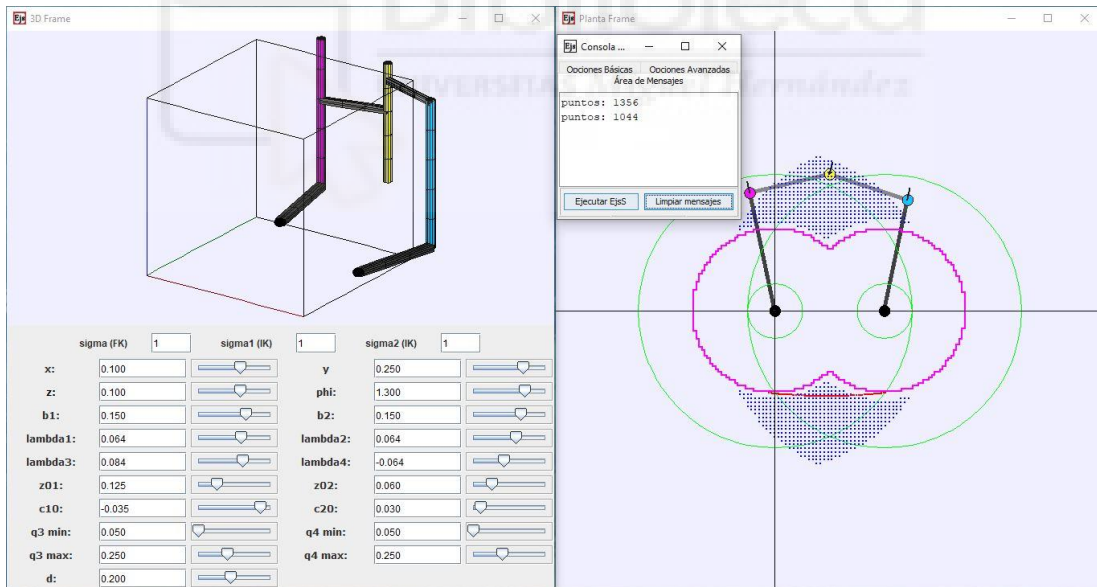


- $\lambda_2 = 0.02 \text{ m/rad} \rightarrow$ menor espacio de trabajo (842 puntos)

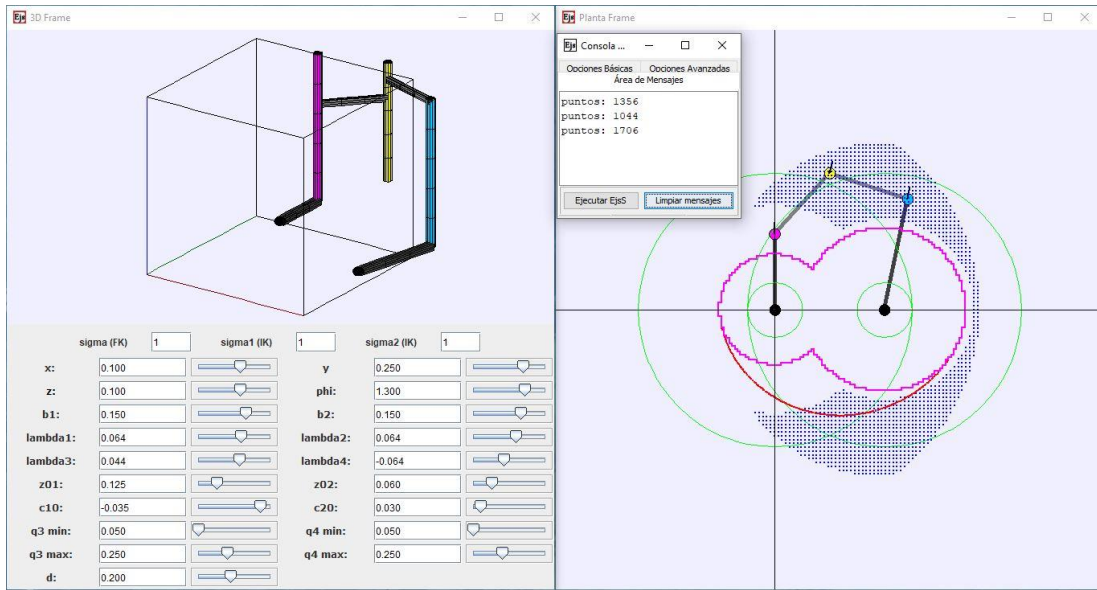


- Parámetro $\lambda_3 = 0.064 \text{ m/rad}$ (1356 puntos)

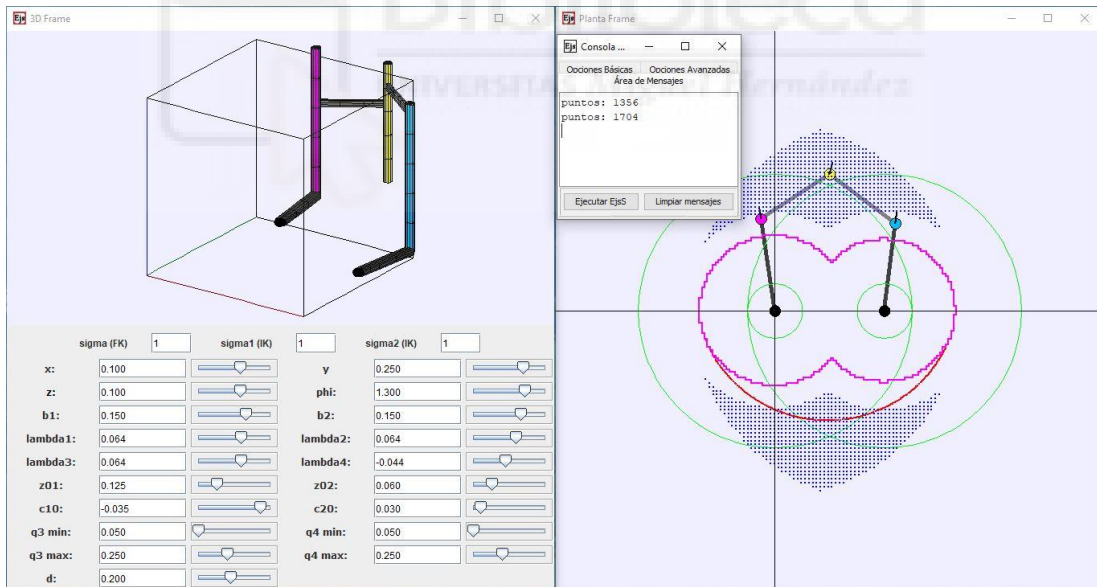
- $\lambda_3 + 0.02 \text{ m/rad} \rightarrow$ menor espacio de trabajo (1044 puntos)



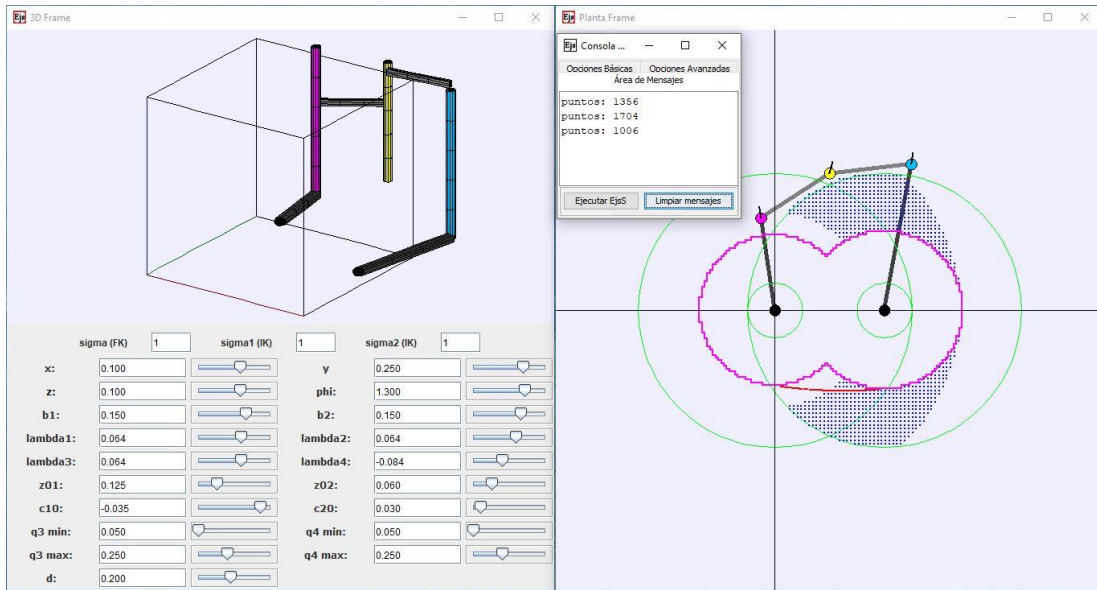
- $\lambda_3 - 0.02 \text{ m/rad} \rightarrow$ mayor espacio de trabajo (1706 puntos)



- Parámetro $\lambda_4 = -0.064 \text{ m/rad}$ (1356 puntos)
- $\lambda_4 + 0.02 \text{ m/rad} \rightarrow$ mayor espacio de trabajo (1704 puntos)

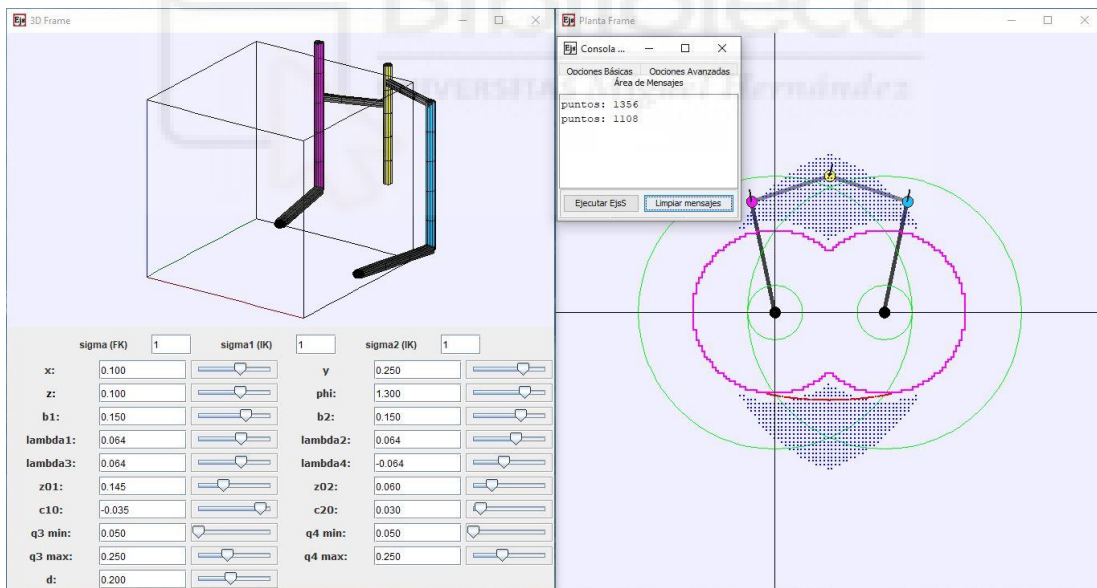


- $\lambda_4 = 0.02 \text{ m/rad} \rightarrow$ menor espacio de trabajo (1006 puntos)

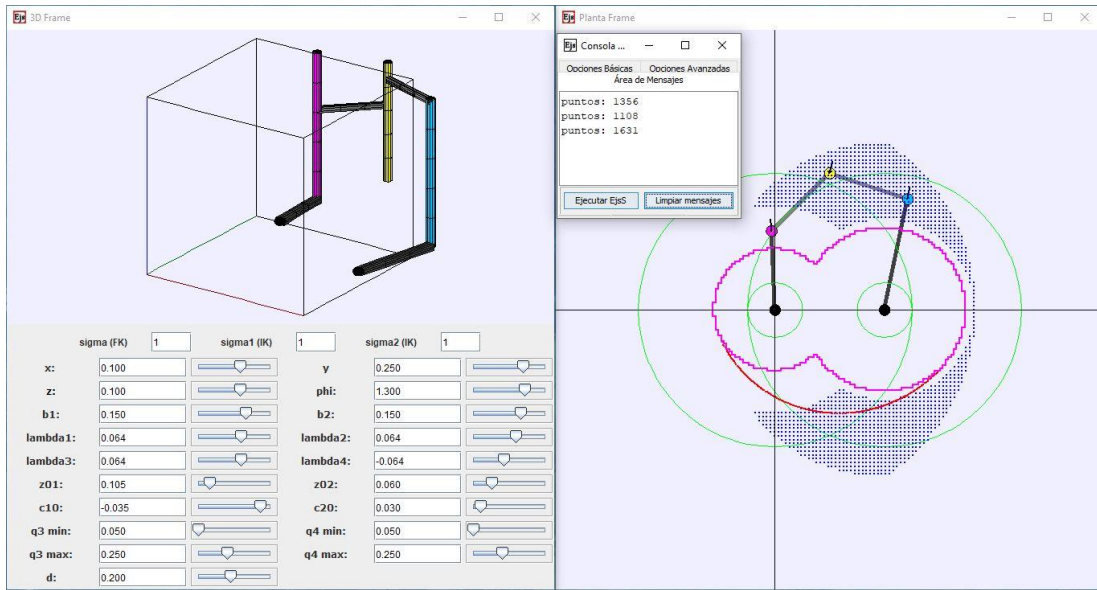


- Parámetro $z_{01} = 0.125 \text{ m}$ (1356 puntos)

- $z_{01} + 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1108 puntos)

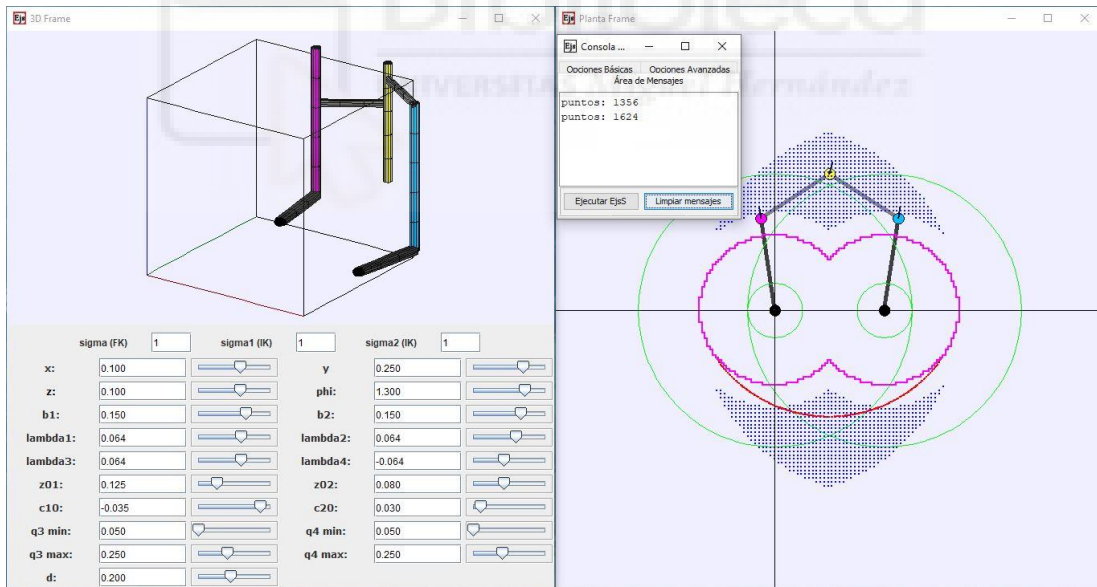


- $z_{01} - 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1631 puntos)

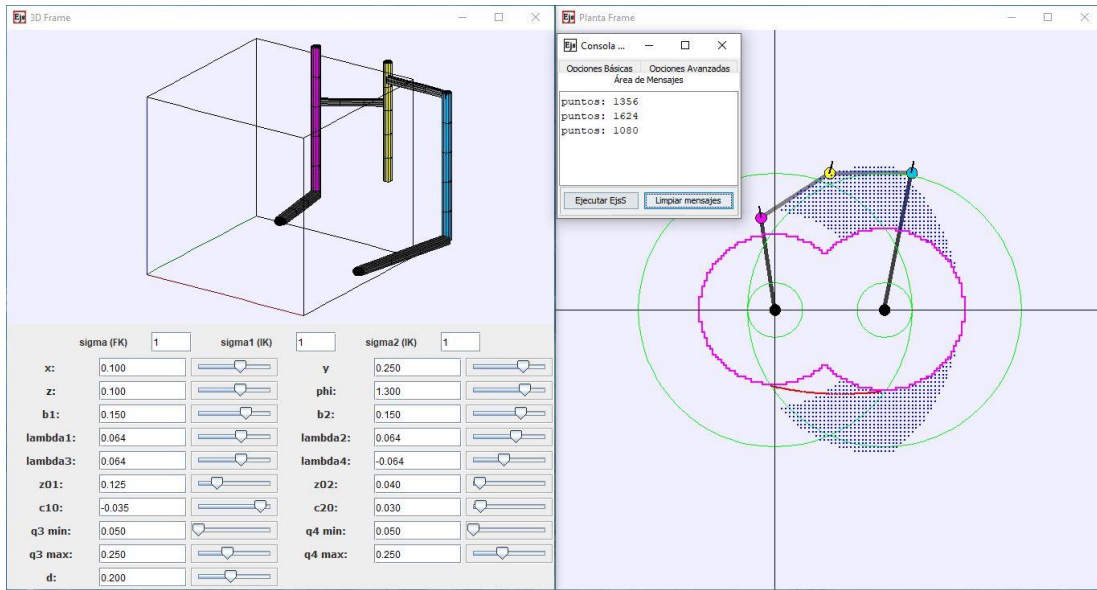


- Parámetro $z_{02} = 0.06 \text{ m}$ (1356 puntos)

- $z_{02} + 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1624 puntos)

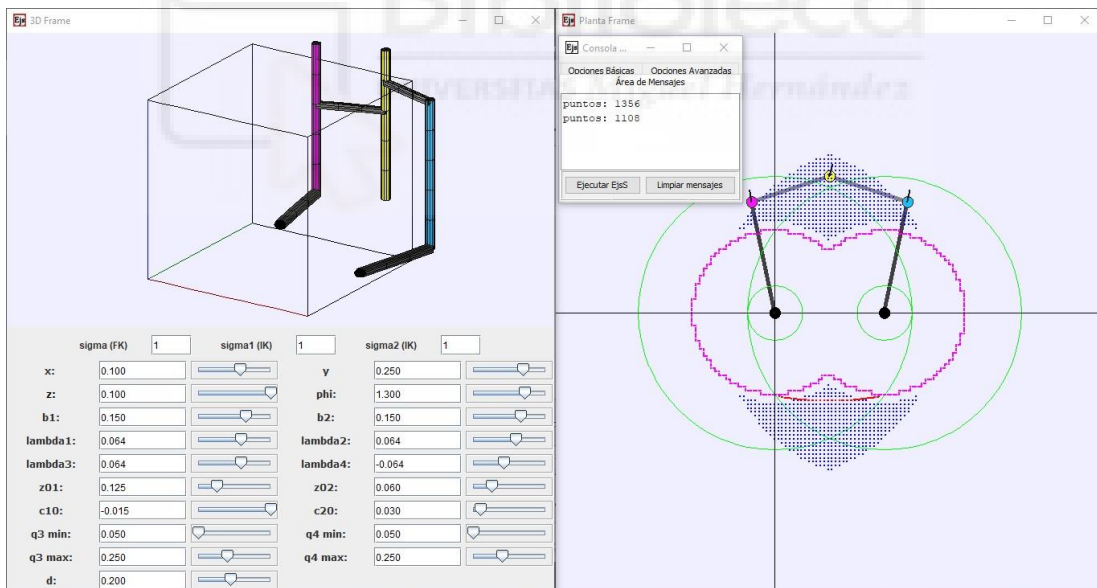


- $z_{02} - 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1080 puntos)

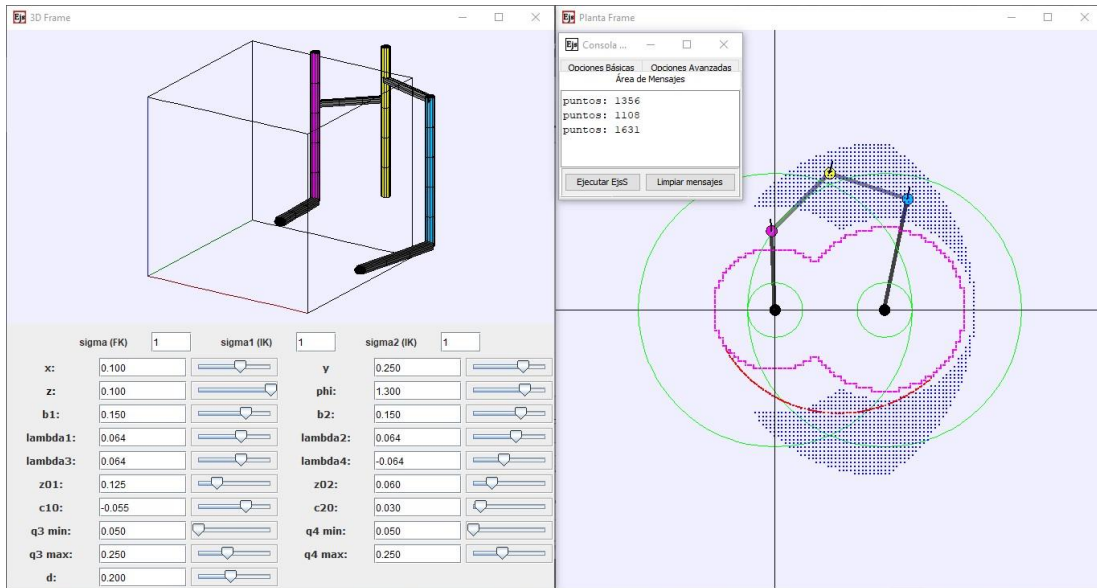


- Parámetro $c_{10} = -0.035$ (1356 puntos)

- $c_{10} + 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1108 puntos)

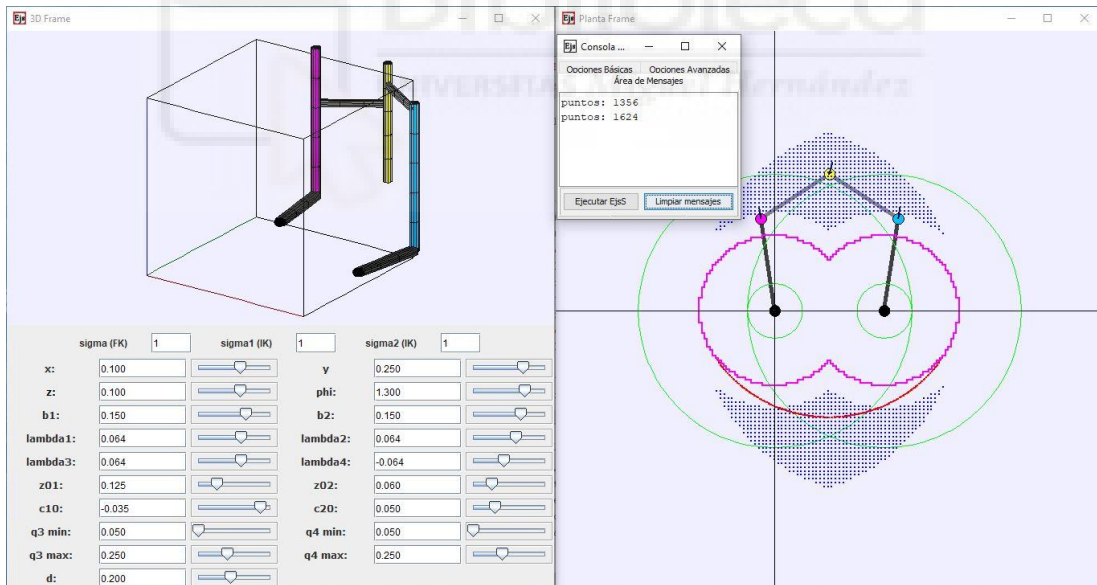


- $c_{10} - 0.02\text{ m} \rightarrow$ mayor espacio de trabajo (1631 puntos)

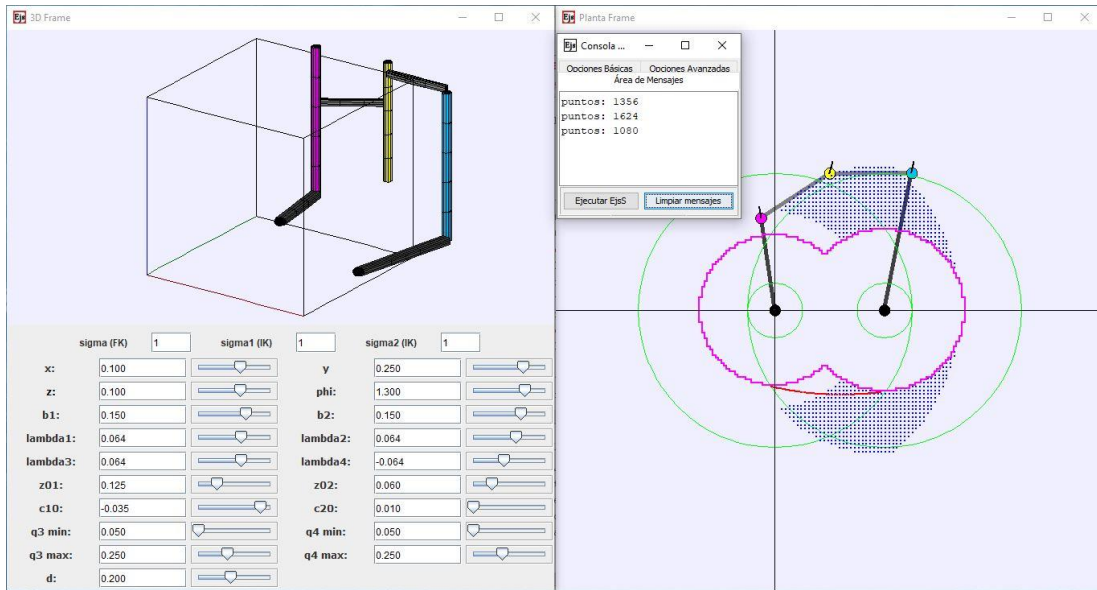


- Parámetro $c_{20} = 0.03\text{ m}$ (1356 puntos)

- $c_{20} + 0.02\text{ m} \rightarrow$ mayor espacio de trabajo (1624 puntos)

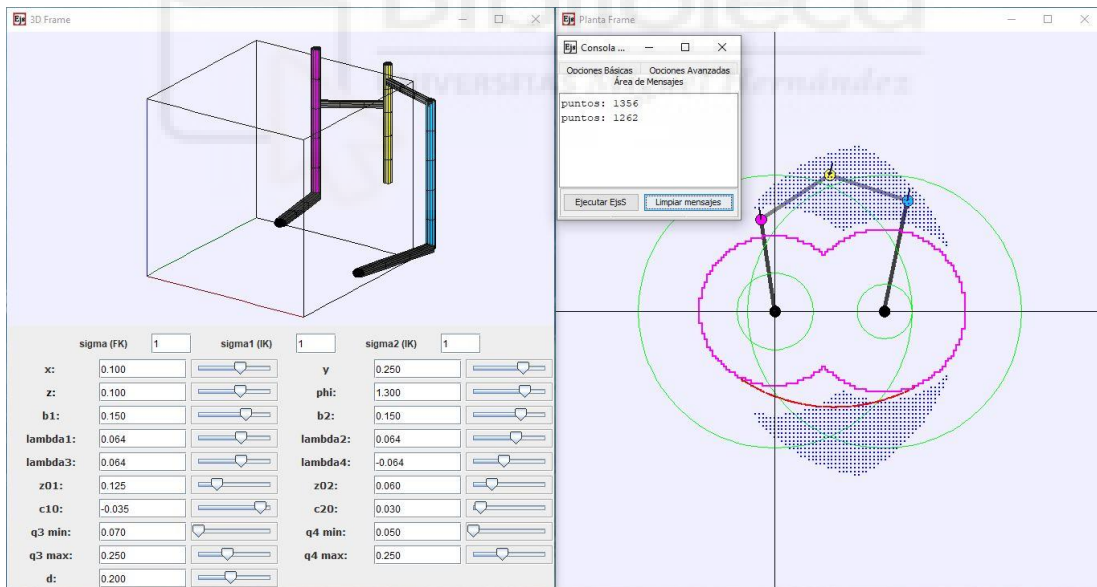


- $c_{20} = 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1080 puntos)

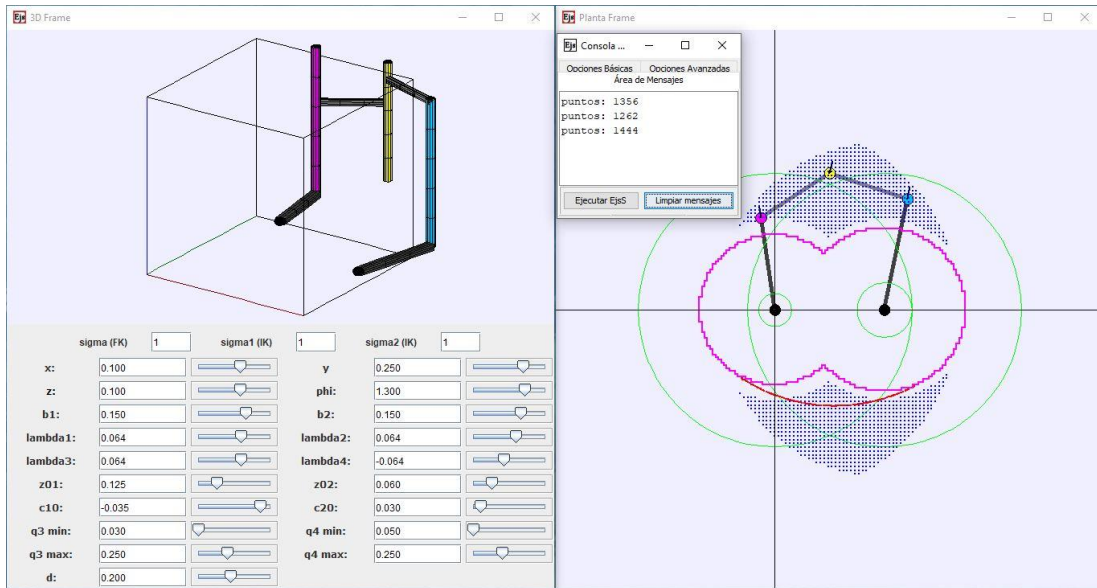


- Parámetro $q_{3_min} = 0.05 \text{ m}$ (1356 puntos)

- $q_{3_min} + 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1262 puntos)

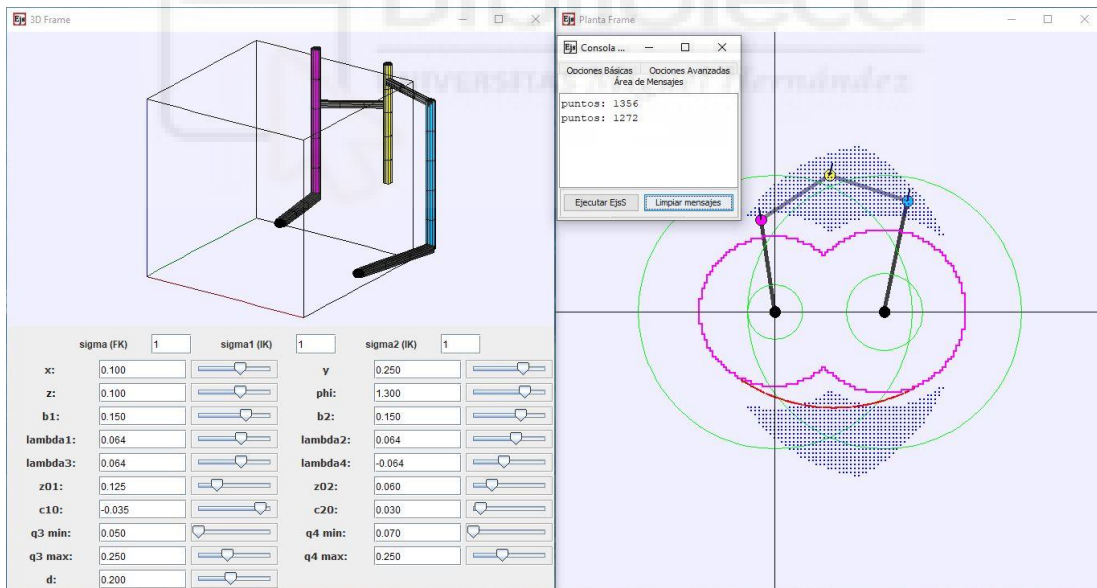


- $q_{3_min} - 0.02\text{ m} \rightarrow$ mayor espacio de trabajo (1444 puntos)

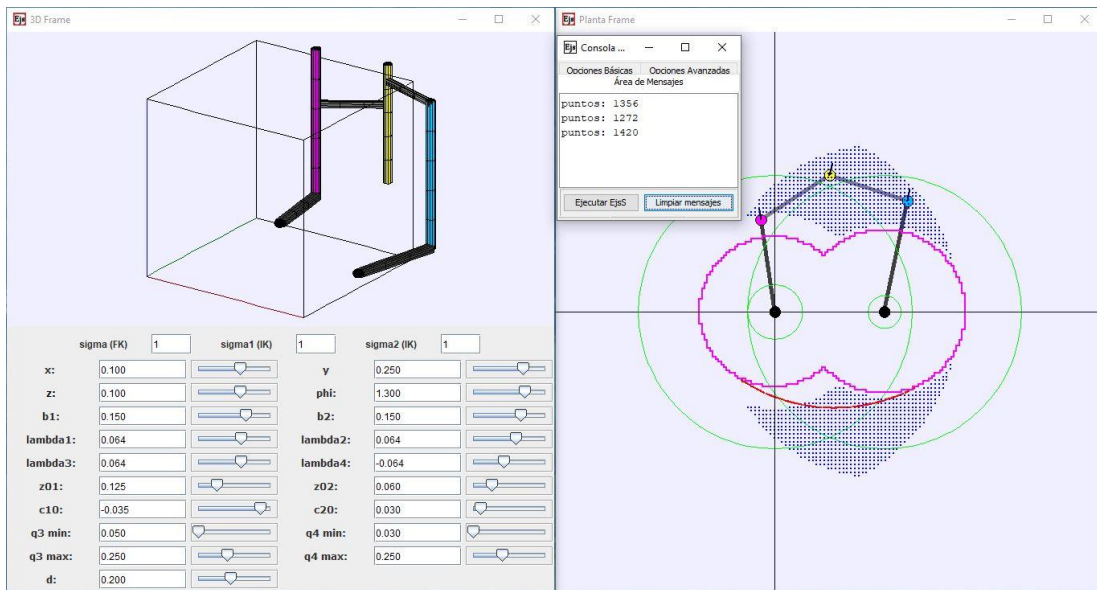


- Parámetro $q_{4_min} = 0.05\text{ m}$ (1356 puntos)

- $q_{4_min} + 0.02\text{ m} \rightarrow$ menor espacio de trabajo (1272 puntos)

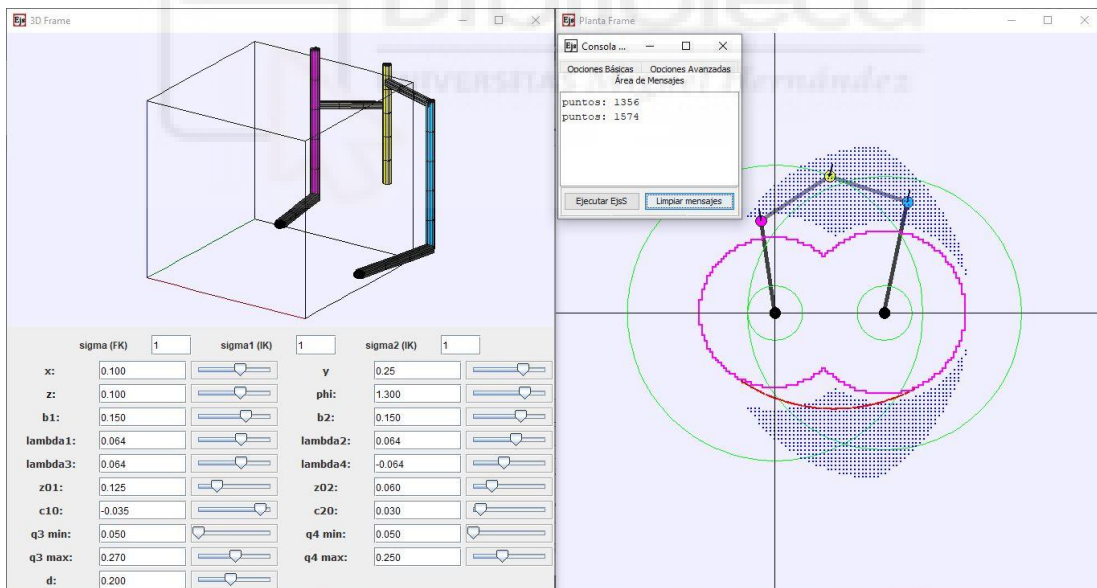


- $q_{4_min} = 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1420 puntos)

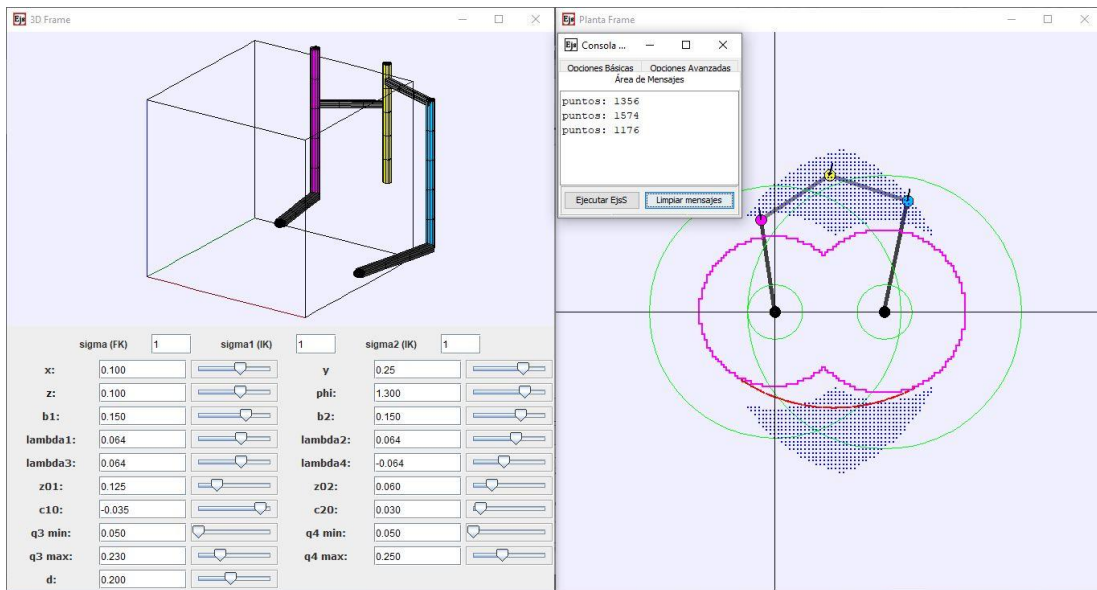


- Parámetro $q_{3_max} = 0.25 \text{ m}$ (1356 puntos)

- $q_{3_max} + 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1574 puntos)

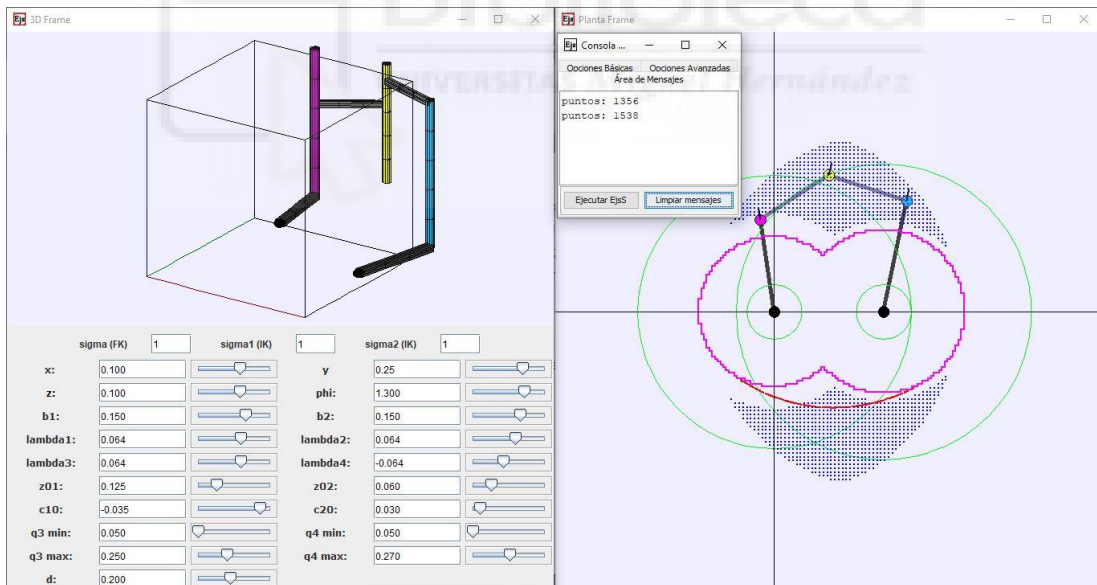


- $q_{3_max} - 0.02\text{ m} \rightarrow$ menor espacio de trabajo (1176 puntos)

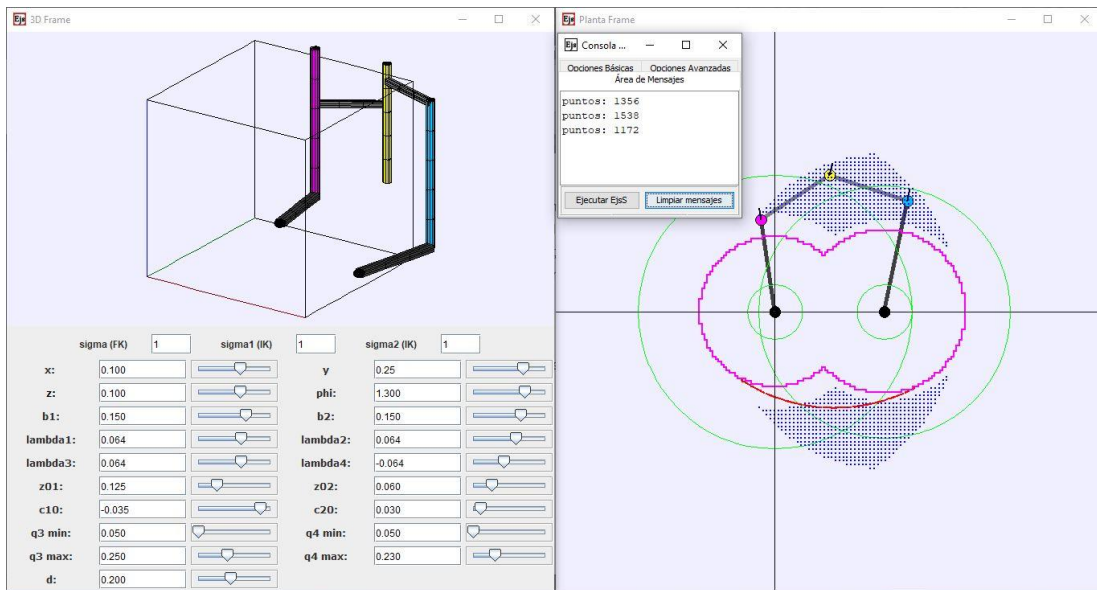


- Parámetro $q_{4_max} = 0.25\text{ m}$ (1356 puntos)

- $q_{4_max} + 0.02\text{ m} \rightarrow$ mayor espacio de trabajo (1538 puntos)

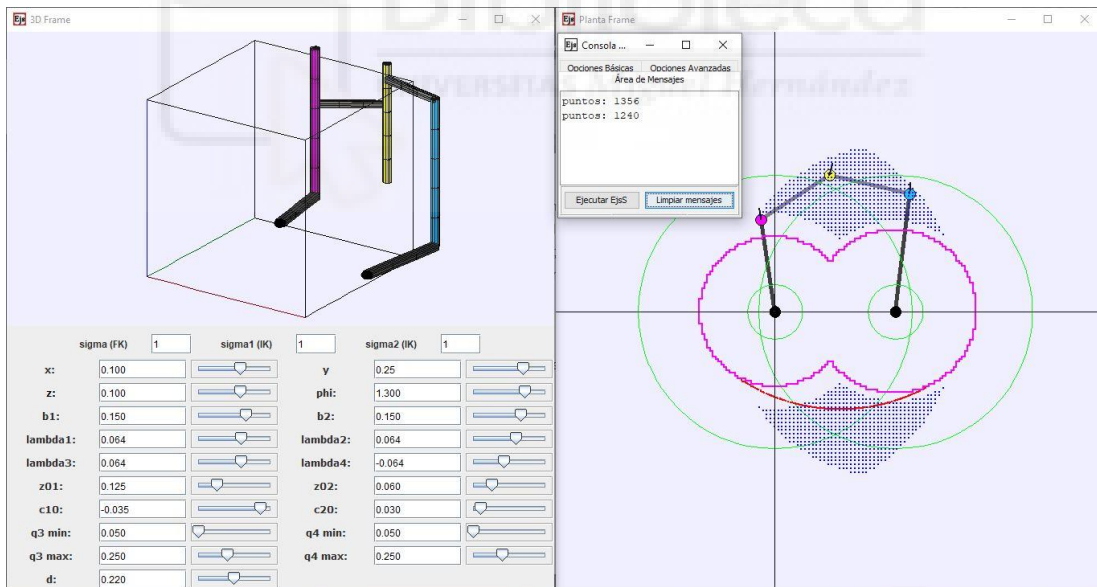


- $q_{4_max} = 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1172 puntos)

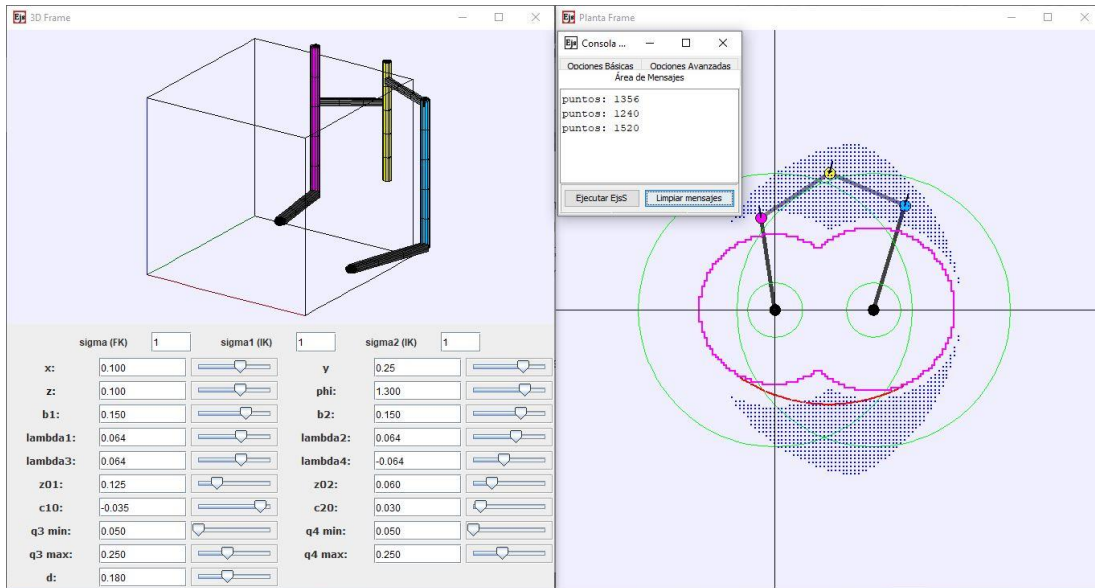


- Parámetro $d = 0.2 \text{ m}$ (1356 puntos)

- $d + 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1240 puntos)



- $d = 0.02 \text{ m} \rightarrow$ mayor espacio de trabajo (1520 puntos)

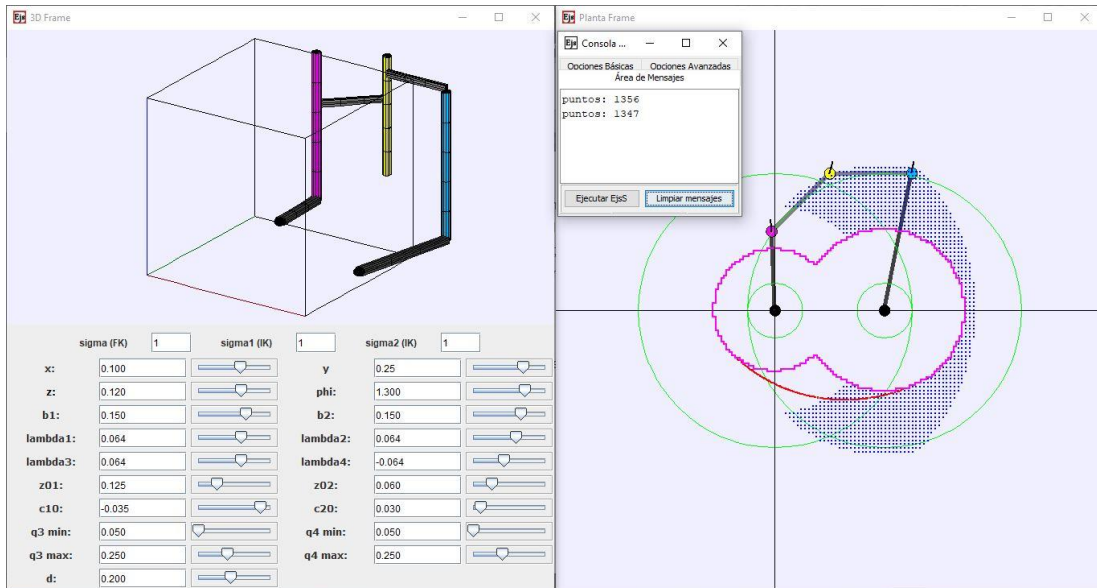


Dado los resultados obtenidos, se distinguen dos tipos de parámetros que incrementan el espacio de trabajo cuando:

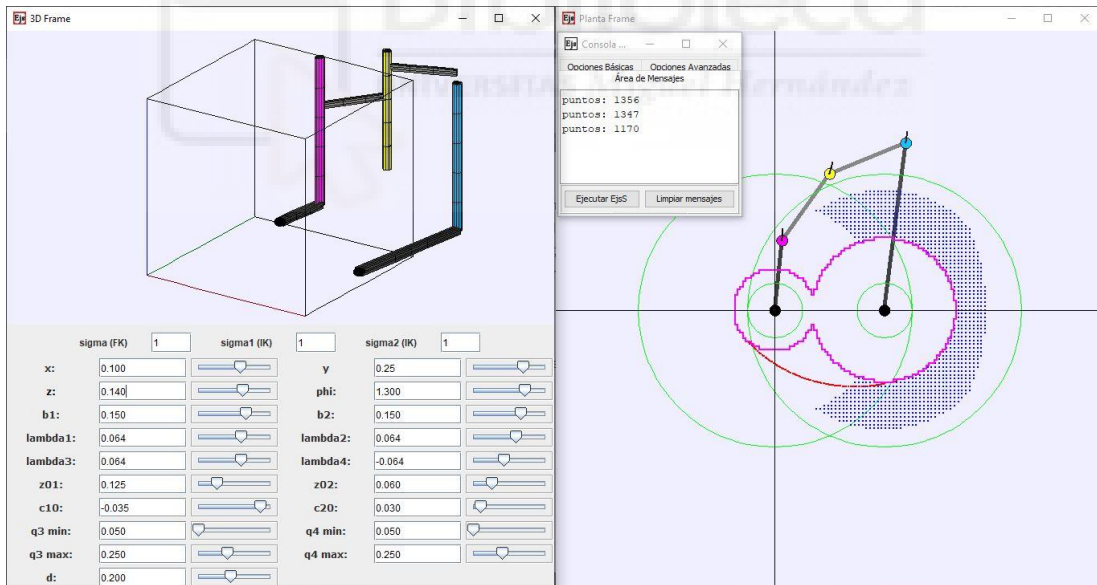
- Incrementan su valor: $b_1, \lambda_1, \lambda_2, \lambda_4, z_{02}, c_{20}, q_{3_max}, q_{4_max}$
- Decrementan su valor: $b_2, \lambda_3, z_{01}, c_{10}, q_{3_min}, q_{4_min}, d$

A continuación, se estudia también el efecto de las coordenadas operacionales z y φ que, al igual que el resto de parámetros geométricos, hacen variar el espacio de trabajo según sus valores.

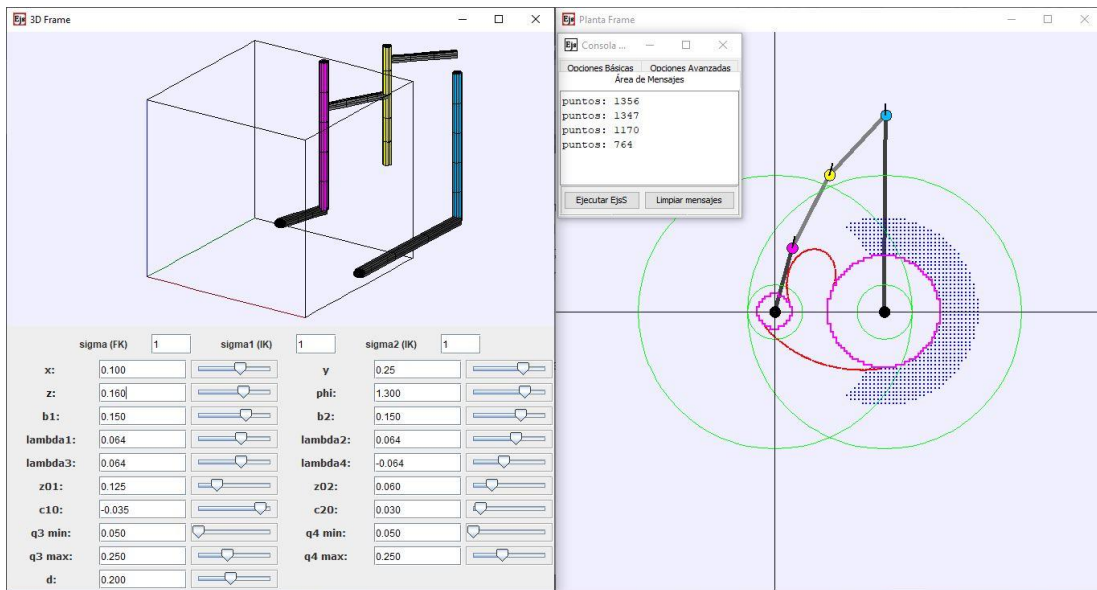
- Parámetro $z = 0.1\text{ m}$ (1356 puntos)
 - $z + 0.02\text{ m} \rightarrow$ menor espacio de trabajo (1347 puntos)



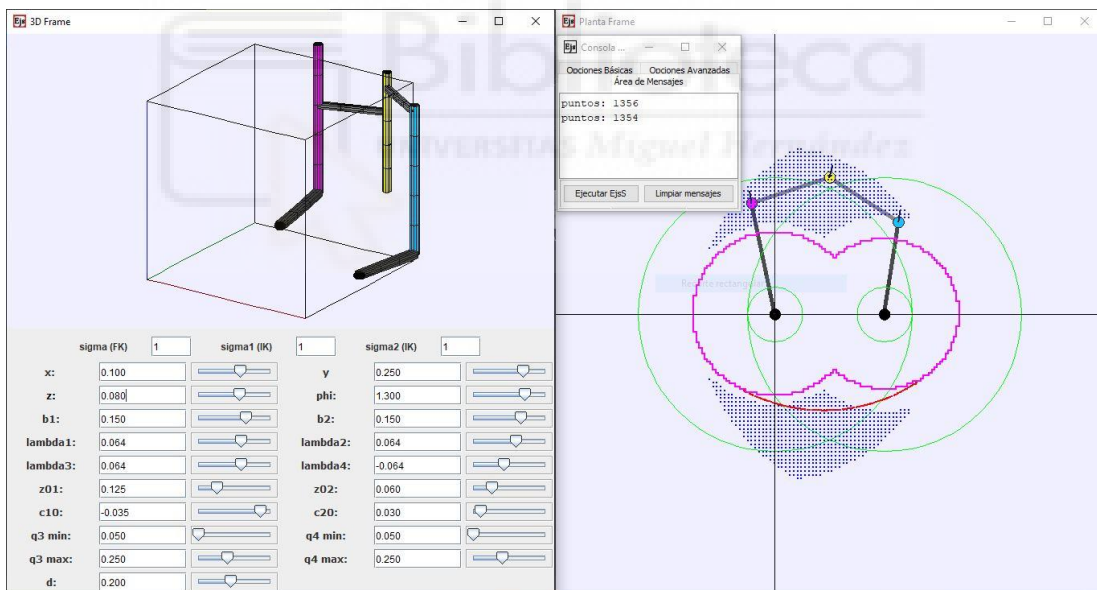
- $z + 0.04\text{ m} \rightarrow$ menor espacio de trabajo (1170 puntos)



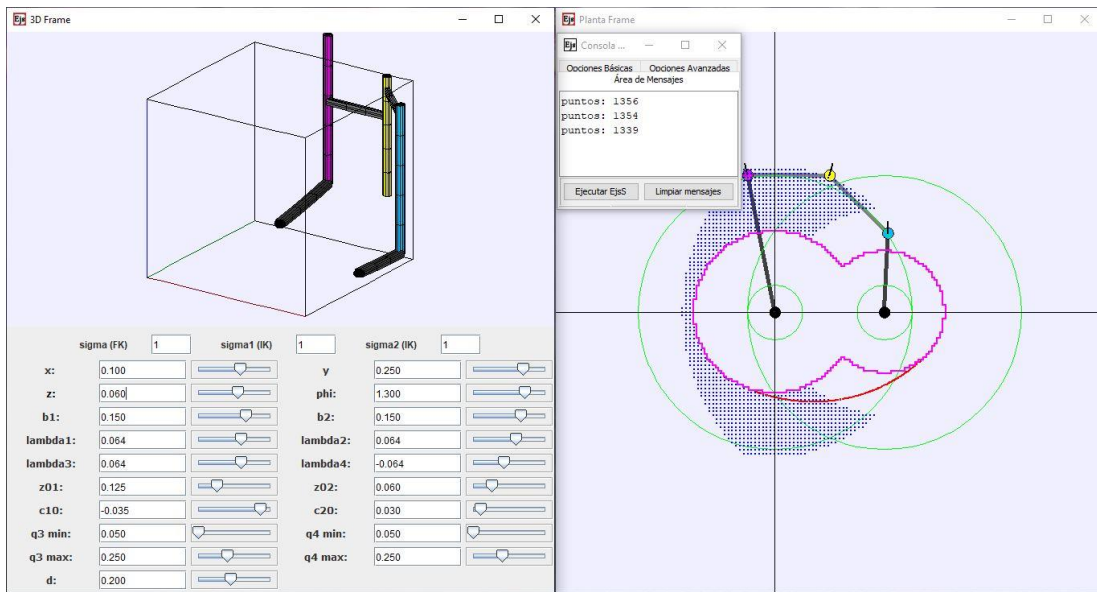
▫ $z + 0.06 \text{ m} \rightarrow$ menor espacio de trabajo (764 puntos)



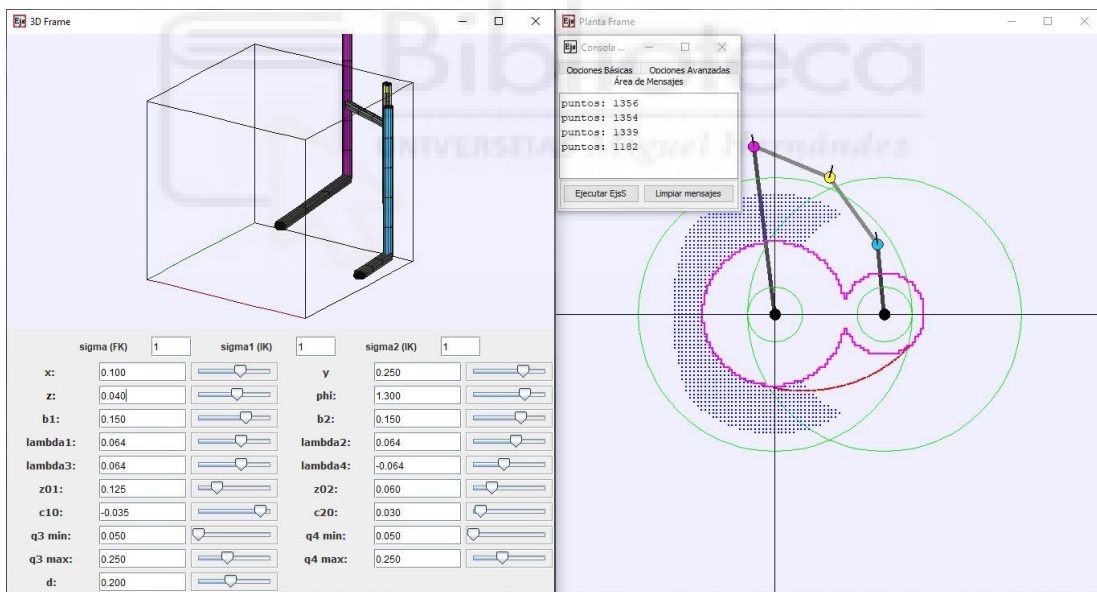
▫ $z - 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1354 puntos)



- $z = 0.04 \text{ m}$ → menor espacio de trabajo (1339 puntos)

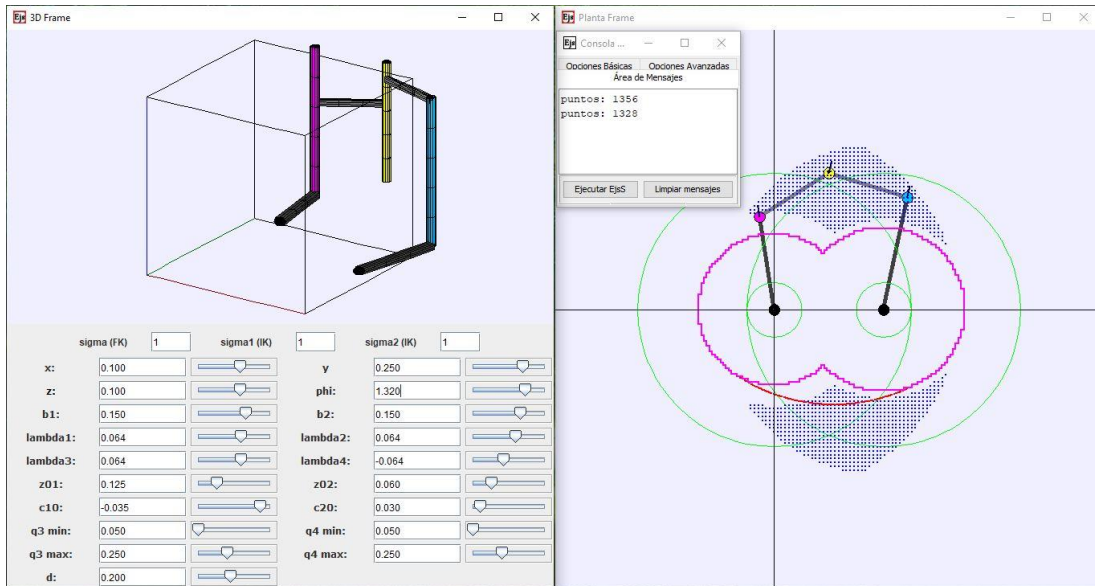


- $z = 0.06 \text{ m}$ → menor espacio de trabajo (1182 puntos)

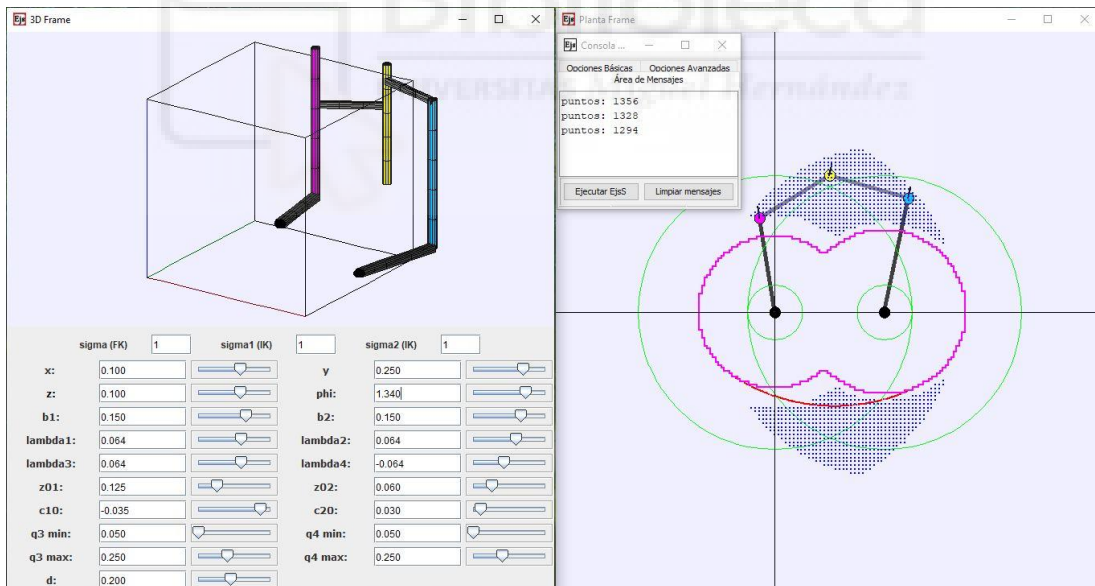


Como se puede ver, el espacio de trabajo variando z siempre disminuye, por lo que el valor actual para el análisis de sensibilidad es el más adecuado.

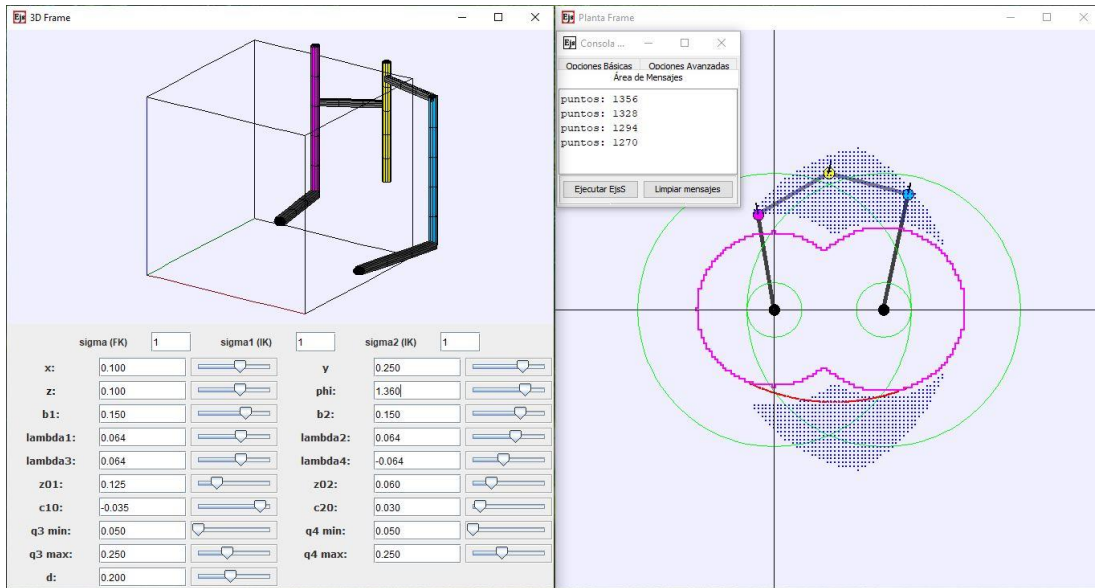
- Parámetro $\varphi = 1.3 \text{ rad}$ (1356 puntos)
 - $\varphi + 0.02 \text{ m} \rightarrow$ menor espacio de trabajo (1328 puntos)



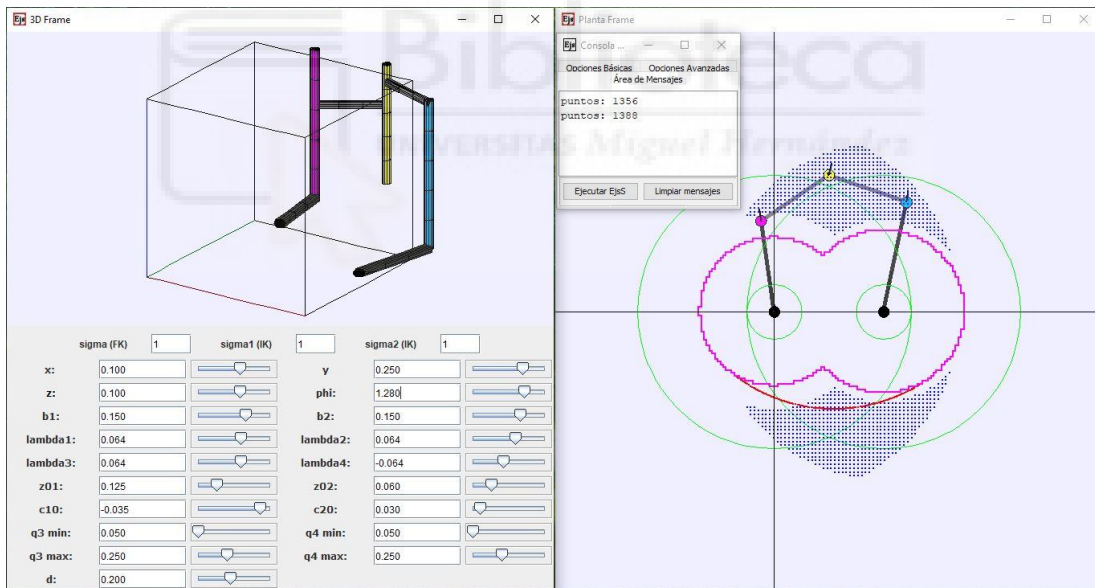
- $\varphi + 0.04 \text{ m} \rightarrow$ menor espacio de trabajo (1294 puntos)



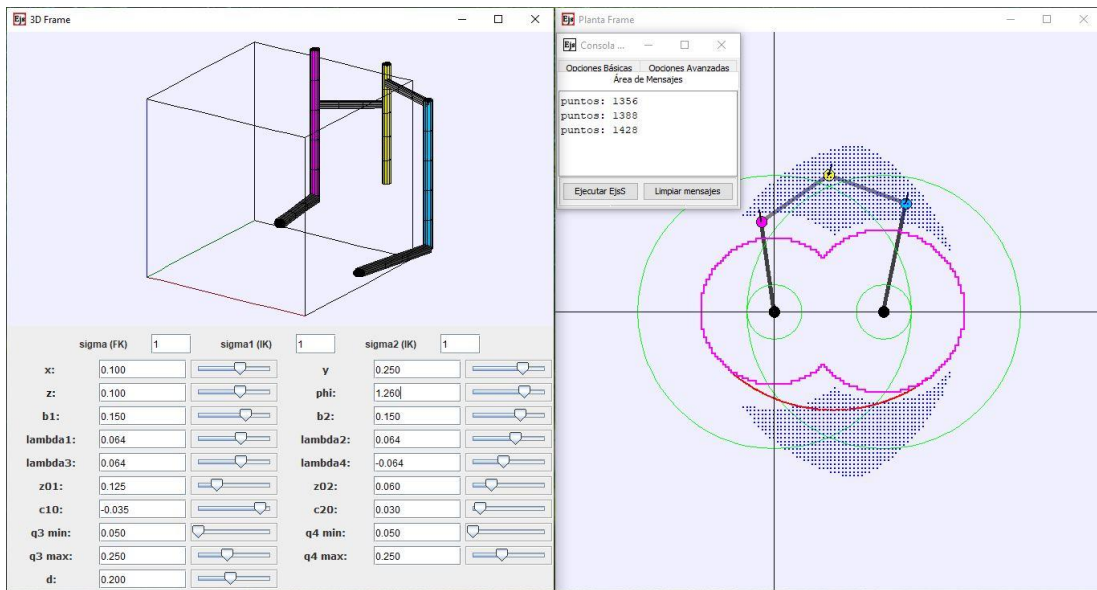
▫ $\varphi + 0.06\text{ m} \rightarrow$ menor espacio de trabajo (1270 puntos)



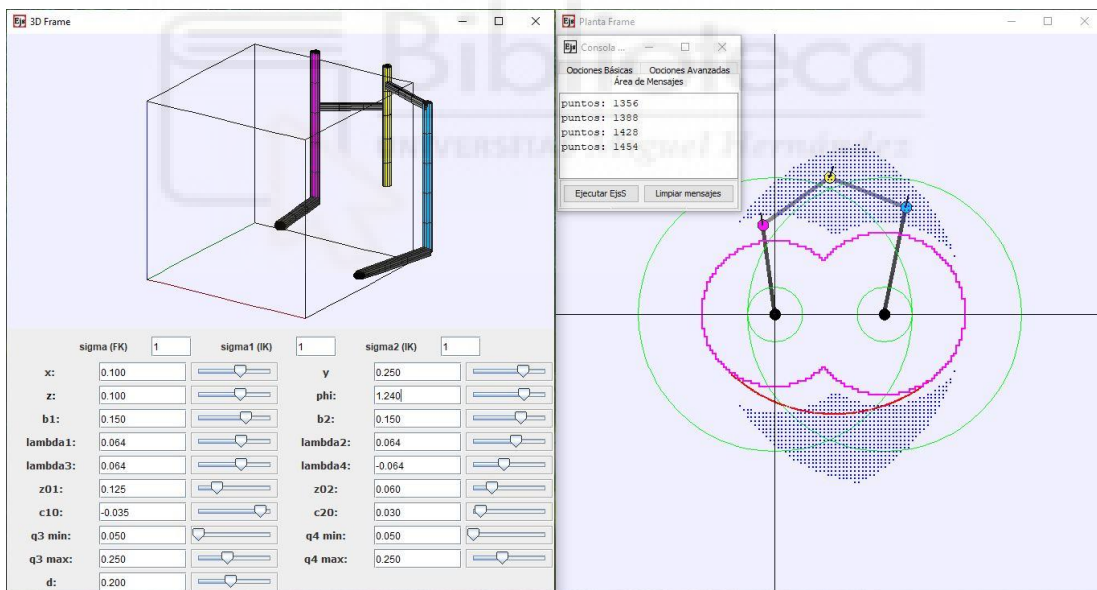
▫ $\varphi - 0.02\text{ m} \rightarrow$ mayor espacio de trabajo (1388 puntos)



- $\varphi = 0.04 \text{ m} \rightarrow$ mayor espacio de trabajo (1428 puntos)



- $\varphi = 0.06 \text{ m} \rightarrow$ mayor espacio de trabajo (1454 puntos)



La variable φ incrementa el espacio de trabajo cuando su valor disminuye, por lo que, para este diseño inicial del análisis de sensibilidad, el valor no es el adecuado.

Anexo 3. Código implementado en Matlab

Las funciones implementadas para la optimización del espacio de trabajo en el capítulo 4 se muestran a continuación.

- Espacio de trabajo 2D

```
function espacio = esp_trabajo2d(dis)
```

```
% Asignacion de los valores iniciales de los parametros geometricos
```

```
b1 = dis(1);
```

```
b2 = dis(2);
```

```
lambda1 = dis(3);
```

```
lambda2 = dis(4);
```

```
lambda3 = dis(5);
```

```
lambda4 = dis(6);
```

```
z01 = dis(7);
```

```
z02 = dis(8);
```

```
c10 = dis(9);
```

```
c20 = dis(10);
```

```
d = dis(11);
```

```
% Variables fijas
```

```
sigma1 = 1;
```

```
sigma2 = 1;
```

```
q3_min = 0.17;
```

```
q3_max = 0.37;
```

```
q4_min = 0.17;
```

```
q4_max = 0.37;
```

```
z = 0;
```

```
phi = 0;
```

```
x_min= -0.70;
```

```
x_max= 0.70;
```

```
y_min= -0.70;
```

```
y_max= 0.70;
```

```
Nx= 201;
```

```
Ny= 201;
```

```
dx = (x_max - x_min) / (Nx-1);
```

```
dy = (y_max - y_min) / (Ny-1);
```

```
espacio = 0;
```

```
x = x_min;
```

```
contador=1;
```

```
mallá = [ ];
```

```
% Líneas del Algoritmo 1 para el cálculo del espacio de trabajo 2D
```

```
for i = 1:Nx;
```

```
    y = y_min;
```

```
    for j = 1:Ny;
```

```
        [q1, q2, q3, q4, ~, ~] = cinematica_inversa(x, y, z, phi, d, b1, b2, z01, z02, c10,  
            c20, lambda1, lambda2, lambda3, lambda4, sigma1,  
            sigma2);
```

```
        if isreal(q1) & isreal(q2) & isreal(q3) & isreal(q4)
```

```
            if q3>=q3_min && q3<=q3_max && q4>=q4_min && q4<=q4_max
```

```
                mallá(contador, 1) = x;
```

```
                mallá (contador, 2) = y;
```

```
                contador = contador+1;
```

```
            end
```

```
        end
```

```
        y = y+dy;
```

```
    end
```

```
    x = x+dx;
```

```
end
```

```
% Código para visualizar el espacio de trabajo 2D
```

```
% scatter(mallá (:,1), mallá (:,2), 5)
```

```
% axis([-0.7 0.7 -0.7 0.7])
```

```
espacio = -(contador-1);
```

```
end
```

- Espacio de trabajo 4D

```
function espacio = esp_trabajo4d(dis)
```

```
% Asignacion de los valores iniciales de los parametros geometricos
```

```
b1 = dis(1);
```

```
b2 = dis(2);
```

```
lambda1 = dis(3);
```

```
lambda2 = dis(4);
```

```
lambda3 = dis(5);
```

```
lambda4 = dis(6);
```

```
z01 = dis(7);
```

```
z02 = dis(8);
```

```
c10 = dis(9);
```

```
c20 = dis(10);
```

```
d = dis(11);
```

```
% Variables fijas
```

```
sigma1 = 1;
```

```
sigma2 = 1;
```

```
q3_min = 0.17;
```

```
q3_max = 0.37;
```

```
q4_min = 0.17;
```

```
q4_max = 0.37;
```

```
x_min= -0.70;
```

```
x_max= 0.70;
```

```
y_min= -0.70;
```

```
y_max= 0.70;
```

```
z_min= -0.70;
```

```
z_max= 0.70;
```

```
phi_min= -pi;
```

```
phi_max= pi;
```

```
Nx= 17;
```

```
Ny= 17;
```

```
Nz= 17;
```

```
Nphi= 17;
```

```
dx = (x_max - x_min) / (Nx-1);
```

```
dy = (y_max - y_min) / (Ny-1);
```

```
dz = (z_max - z_min) / (Nz-1);
```

```
dphi = (phi_max - phi_min) / (Nphi-1);
```

```
espacio = 0;
```

```
phi = phi_min;
```

```
contador = 1;
```

```
mallas = [];
```

```
% Algoritmo 1
```

```
for m = 1:Nphi;
```

```
    z = z_min;
```

```
    for k = 1:Nz;
```

```
        x = x_min;
```

```
        for i = 1:Nx;
```

```
            y = y_min;
```

```
            for j = 1:Ny;
```

```
                [q1, q2, q3, q4, ~, ~] = cinematica_inversa(x, y, z, phi, d, b1, b2, z01, z02,  
                    c10, c20, lambda1, lambda2, lambda3, lambda4,  
                    sigma1, sigma2);
```

```
                if isreal(q1) & isreal(q2) & isreal(q3) & isreal(q4)
```

```
                    if q3>=q3_min && q3<=q3_max && q4>=q4_min && q4<=q4_max
```

```
                        mallas(contador, 1) = x;
```

```
                        mallas(contador, 2) = y;
```

```
                        mallas(contador, 3) = z;
```

```
                        mallas(contador, 4) = phi;
```

```
                        contador = contador+1;
```

```
                    end
```

```
                end
```

```
                y = y+dy;
```

```
            end
```

```
            x = x+dx;
```

```
        end
```

```

    z=z+dz;
end
phi=phi+dphi;
end
% Código para visualizar el espacio de trabajo 4D
% scatter3(malla(:,1),malla(:,2), malla(:, 3), 3)
% axis([-0.7 0.7 -0.7 0.7 -0.7 0.7])

% Código para visualizar el espacio de trabajo 2D en el plano XY
% scatter(malla(:,1),malla(:,2), 3)
% axis([-0.7 0.7 -0.7 0.7])

% % Código para visualizar el espacio de trabajo 2D en el plano YZ
% scatter(malla(:,2),malla(:,3), 3)
% axis([-0.7 0.7 -0.7 0.7])

```

```

    espacio = -(contador-1);
end

```

- Cinemática inversa

```

function [q1, q2, q3, q4, phi1, phi2] = cinematica_inversa(x, y, z, phi, d, b1, b2, z01,
z02, c10, c20, lambda1, lambda2, lambda3, lambda4, sigma1, sigma2)

```

```

psi1 = (z - z01 - phi*lambda3 - c10) / (lambda1);
psi2 = (z - z02 - phi*lambda4 - c20) / (lambda2);

```

```

%Resolucion de q1, q3

```

```

q3 = sigma1*sqrt(- b1*b1*sin(psi1)*sin(psi1) + x*x + y*y) - b1*cos(psi1);

```

```

if isreal(q3)

```

```

    cos_q1 = (b1*x*cos(psi1) + b1*y*sin(psi1) + q3*x) /
(2*b1*q3*cos(psi1) + b1*b1 + q3*q3);

```

```

    sin_q1 = (b1*y*cos(psi1) - b1*x*sin(psi1) + q3*y) /
(2*b1*q3*cos(psi1) + b1*b1 + q3*q3);

```

```

    q1 = atan2(sin_q1, cos_q1);

```

```

    phi1 = q1 + psi1;

```

```

else
    q1=0+1i;
    phi1=0;
end

%Resolucion de q2, q4
q4 = - b2*cos(psi2) + sigma2 *
    sqrt(- b2*b2*sin(psi2)*sin(psi2) + x*x - 2*d*x + y*y + d*d);
if isreal(q4)
    cos_q2 = (b2*(x-d) * cos(psi2) + b2*y*sin(psi2) + q4*(x-d)) /
        (2*b2*q4*cos(psi2) + b2*b2 + q4*q4);
    sin_q2 = (b2*y*cos(psi2) + b2*(d-x) * sin(psi2) + q4*y) /
        (2*b2*q4*cos(psi2) + b2*b2 + q4*q4);
    q2 = atan2(sin_q2, cos_q2);
    phi2 = q2 + psi2;
else
    q2=0+1i;
    phi2=0;
end
end
end

```

