Diseño de Robot Delta

Roberto Carlos Escobar Salcedo
8vo Semestre
roberto.escobar@ucb.edu.bo

Sebastian Osorio Ferrufino
8vo Semestre
sebastian.osorio@ucb.edu.bo

Matías adorno Mazocato
8vo Semestre
matias.adorno@ucb.edu.bo

Jose Jesus Cabrera Pantoja
Robótica IMT-342
Departamento ingeniería mecatrónica

Resumen **— Este proyecto presenta el diseño, construcción y control de un robot delta, en el cual se llevaron a cabo cálculos de cinemática directa e inversa para determinar posiciones y ángulos articulares. Se realizaron simulaciones estructurales, incluyendo análisis de tensión nodal, desplazamiento estático y deformación unitaria para evaluar el comportamiento mecánico del robot. Se realizó el análisis del espacio de trabajo de este robot y se identificaron sus rangos de movimiento. El control de este robot se realizó mediante la placa ESP32 que se encargó de la captura de datos de la posición angular de los motores y de su movimiento. El proceso incluyó el diseño estructural, la selección de materiales y componentes, y la implementación de pruebas experimentales para validar el comportamiento del sistema mediante la comparación entre valores teóricos y mediciones de los valores reales.**

Palabras clave: Robot delta, SCP32, Cinemática inversa, Cinemática directa, servomotores.

Abstract **— This project presents the design, construction and control of a delta robot, in which forward and inverse kinematics calculations were carried out to determine joint positions and angles. Structural simulations were performed, including nodal stress analysis, static displacement and unitary deformation to evaluate the mechanical behavior of the robot. The workspace of this robot was analyzed, and its ranges of motion were identified. The control of this robot was carried out by the ESP32 board that was responsible for capturing data on the angular position of the motors and their movement. The process included the structural design, the selection of materials and components, and the implementation of experimental tests to validate the behavior of the system by comparing theoretical values with measurements of the real values.**

I. INTRODUCCIÓN

La robótica ha avanzado considerablemente en los últimos años, impulsada por la necesidad de optimizar procesos en industrias que requieren rapidez, precisión y eficiencia. En este contexto, los robots delta han emergido como una solución ideal en tareas de alta velocidad y precisión, especialmente en industrias como la alimentaria, farmacéutica y electrónica, donde el manejo de objetos ligeros es crucial. Diseñado inicialmente por el Dr. Reymond Clavel, el robot delta se distingue por su arquitectura paralela, la cual permite una manipulación rápida y precisa de objetos, minimizando la inercia debido a su diseño liviano y simplificado [1].

Los robots delta utilizan tres brazos articulados que convergen en una plataforma móvil, lo que les permite alcanzar velocidades y aceleraciones superiores a las de robots con arquitecturas seriales. Este diseño ofrece además una ventaja significativa en cuanto a rigidez y precisión, lo que facilita aplicaciones de manipulación en las que se requieren altos niveles de repetibilidad [2]. Estas características han impulsado el uso de los robots delta en el ámbito industrial y en la investigación, donde la tendencia actual es integrar sensores avanzados y sistemas de visión para mejorar la precisión en entornos complejos [3].

Con lo mencionado líneas arriba este proyecto plantea el desarrollo de un robot delta para esto de debe: Obtener la cinemática directa e inversa del robot delta, realizar el dimensionamiento de los motores, diseñar la estructura en SolidWorks y comprobar funcionamiento del robot.

II. GEOMETRIA DEL ROBOT

El sistema de coordenadas inercial del robot delta se encuentra en un punto O en la base donde están ubicados los tres motores. El eje Z de este sistema de coordenadas es perpendicular a la base y el eje X perpendicular al eje del motor número 1.

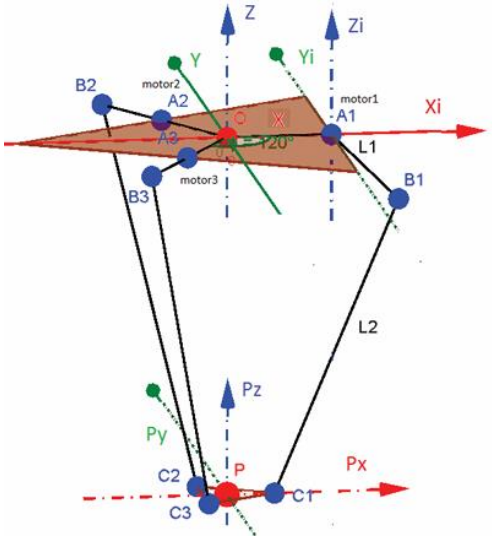

Figura 1. Representación geométrica del robot delta

En la Figura 1 se puede observar que cada motor tiene un brazo móvil conformado por dos eslabones, cuyas distancias son L1 y L2, las cuales son las mismas para cada brazo de cada motor. Cada motor se une al eslabón L1 por medio de una unión A_i, siendo un valor de i por cada motor (i=1, 2, 3). Los eslabones L1 y L2 se conectan mediante una unión B_i, y el eslabón L2 se une con una base móvil cuyo punto medio es el efector final P, esta unión de L2 con P se representa como C_i.

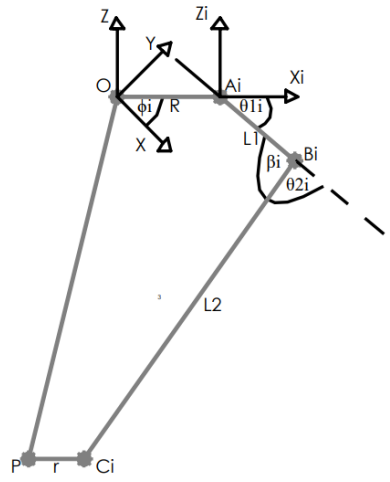

Figura 2. Sistemas de coordenadas y ángulos del robot delta

Cada brazo se encuentra sobre un sistema de coordenadas distinto al sistema de coordenadas inercial en el punto O, estos sistemas de coordenadas se encuentran a una distancia R con respecto al sistema de coordenadas inercial y con una rotación de un ángulo ϕ_i, siendo este 0°, 120° y 240° para los motores 1, 2 y 3 respectivamente, debido a que se debe tener una separación angular igual entre cada motor. Se tiene un ángulo θ_{1i} entre el eje X_i y el vector $\overline{AB_i}$, el cual es el ángulo que giran los motores, y otor ángulo θ_{1i} entre la proyección del vector $\overline{AB_i}$ y el eslabón L2. De igual manera, se tiene un ángulo interno β_{1i} entre los eslabones L1 y L2.

Para conocer los vectores generados entre el punto inicial O y cada punto de unión se utiliza el método de la matriz de transformación, el cual representa la orientación y posición de un sistema de coordenadas rotado y trasladado respecto a otro.

$$\overline{OA_i} = \begin{bmatrix} C\emptyset_i & -S\emptyset_i & 0 & RC\emptyset_i \\ S\emptyset_i & C\emptyset_i & 0 & RS\emptyset_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} RC\emptyset_i \\ RS\emptyset_i \\ 0 \\ 1 \end{bmatrix} \qquad Ec.1$$

$$\overline{OB_i} = \begin{bmatrix} C\emptyset_i & -S\emptyset_i & 0 & RC\emptyset_i \\ S\emptyset_i & C\emptyset_i & 0 & RS\emptyset_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L_1C\theta_{1i} \\ 0 \\ -L_1S\theta_{1i} \\ 1 \end{bmatrix} = \begin{bmatrix} C\emptyset_i(L_1C\theta_{1i} + R) \\ S\emptyset_i(L_1C\theta_{1i} + R) \\ -L_1S\theta_{1i} \\ 1 \end{bmatrix} \quad Ec.2$$

$$\overline{OC_i} = \begin{bmatrix} C\emptyset_i & -S\emptyset_i & 0 & RC\emptyset_i \\ S\emptyset_i & C\emptyset_i & 0 & RS\emptyset_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_xC\emptyset_i + P_yS\emptyset_i + r - R \\ P_yC\emptyset_i - P_xS\emptyset_i \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x + rC\emptyset_i \\ P_y + rS\emptyset_i \\ P_z \\ 1 \end{bmatrix} Ec.3$$

Con las ecuaciones 1, 2 y 3 se obtienen los vectores $\overline{BA_i}$, $\overline{BC_i}$ y $\overline{AC_i}$.

$$\overline{BA_i} = \overline{OA_i} - \overline{OB_i} = \begin{bmatrix} -C\emptyset_iL_1C\theta_{1i} \\ -C\emptyset_iL_1C\theta_{1i} \\ L_1S\theta_{1i} \end{bmatrix} \qquad Ec.4$$

$$\overline{BC_i} = \overline{OC_i} - \overline{OB_i} = \begin{bmatrix} P_x - C\emptyset_i(R + L_1C\theta_{1i} - r) \\ P_y - S\emptyset_i(R + L_1C\theta_{1i} - r) \\ P_z + L_1S\theta_{1i} \end{bmatrix} \qquad Ec.5$$

$$\overline{AC_i} = \overline{OC_i} - \overline{OA_i} = \begin{bmatrix} P_x - C\emptyset_i(r - R) \\ P_y - S\emptyset_i(r - R) \\ P_z \end{bmatrix} \qquad Ec.6$$

III. CINEMATICA INVERSA

La cinemática inversa permite encontrar los ángulos de entrada θ_{1i} de cada motor a partir de conocer la posición P del efector final. Para esto se puede observar al robot de la siguiente manera:

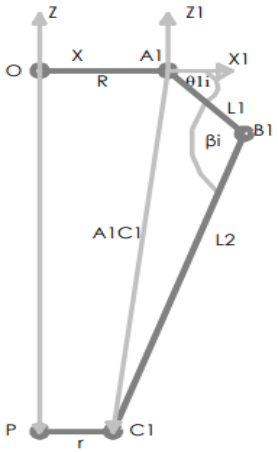

Figura 3. Representación trigonométrica del robot delta

En la Figura 3 se puede observar que se forma un triángulo entres los puntos A_i, B_i y C_i y se aplica teorema de cosenos para hallar la longitud de $\overline{AC_i}$.

$$AC_1^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(\beta_i) \qquad Ec.7$$

Se calcula el coseno del ángulo utilizando la ecuación del producto escalar:

$$\overline{BA_i} * \overline{BC_i} = |BA|_i|BC|_i\cos(\beta_i) \qquad Ec.8$$

Dado que L1 es la unión de los puntos A y B, y de igual manera, L2 es la unión de los puntos B y C, se tiene que:

$$L_1 = |BA|_i \qquad Ec.9$$

$$L_2 = |BC|_i \qquad Ec.10$$

Se reemplazan las ecuaciones 9 y 10 en la ecuación 8 y se obtiene:

$$\cos(\beta_i) = \frac{\overline{BA_i} * \overline{BC_i}}{L_1 L_2} \qquad Ec.11$$

Se reemplaza la ecuación 11 en la ecuación 7:

$$AC_1^2 = L_1^2 + L_2^2 - 2(\overline{BA_i} * \overline{BC_i}) \qquad Ec.12$$

Reemplazando los valores de las ecuaciones 5 y 4 en la ecuación 12:

$$AC_1^2 = L_1^2 - L_2^2 + 2L_1C\theta_{1i}\left(P_xC\emptyset_i + P_yS\emptyset_i + r - R\right) - 2L_1S\theta_{1i}P_z \quad Ec.13$$

Se tiene que la magnitud del vector $\overline{AC_i}$ según la ecuación 6 es:

$$|AC|_1 = AC_1^2 = \left(P_x + C\emptyset_i(r-R)\right)^2 + \left(P_y + S\emptyset_i(r-R)\right)^2 + P_z \ldots Ec.14$$

Igualando las ecuaciones 13 y 14 se obtiene:

$$a_iS\theta_{1i} + b_iC\theta_{1i} = c_i \qquad Ec.15$$

Donde:

$$a_i = -2P_zL_1 \qquad Ec.16$$

$$b_i = 2L_1(P_xC\emptyset_i + P_yS\emptyset_i + r - R) \qquad Ec.17$$

$$c_i = \left(P_x + C\emptyset_i(r-R)\right)^2 + \left(P_y + S\emptyset_i(r-R)\right)^2 + P_z^2 + L_1^2L_2^2 \qquad Ec.18$$

Utilizando identidades trigonométricas en la ecuación 15 se obtiene:

$$a_iS\theta_{1i} + b_iC\theta_{1i} = \sqrt{a_i^2 + b_i^2} * \sin\left(\theta_{1i} + tan^{-1}\left(\frac{b_i}{a_i}\right)\right) \qquad Ec.19$$

Se igualan las ecuaciones 15 y 19, y se procede a despejar el ángulo de rotación de los motores:

$$\theta_{1i} = \sin^{-1}\left(\frac{c_i}{\sqrt{a_i^2 + b_i^2}}\right) - \tan^{-1}\left(\frac{b_i}{a_i}\right) \qquad Ec.20$$

IV. CINEMATICA DIRECTA

La cinemática directa permite encontrar la posición P del efector final a partir del conocimiento de los ángulos de la posición de los motores.

A partir de la ecuación 5, que representa el punto de conexión del brazo con el efector final, se tiene que su magnitud es igual a L2, por lo cual se tiene que:

$$|BC|_i = L_2^2 = \left(P_x - C\emptyset_i(R + L_1C\theta_{1i} - r)\right)^2 + \left(P_y - S\emptyset_i(R + L_1C\theta_{1i} - r)\right)^2 + (P_z + L_1 + S\theta_{1i})^2 \qquad Ec.21$$

A partir de la ecuación 21 se realiza un sistema de ecuaciones reemplazando las 3 posiciones angulares de ϕ_i por cada motor, siendo estas 0°, 120° y 240° para los motores 1, 2 y 3 respectivamente.

$$\begin{cases} (P_x - R - L_1cC\theta_{11} + r)^2 + P_y^2 + (Pz_z + L1_1S\theta_{11})^2 = L_2^2 \\ \left(P_x + \frac{1}{2}(R + L_1C\theta_{12} - r)\right)^2 + \left(P_y - \frac{\sqrt{3}}{2}(R + L_1C\theta_{12} - r)\right)^2 + (P_z + L_1S\theta_{12})^2 = L_2^2 \\ \left(P_x + \frac{1}{2}(R + L_1C\theta_{13} - r)\right)^2 + \left(P_Y + \frac{\sqrt{3}}{2}(R + L_1C\theta_{13} - r)\right)^2 + (Pz + L_1S\theta_{13})^2 = L_2^2 \end{cases} \qquad Ec.22$$

V. ESPACIO DE TRABAJO

El espacio de trabajo de un robot define la región tridimensional donde su efector final puede operar, teniendo en cuenta las limitaciones mecánicas, cinemáticas y estructurales. Para un robot delta, este espacio está influenciado por la geometría de los brazos, las restricciones de los actuadores y el rango angular de las articulaciones. Además, dentro de este espacio general, se encuentra el espacio de trabajo efectivo, que representa la región donde el robot puede operar de manera óptima y segura. Evaluar estas regiones es crucial para determinar la capacidad y alcance del robot en tareas específicas.

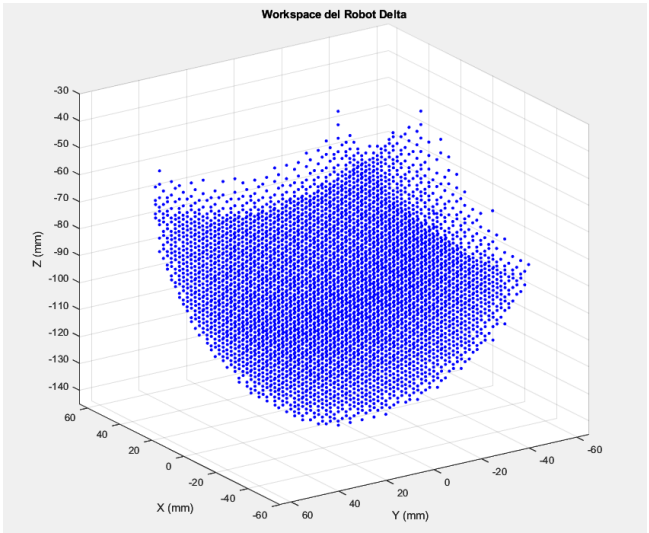

Figura 4. Espacio de trabajo robot delta.

Esta figura muestra el espacio de trabajo total del robot delta, representado como un conjunto de puntos en un sistema tridimensional. Cada punto corresponde a una posición alcanzable por el efector final del robot, considerando todas las configuraciones posibles. La forma generada refleja las restricciones geométricas y mecánicas del sistema, destacando las áreas donde el robot tiene mayor accesibilidad.

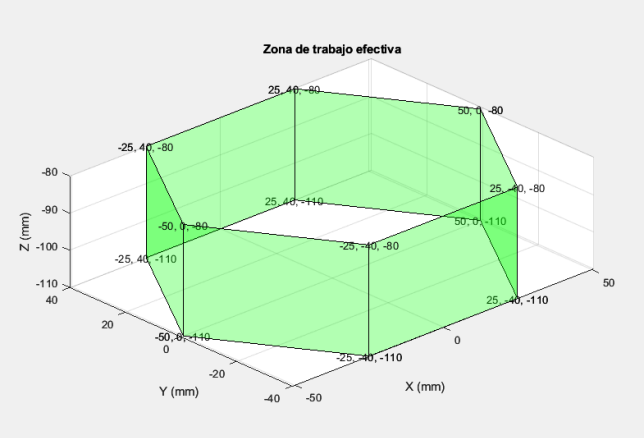

Figura 5. Espacio de trabajo efectivo robot delta

Aquí se representa el espacio de trabajo efectivo, una subregión dentro del espacio total donde el robot puede operar de forma segura y eficiente. Esta región está delimitada por factores como límites de velocidad, carga y precisión requerida. La figura utiliza una superficie verde para identificar claramente esta zona, resaltando las posiciones útiles y prácticas del efector final.

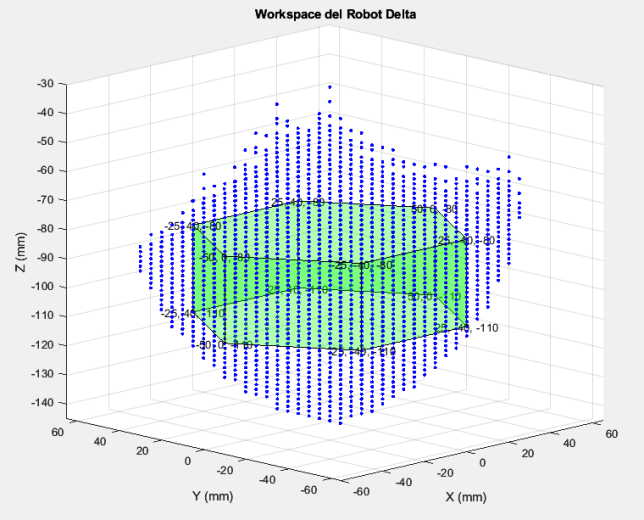

Figura 6. Espacio de trabajo efectivo en el espacio de trabajo total.

En esta figura se superpone el espacio de trabajo efectivo (en verde) dentro del espacio de trabajo completo (en azul). Esta representación combinada permite visualizar la relación entre ambas regiones, mostrando cómo el espacio efectivo ocupa solo una porción del espacio total. Esto es útil para identificar posibles áreas de operación óptima y para ajustar las tareas del robot a sus capacidades reales.

VI. REQUERIMIENTOS

Se presenta una tabla que muestra los requerimientos que debe cumplir el robot delta que se ha diseñado. Estos requerimientos se obtuvieron mediante el análisis de las especificaciones técnicas de los motores paso a paso utilizados, del análisis estructural y del cálculo del espacio de trabajo.

Tabla 1. Requerimientos

Requerimientos	Valores
Velocidad máxima	300 RPM
Rango de movimiento motores	0 a 90°
Carga máxima	750g

VII. DISEÑO DE LA ESTRUCTURA

El proceso de diseño de la estructura del robot delta se desarrolló de manera iterativa, comenzando con la base superior, la cual se definió con una geometría triangular y lados redondeados, asegurando un soporte adecuado para los motores y los brazos. Posteriormente, se diseñó la base intermedia, con cortes rectangulares en los vértices para permitir el encaje con las patas y garantizar estabilidad. Las patas fueron diseñadas en MDF de 3 mm de grosor, con una longitud de 200 mm y segillas de 10 mm en los extremos, facilitando un ensamblaje firme con la base intermedia. Finalmente, se integraron componentes como el efector final, los brazos y los eslabones, fabricados en PLA mediante impresión 3D, asegurando precisión en las articulaciones y funcionalidad en el movimiento del robot.

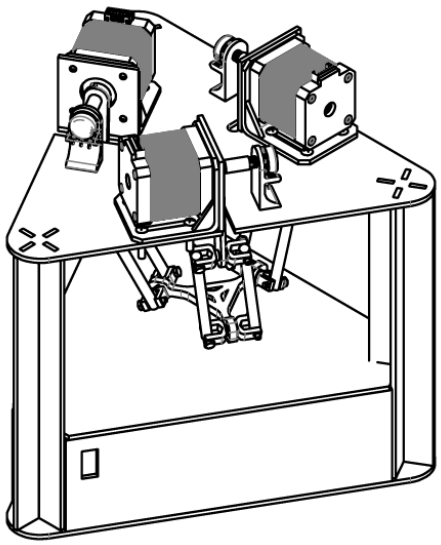

Figura 7. Robot delta

El criterio principal para el diseño fue la optimización de materiales y la facilidad de manufactura, utilizando MDF y PLA debido a su disponibilidad y facilidad de corte o impresión. Además, se priorizó la estabilidad estructural y el ensamblaje eficiente, garantizando que todas las piezas encajaran de forma precisa para soportar los esfuerzos durante la operación. Las dimensiones y detalles específicos, como las perforaciones para motores y las ranuras de encaje, se ajustaron para cumplir con los requisitos de funcionalidad y precisión del robot delta.

Los criterios de selección y diseño de piezas se han basado en tres factores principales: disponibilidad, costos y tiempos. La disponibilidad ha sido el criterio más importante, ya que representa la capacidad de acceder a cada componente de manera oportuna para evitar retrasos en el ensamblaje. Los costos también se han considerado cuidadosamente para mantener el proyecto dentro del presupuesto asignado, sin comprometer la funcionalidad y resistencia del robot. Además, los tiempos de armado y fabricación han sido cruciales debido a la restricción temporal del proyecto, buscando métodos de fabricación rápidos para la mayoría de las piezas y reservando métodos más complejos para los componentes que lo requieran específicamente. Estos criterios han sido clave para lograr un diseño efectivo y funcional, adaptado tanto al presupuesto como a la restricción temporal del proyecto.

VIII. SIMULACION DE ESFUERZOS

La simulación de esfuerzos es una herramienta fundamental en el diseño de sistemas mecánicos, ya que permite evaluar cómo se comportan los componentes bajo diversas condiciones de carga. A través de este análisis, es posible identificar zonas críticas de tensión, desplazamiento y deformación, garantizando así la fiabilidad y seguridad del diseño. En este caso, se realizó un análisis estático del robot delta, enfocándose en los brazos y los componentes principales, para evaluar su comportamiento estructural.

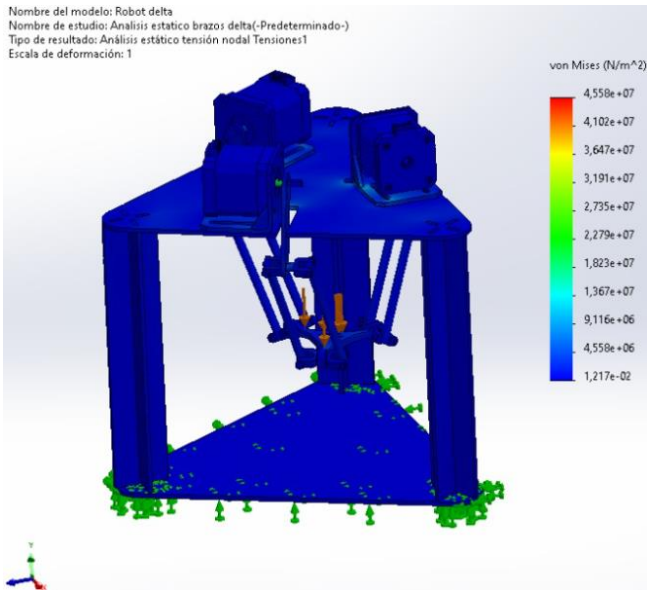

Figura 8. Simulación tensión nodal.

Esta simulación muestra la distribución de tensiones von Mises en el robot delta bajo una carga específica. Los colores representan los niveles de tensión, desde azul (baja tensión) hasta rojo (alta tensión). Las zonas más críticas se identifican en las áreas rojas, donde el material podría estar más cerca de su límite elástico. Esto permite evaluar la resistencia de los materiales en las partes más exigidas.

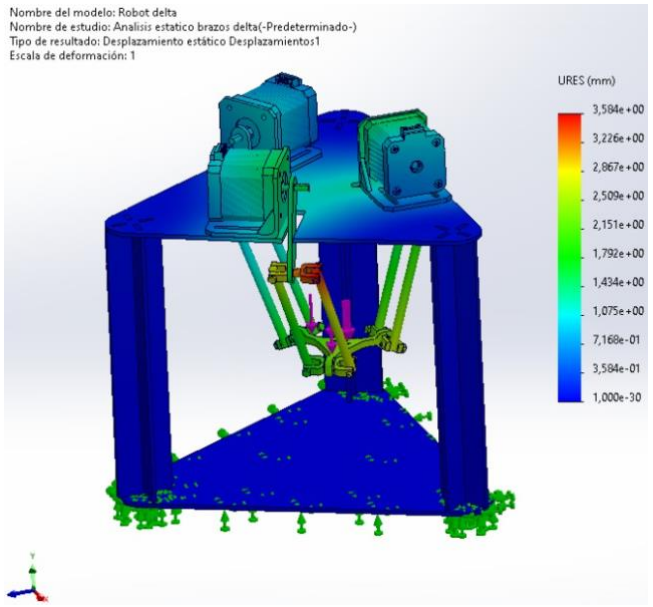

Figura 9. Simulación desplazamiento estático.

En este análisis, se presenta el desplazamiento total en cada nodo del robot delta. Los colores van desde azul (mínimo desplazamiento) hasta rojo (máximo desplazamiento). Este resultado es clave para entender cómo las fuerzas aplicadas afectan la estabilidad y precisión del sistema, ya que desplazamientos excesivos pueden comprometer el rendimiento del robot.

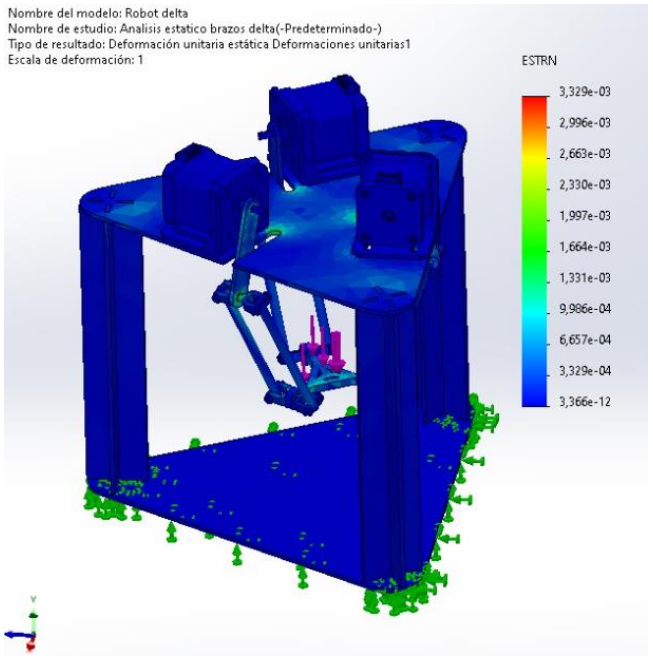

Figura 10. Simulación deformación unitaria estática.

La simulación muestra la deformación unitaria de los componentes del robot delta. Los colores indican la magnitud de la deformación, desde azul (mínima deformación) hasta rojo (máxima deformación). Este análisis permite verificar si las deformaciones son aceptables dentro de los límites de diseño, garantizando que no se produzcan fallos por deformación excesiva.

IX. SELECCIÓN DE MATERIALES Y COMPONENTES

A. Motor

Se ha seleccionado el motor NEMA 16, un motor paso a paso, debido a su capacidad para controlar con precisión los ángulos de los brazos del robot, lo cual es crucial para el movimiento del efector final. Este motor cuenta con un torque máximo de 1.2 kg·cm, lo que se ha evaluado como suficiente para soportar la carga requerida de 2 kg, considerando la distribución del esfuerzo entre los tres motores y la geometría del sistema. Con una longitud de eslabón de 6 cm, la fuerza máxima que cada motor puede generar se ha calculado como $F = \frac{T}{d} = \frac{1.2kg \cdot cm}{6cm}$, resultando en una capacidad de carga de aproximadamente 0.2 kg por motor a esa distancia. Al estar distribuidos los 0.5 kg entre los tres motores, cada uno soporta una carga de aproximadamente 0.2 kg, lo que se encuentra dentro de su capacidad operativa.

Esta selección también se ha fundamentado en la disponibilidad del motor como recurso proporcionado por la universidad, lo que ha permitido su integración sin costos adicionales ni retrasos en la adquisición. Además, la naturaleza precisa y controlable de los motores NEMA 16 ha sido ideal para garantizar la estabilidad y el desempeño del efector final, validando su inclusión como componente clave en el diseño del robot delta.

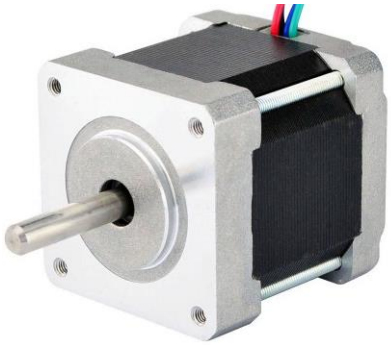

Figura 11. Nema 17.

B. Base del motor

Se ha decidido utilizar una base en forma de "L" para el motor NEMA 17, con el propósito de reducir los tiempos de fabricación, evitando el proceso de impresión de una base personalizada. La forma en "L" ha permitido fijar el motor de manera estable sobre la cara superior de la base estructural, asegurando una alineación adecuada en el montaje. Aunque la base específica para el NEMA 16 no se encontraba disponible, se optó por la del NEMA 17 debido a que, a pesar de ser ligeramente más grande, comparte el mismo tamaño y posición de tornillos, permitiendo su compatibilidad sin afectar la estabilidad o funcionalidad del sistema.

Figura 12. Base del motor Nema 17.

X. DISEÑO DE ESTRUCTURA

A. Base superior

El diseño de la base superior se ha realizado en una estructura triangular con lados redondeados, con una distancia de 85 mm desde el centro hasta cada arista y bordes con un radio de 20 mm. Cada cara cuenta con un corte central de 4 mm de grosor y 50 mm de longitud, terminando en un redondeado de 4 mm de radio. En cada vértice se han integrado ranuras rectangulares de 3 mm por 10 mm para fijar las patas, organizadas en forma de cruz con una separación de 10 mm entre las ranuras colineales y orientadas hacia el centro. La pieza, con un grosor uniforme de 3 mm, ha sido cortada a partir de una lámina de madera mediante una cortadora láser, logrando bordes precisos y uniformes. Este diseño ha proporcionado una base estable y alineada, con perforaciones para los motores y aperturas que garantizan el movimiento sin obstrucciones de los brazos del robot, asegurando la funcionalidad y precisión del sistema.

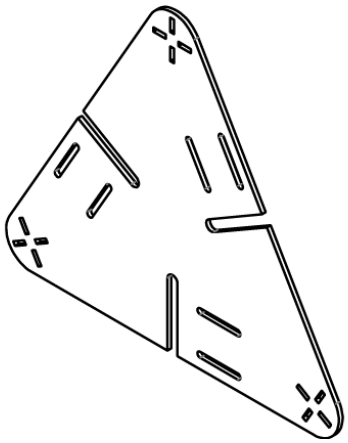

Figura 13. Base superior

B. Base intermedia

El diseño de la base intermedia se ha realizado en una estructura triangular con esquinas adaptadas para incluir cortes

rectangulares perpendiculares de 4 mm por 10 mm. Estos cortes están diseñados para encajar perfectamente con las patas del robot, proporcionando soporte estructural adicional. A diferencia de la base superior, esta pieza no incluye esquinas redondeadas ni perforaciones para motores, enfocándose en su función como piso para el robot, colocándose sobre la parte superior de la circuitería. La base intermedia tiene una distancia de 85 mm desde el centro hasta cada arista y un grosor uniforme de 3 mm. Ha sido cortada a partir de una lámina de madera utilizando una cortadora láser, logrando una estructura precisa y estable que asegura el soporte

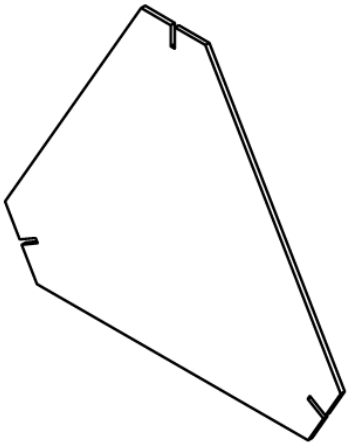

Figura 14. Base intermedia

C. Base inferior

El diseño de la base inferior se ha realizado en una estructura triangular con lados redondeados, con una distancia de 85 mm desde el centro hasta cada arista y bordes con un radio de 20 mm. A diferencia de la base superior, esta pieza no cuenta con cortes en las aristas ni perforaciones específicas para la fijación de motores. Sin embargo, conserva la forma triangular y un grosor uniforme de 3 mm, lo que asegura estabilidad y rigidez en la estructura general del robot. La base intermedia ha sido cortada a partir de una lámina de madera utilizando una cortadora láser, logrando bordes precisos y uniformes. Este diseño ha proporcionado soporte adicional a la estructura, mejorando su alineación y robustez para garantizar un funcionamiento óptimo del robot

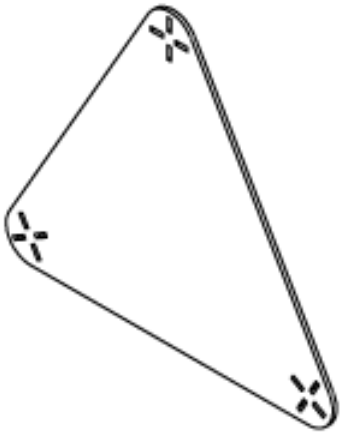

Figura 14. Base inferior

C. Brazo

El brazo se ha construido en MDF de 3 mm de grosor y presenta una estructura también triangular, manteniendo una distribución uniforme de la carga en el movimiento. En el extremo destinado al motor, se ha diseñado una apertura especial que encaja con el eje, permitiendo la transmisión directa del movimiento. En el otro extremo, se ha integrado una apertura circular para el eje de la junta universal, asegurando un rango de movimiento adecuado. La distancia entre ambas aperturas es de 90 mm, y el ancho del brazo se ha definido en 15 mm, logrando así una estructura ligera y estable. Este diseño ha sido fundamental para asegurar que el brazo funcione en conjunto con el sistema, manteniendo precisión y estabilidad en cada movimiento.

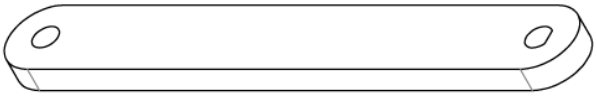

Figura 15. Brazo

D. Paralelogramo

El diseño del paralelogramo se ha basado en el uso de pares de varillas de hierro liso de 5 mm de diámetro y 100 mm de longitud, seleccionadas por su rigidez y capacidad de mantener la estabilidad estructural del robot. Para proporcionar la movilidad necesaria en las articulaciones, se han realizado perforaciones tangenciales de 3 mm de diámetro a 3 mm de cada extremo de las varillas. Estas perforaciones permiten la inserción de ejes, garantizando que las varillas se conecten de forma articulada y mantengan una orientación constante del efector final durante el movimiento. Este diseño ha sido esencial para asegurar la precisión y fiabilidad del robot en tareas que requieren un control exacto de la posición y orientación.

Figura 16. Varilla

E. Junta universal

El diseño de la junta universal se ha compuesto por dos medias juntas impresas en 3D, las cuales presentan una estructura simétrica. Cada media junta incluye un eje circular con un corte cuadrado tangencial en su base, lo que permite que ambas piezas puedan unirse de manera precisa al voltear una de ellas y pegarla a la otra utilizando cianoacrilato en la zona de corte. Después del eje circular, la pieza se extiende en una forma de “U” con ángulos rectos y una perforación circular de 3 mm en cada lado. Esta perforación permite la inserción de un eje que conecta la junta universal a la varilla del robot. La estructura del eje central proporciona rotación en el eje y,

mientras que la perforación y su eje permiten rotación en el eje x, otorgando así un rango de movimiento multidireccional al sistema. Este diseño ha sido esencial para proporcionar flexibilidad y precisión en los movimientos del efector final.

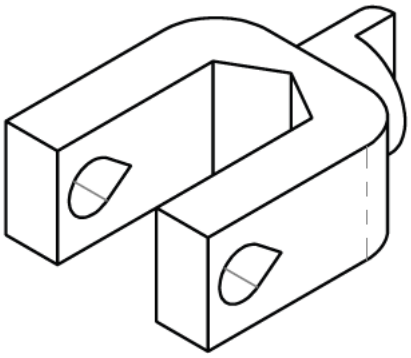

Figura 17. Medio eje universal.

F. Efector final

El efector final tiene una forma en "Y", con ángulos de 120 grados entre cada uno de sus brazos, lo que permite una distribución simétrica ideal para controlar de manera precisa el movimiento. Cada brazo tiene una longitud de 35 mm medida desde el centro hasta el centro de los orificios, un ancho de 5 mm y un grosor de 5 mm, proporcionando una estructura ligera pero robusta. En los extremos de cada brazo se encuentran perforaciones diseñadas específicamente para conectar juntas universales, asegurando el rango de movimiento necesario para adaptarse a las orientaciones y posiciones requeridas por el robot. Este componente ha sido fabricado en material PLA mediante impresión 3D, lo que ha permitido obtener un diseño preciso y funcional. Su estructura ha sido clave para garantizar la estabilidad y precisión del efector final en las tareas asignadas.

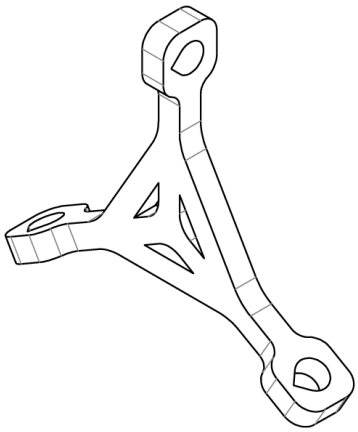

Figura 18. Efector final

G. Soporte

Los soportes del robot han sido fabricados en MDF con un grosor de 3 mm, proporcionando rigidez estructural y estabilidad al diseño. Cada pata tiene una longitud de 200 mm, excluyendo las rejillas que se encuentran en cada extremo. Estas rejillas, diseñadas para facilitar el encaje y soporte en la estructura, tienen un largo de 10 mm cada una y están separadas entre sí por una distancia de 10 mm. La pata presenta un ancho total de 40 mm, lo que asegura una base sólida y resistente para soportar el peso y las dinámicas del robot durante su operación. Este diseño ha sido clave para garantizar una conexión estable con el resto de la estructura y para mantener la alineación adecuada del sistema.

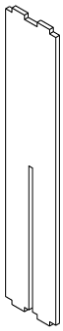

Figura 19. Soporte

H. Seguidor del eslabón

El seguidor es una pieza fabricada en PLA mediante impresión 3D, diseñada para ajustar y fijar el eslabón en el sistema del robot delta. Tiene una forma semicircular con un diámetro externo de 8.10 mm y un radio interno de 2 mm, con una altura total de 10.93 mm y un grosor de base de 2 mm, y un grosor de la lengüeta de 3mm equivalente al grosor del eslabón. En su parte superior, cuenta con un corte plano de 4.10 mm en la sección interna para asegurar un encaje preciso con el eslabón, mientras que su orificio central, con un diámetro de 3.07 mm, está diseñado para alojar un eje que permite un movimiento articulado. Además, la base del seguidor incluye una extensión de 4.30 mm que facilita su integración con otros componentes del robot. Este diseño compacto y funcional garantiza la estabilidad y precisión del eslabón, contribuyendo al correcto desempeño del sistema en tareas asignadas.

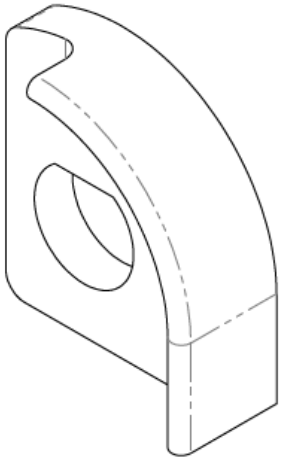

Figura 20. Seguidor de eslabón

I. Base de potenciómetro

La base del potenciómetro, fabricada en PLA mediante impresión 3D, tiene una altura total de 38.5 mm y un diseño optimizado para sostener el potenciómetro de manera precisa. Cuenta con un orificio central de 6.76 mm de diámetro ubicado a 29 mm desde la base, mientras que su parte superior incluye un redondeo con un radio de 9.5 mm para facilitar el montaje. En la parte trasera, posee dos brazos laterales de 2 mm de grosor y 9 mm de longitud, situados a una altura de 17.12 mm desde la base principal, proporcionando soporte adicional. La base inferior, con una longitud de 13 mm, garantiza la estabilidad estructural y permite una fijación firme al sistema. Este diseño asegura que el potenciómetro se mantenga alineado y funcional dentro del conjunto del robot delta.

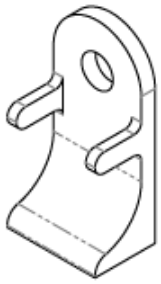

Figura 21. Base del potenciómetro

X. PRUEBAS EXPERIMENTALES

En esta sección se describen las pruebas experimentales realizadas para evaluar el desempeño del sistema de control del robot, comparando los valores teóricos de los ángulos articulares (q) obtenidos a partir de modelos matemáticos, con los valores medidos por los sensores (potenciómetros).

Se realizó un análisis experimental para encontrar la resolución de los potenciómetros en la cual se utilizó una placa ESP32 para su alimentación y lectura, la cual tiene 12 bits de resolución, es decir, que realiza lecturas analógicas de 0 a 3,3V representándolas en un rango de 0 a 4095. En dicho análisis se obtuvieron los siguientes rangos:

Potenciómetro 1 (s1): 1790 a 3200, donde 1790 es igual a una posición angular de 90° del motor 1 y 3200 es igual a 0°.

Potenciómetro 2 (s2): 1810 a 3160, donde 1810 es igual a una posición angular de 90° del motor 2 y 3160 es igual a 0°.

Potenciómetro 3 (s3): 2000 a 3160, donde 2000 es igual a una posición angular de 90° del motor 3 y 3160 es igual a 0°.

La diferencia entre los valores teóricos y los medidos permite calcular el error para cada articulación, lo que se presenta en gráficos de barras. Por otro lado, los gráficos de líneas muestran una comparación directa entre los ángulos teóricos (q) y los valores reales obtenidos de los potenciómetros.

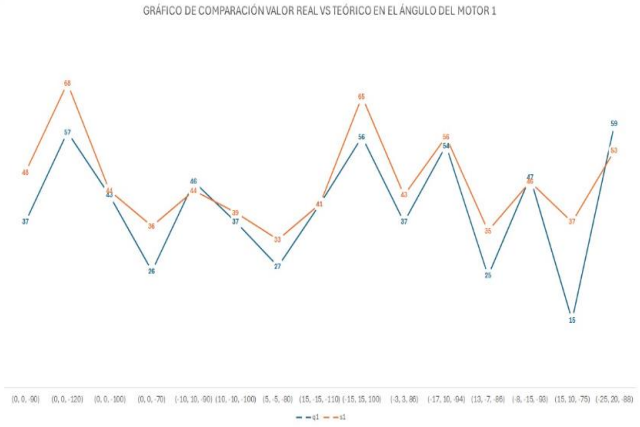

Figura 22. Gráfico comparativo entre q1 y s1.

La Figura 22 compara el ángulo teórico q1 (Línea azul) con el valor medido s1 (Línea naranja) del potenciómetro correspondiente al motor 1. Las líneas muestran cómo se comportan los valores teóricos y medidos a lo largo del tiempo o del rango de movimiento.

Figura 23. Gráfico del error del motor 1.

La Figura 23 presenta el error absoluto del motor 1. El error se calcula como la diferencia entre el valor teórico q1 y el valor medido por el potenciómetro s1. Las barras muestran la magnitud del error en distintas posiciones del motor.

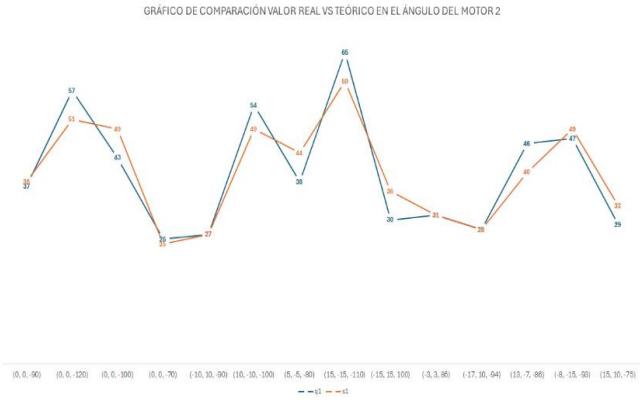

Figura 24. Gráfico comparativo entre q2 y s2.

La Figura 24 representa la comparación entre el ángulo teórico q2 (Línea azul) y el valor obtenido de s2 (Línea naranja). Las líneas permiten observar posibles discrepancias entre ambos valores.

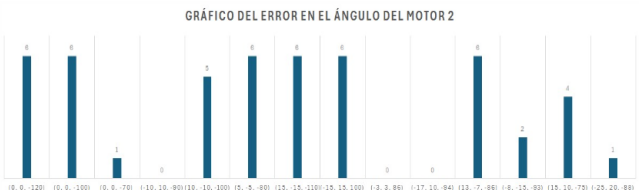

Figura 25. Gráfico de error del motor 2.

La Figura 25 muestra el error absoluto para el motor 2. Se representa la diferencia entre q2 (teórico) y s2 (medido por el potenciómetro). Se puede observar que se tiene una repetición constante de un error de 6°.

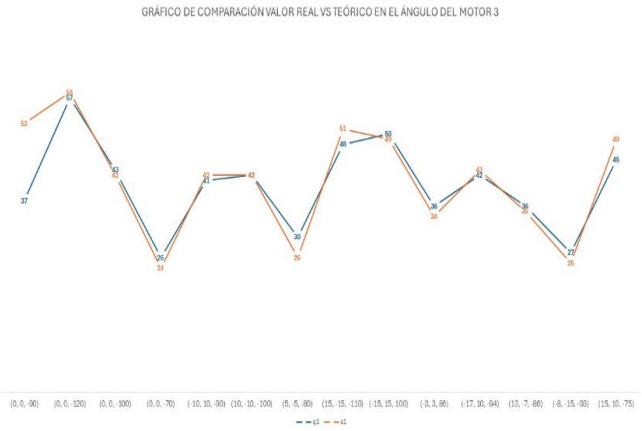

Figura 26. Gráfico comparativo entre q3 y s3.

La Figura 26 muestra la comparación entre el ángulo teórico q1 y el valor medido s3 del potenciómetro del motor 3. La similitud entre las líneas refleja la precisión del sistema de medición y control a diferencia a los gráficos de q1 y q2.

Figura 27. Gráfico del error del motor 3.

En la Figura 27 se visualiza el error absoluto para el motor 3. La diferencia entre q3 y s3 se presenta en forma de barras, facilitando la identificación de desviaciones en las mediciones en donde se puede observar que la primera barra es un valor mucho mayor comparado al del resto que no tiene mucha diferencia.

XI. CONCLUSIONES

En este proyecto se realizó el diseño y construcción de un robot delta para lo cual se realizaron cálculos de cinemática inversa, cinemática directa, espacio de trabajo, espacio de trabajo efectivo, diseño de la estructura en SolidWorks, con el cual se realizó la simulación de la tensión nodal, desplazamiento estático y la deformación unitaria estática sobre el robot. Para la construcción de robot se utilizaron principalmente dos materiales: PLA (ácido poliláctico) y MDF (Fibras de densidad media), la estructura y los primeros eslabones de los brazos se realizaron de mdf por otro lado los ejes, las varillas y el efector final están hechos de PLA.

Finalmente, se ejecutaron pruebas experimentales para verificar el desempeño del sistema. Se compararon los valores teóricos de los ángulos de los motores obtenidos mediante cálculos de cinemática inversa con las mediciones proporcionadas por los sensores (potenciómetros) instalados en el robot. Con estos valores se realizaron gráficos con los cuales se logró visualizar el error que este robot presentó.

XII. BIBLIOGRAFÍA

[1] R. Clavel, *Conception d'un robot parallèle rapide à 4 degrés de liberté*, Lausanne, Switzerland: École Polytechnique Fédérale de Lausanne, 1991.

[2] C. Gosselin and J. Angeles, "Singularity Analysis of Closed-Loop Kinematic Chains," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281-290, Jun. 1990.

[3] M. Honegger, H. Yoshihito, and I. Codourey, "Sensor Fusion for High-Performance Robots: The Delta Example," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Nagoya, Japan, 1995, pp. 1717-1722.

[4] J. Gudiño-Lau, J. Alcalá-Rodríguez, H. Narrarro, D. Velez-Díaz, y S. Charre-Ibarra, "Diseño y modelo cinemático de un robot delta para el diagnóstico y rehabilitación," *Boletín Científico Tlahuelilpan*, vol. 8, pp. 1-10, 2017

[5] W. C. Pachacama Oña y J. F. Salazar Muzo, *Diseño y construcción de un prototipo de robot delta para aplicaciones pick & place*, tesis de grado, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2016

XII. ANEXOS

A. Código para la interfaz de control del robot delta

```
#include <WiFi.h>
#include <WiFiMulti.h>

#define paso1 15
#define dire1 2
#define paso2 13
#define dire2 12
#define paso3 25
#define dire3 26
#define sensor_1 33
#define sensor_2 35
#define sensor_3 32

int s1, s2, s3;
int grado1;
int grado2;
int grado3;
int pasos1;
int pasos2;
int pasos3;

int R = 60;
int r = 35;
int L1 = 60;
int L2 = 90;
int retardo = 2000;

WiFiMulti wifiMulti;
WiFiServer servidor(80);

String modo = "cordenadas";
float X = 0.0, Y = 0.0, Z = -90.0;
float q1 = 0.0, q2 = 0.0, q3 = 0.0;

bool bandera = false;

void calculo_de_cordenadas();
void calculo_de_q();
void pasopaso();
void control();
void calculo_de_s();

void setup() {
    pinMode(paso1, OUTPUT);
    pinMode(dire1, OUTPUT);
    pinMode(paso2, OUTPUT);
    pinMode(dire2, OUTPUT);
    pinMode(paso3, OUTPUT);
    pinMode(dire3, OUTPUT);

    Serial.begin(115200);
    Serial.println("\nIniciando WiFi Multi");

    //wifiMulti.addAP("mazzocato", "leon6734");
    wifiMulti.addAP("MATIAS", "Emily123");

    WiFi.mode(WIFI_STA);
    Serial.print("Conectando a WiFi...");
    while (wifiMulti.run() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
    }
    if (WiFi.localIP() == INADDR_NONE || WiFi.localIP().toString() == "0.0.0.0") {
        Serial.println("\nError: No se obtuvo una IP válida. Reiniciando...");
        ESP.restart(); // Reinicia el microcontrolador
    }
    Serial.println("\nConectado a WiFi");
    Serial.print("IP local: ");
    Serial.println(WiFi.localIP());
    servidor.begin();
    Serial.println("Servidor iniciado correctamente");
}

void loop() {
  WiFiClient cliente = servidor.available();
  if (cliente) {
    Serial.println("Nuevo cliente conectado");
    String peticion = cliente.readStringUntil('\r');
    cliente.flush();
    int q1Index = peticion.indexOf("q1=");
    int q2Index = peticion.indexOf("q2=");
    int q3Index = peticion.indexOf("q3=");
    int xIndex = peticion.indexOf("X=");
    int yIndex = peticion.indexOf("Y=");
    int zIndex = peticion.indexOf("Z=");
    if (q1Index!=q1 && q2Index!=q2 && q3Index!=q3 && xIndex!=X && yIndex!=Y && zIndex!=Z){
      if (peticion.indexOf("GET /modo/cordenadas") >= 0) {
        modo = "cordenadas";
      }else if (peticion.startsWith("GET /actualizar?")) {
        s1 = analogRead(sensor_1);
        s2 = analogRead(sensor_2);
        s3 = analogRead(sensor_3);

        if (q1Index >= 0) q1 = peticion.substring(q1Index + 3, peticion.indexOf('&', q1Index)).toFloat();
        if (q2Index >= 0) q2 = peticion.substring(q2Index + 3, peticion.indexOf('&', q2Index)).toFloat();
        if (q3Index >= 0) q3 = peticion.substring(q3Index + 3, peticion.indexOf('&', q3Index)).toFloat();
        if (xIndex >= 0) X = peticion.substring(xIndex + 2, peticion.indexOf('&', xIndex)).toFloat();
        if (yIndex >= 0) Y = peticion.substring(yIndex + 2, peticion.indexOf('&', yIndex)).toFloat();
        if (zIndex >= 0) Z = peticion.substring(zIndex + 2, peticion.indexOf('&', zIndex)).toFloat();

        bandera = false;
      }
      enviarPaginaWeb(cliente);
      if (modo == "cordenadas") {
        if (bandera == false) {
          calculo_de_s(); // Calcula los valores actuales de los sensores
          control();
          bandera = true;
        }
      }
    }
    delay(6500);
    cliente.stop();
    Serial.println("Cliente desconectado");
  }
}

void enviarPaginaWeb(WiFiClient &cliente) {
  cliente.println("HTTP/1.1 200 OK");
  cliente.println("Content-Type: text/html; charset=UTF-8");
  cliente.println("Connection: close");
  cliente.println();

  cliente.println("<!DOCTYPE HTML>");
  cliente.println("<html>");
```

```
  cliente.println("<head>");
  cliente.println("<style>");
  cliente.println("button  {  background-color:  #4CAF50;
color: white; padding: 10px; margin: 5px; }");
  cliente.println("input { width: 50px; }");
  cliente.println("</style>");
  cliente.println("</head>");
  cliente.println("<body>");

  if (modo == "cordenadas") {
    cliente.println("<h2>Modo: Coordenadas</h2>");
    cliente.println("<form>");
    cliente.println("<p>X=<input    type='text'    id='X'
value='" + String(X, 2) + "'></p>");
    cliente.println("<p>Y=<input    type='text'    id='Y'
value='" + String(Y, 2) + "'></p>");
    cliente.println("<p>Z=<input    type='text'    id='Z'
value='" + String(Z, 2) + "'></p>");
    cliente.println("<p>q1=" + String(q1, 2) + "</p>");
    cliente.println("<p>q2=" + String(q2, 2) + "</p>");
    cliente.println("<p>q3=" + String(q3, 2) + "</p>");
    cliente.println("</form>");
  } else {
    cliente.println("<h2>Error: Modo no soportado</h2>");
  }
  cliente.println("<button
onclick=\"actualizar()\">Actualizar</button>");
  cliente.println("<script>");
  cliente.println("function actualizar() {");
  cliente.println("  let params = '';");

  cliente.println("                                        if
(!document.getElementById('X').value                    ||
!document.getElementById('Y').value                     ||
!document.getElementById('Z').value) {");
  cliente.println("    alert('Por favor, ingrese todos los
valores.');");
  cliente.println("    return;");
  cliente.println("  }");

  cliente.println("          params     +=     'X='     +
document.getElementById('X').value + '&';");
  cliente.println("          params     +=     'Y='     +
document.getElementById('Y').value + '&';");
  cliente.println("          params     +=     'Z='     +
document.getElementById('Z').value + '&';");

  cliente.println("           fetch('/actualizar?'     +
params).then(response => location.reload());");
  cliente.println("}");
  cliente.println("</script>");

  cliente.println("</body>");
  cliente.println("</html>");
}


void calculo_de_q() {
  float a1 = -2*Z*L1;
  float b1 = 2*L1*(X*cos(0)+Y*sin(0)+r-R);
  float   c1   =   pow(X+cos(0)*(r-R),2)+pow(Y+sin(0)*(r-
R),2)+pow(Z,2)+pow(L1,2)-pow(L2,2);

  float a2 = -2*Z*L1;
  float                      b2                      =
2*L1*(X*cos(radians(120))+Y*sin(radians(120))+r-R);
  float      c2      =      pow(X+cos(radians(120))*(r-
R),2)+pow(Y+sin(radians(120))*(r-
R),2)+pow(Z,2)+pow(L1,2)-pow(L2,2);

  float a3 = -2*Z*L1;
  float                      b3                      =
2*L1*(X*cos(radians(240))+Y*sin(radians(240))+r-R);
  float      c3      =      pow(X+cos(radians(240))*(r-
R),2)+pow(Y+sin(radians(240))*(r-
R),2)+pow(Z,2)+pow(L1,2)-pow(L2,2);

  float   q1rad   =   asin(c1/sqrt(pow(a1,2)+pow(b1,2)))-
atan(b1/a1);
  float   q2rad   =   asin(c2/sqrt(pow(a2,2)+pow(b2,2)))-
atan(b2/a2);
  float   q3rad   =   asin(c3/sqrt(pow(a3,2)+pow(b3,2)))-
atan(b3/a3);

  q1 = (int)(q1rad*(180/PI));
  q2 = (int)(q2rad*(180/PI));
  q3 = (int)(q3rad*(180/PI));
}

void pasopaso() {
  pasos1 = floor((abs(q1-grado1)/1.8)*16);
  pasos2 = floor((abs(q2-grado2)/1.8)*16);
  pasos3 = floor((abs(q3-grado3)/1.8)*16);

  if(q1>grado1 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire1, HIGH);
    for(int i1=0; i1<=pasos1; i1++){
      digitalWrite(paso1, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso1, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q2>grado2 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire2, HIGH);
    for(int i2=0; i2<=pasos2; i2++){
      digitalWrite(paso2, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso2, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q3>grado3 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire3, HIGH);
    for(int i3=0; i3<=pasos3; i3++){
      digitalWrite(paso3, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso3, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q1<grado1 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire1, LOW);
    for(int j1=0; j1<=pasos1; j1++){
      digitalWrite(paso1, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso1, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q2<grado2 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire2, LOW);
    for(int j2=0; j2<=pasos2; j2++){
      digitalWrite(paso2, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso2, LOW);
      delayMicroseconds(retardo);
    }
```

```
  }
  if(q3<grado3 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire3, LOW);
    for(int j3=0; j3<=pasos3; j3++){
      digitalWrite(paso3, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso3, LOW);
      delayMicroseconds(retardo);
    }
  }
}

void calculo_de_s() {
  grado1 = map(s1, 3200, 1790, 0, 90);
  grado2 = map(s2, 3160, 1810, 0, 90);
  grado3 = map(s3, 3160, 2000, 0, 90);

  Serial.print("s1: "); Serial.print(grado1);
  Serial.print(" s2: "); Serial.print(grado2);
  Serial.print(" s3: "); Serial.println(grado3);
}
void control() {
  if (modo == "cordenadas") {
    calculo_de_q();
    pasopaso();
  }
}
```

B. Código para la realización de una secuencia de movimientos del robot delta

```
#define paso_1 15
#define dire_1 2
#define paso_2 13
#define dire_2 12
#define paso_3 25
#define dire_3 26
#define sensor_1 33
#define sensor_2 35
#define sensor_3 32

int R = 60;
int r = 35;
int L1 = 60;
int L2 = 90;
int retardo = 2000;

int secuenciaX[4] = {-30, -30, 30, 30};
int secuenciaY[4] = {-30,  30, 30,-30};
int secuenciaZ[4] = {-90, -90,-90,-90};

void secuencia();
void cinematica(int x, int y, int z, int grado1, int
grado2, int grado3);

void setup() {
  pinMode(paso_1, OUTPUT); pinMode(dire_1, OUTPUT);
  pinMode(paso_2, OUTPUT); pinMode(dire_2, OUTPUT);
  pinMode(paso_3, OUTPUT); pinMode(dire_3, OUTPUT);

  Serial.begin(115200);
  Serial.println("Iniciando secuencia...");
}

void loop() {
  secuencia();
  delay(4000);
}

void secuencia() {
  for (int i = 0; i < 4; i++) {
    int s1 = analogRead(sensor_1);
    int s2 = analogRead(sensor_2);
    int s3 = analogRead(sensor_3);

    int grado1 = map(s1, 3200, 1790, 0, 90);
    int grado2 = map(s2, 3160, 1810, 0, 90);
    int grado3 = map(s3, 3160, 2000, 0, 90);

    Serial.print("Ejecutando paso "); Serial.println(i +
1);
    Serial.print("X: "); Serial.print(secuenciaX[i]);
    Serial.print(", Y: "); Serial.print(secuenciaY[i]);
    Serial.print(", Z: "); Serial.println(secuenciaZ[i]);

    cinematica(secuenciaX[i], secuenciaY[i],
secuenciaZ[i], grado1, grado2, grado3);
    delay(4000);
  }
}

void cinematica(int x, int y, int z, int grado1, int
grado2, int grado3) {
  float a1 = -2*z*L1;
  float b1 = 2*L1*(x*cos(0)+y*sin(0)+r-R);
  float c1 = pow(x+cos(0)*(r-R),2)+pow(y+sin(0)*(r-
R),2)+pow(z,2)+pow(L1,2)-pow(L2,2);

  float a2 = -2*z*L1;
  float b2 =
2*L1*(x*cos(radians(120))+y*sin(radians(120))+r-R);
  float c2 = pow(x+cos(radians(120))*(r-
R),2)+pow(y+sin(radians(120))*(r-
R),2)+pow(z,2)+pow(L1,2)-pow(L2,2);

  float a3 = -2*z*L1;
  float b3 =
2*L1*(x*cos(radians(240))+y*sin(radians(240))+r-R);
  float c3 = pow(x+cos(radians(240))*(r-
R),2)+pow(y+sin(radians(240))*(r-
R),2)+pow(z,2)+pow(L1,2)-pow(L2,2);

  float q1rad = asin(c1/sqrt(pow(a1,2)+pow(b1,2)))-
atan(b1/a1);
  float q2rad = asin(c2/sqrt(pow(a2,2)+pow(b2,2)))-
atan(b2/a2);
  float q3rad = asin(c3/sqrt(pow(a3,2)+pow(b3,2)))-
atan(b3/a3);

  int q1 = q1rad*(180/PI);
  int q2 = q2rad*(180/PI);
  int q3 = q3rad*(180/PI);
  int pasos1 = floor((abs(q1-grado1)/1.8)*16);
  int pasos2 = floor((abs(q2-grado2)/1.8)*16);
  int pasos3 = floor((abs(q3-grado3)/1.8)*16);

  Serial.print("q1: "); Serial.println(q1);
  Serial.print("sensor1: "); Serial.println(grado1);
  Serial.print("q2: "); Serial.println(q2);
  Serial.print("sensor2: "); Serial.println(grado2);
  Serial.print("q3: "); Serial.println(q3);
  Serial.print("sensor3: "); Serial.println(grado3);
  Serial.println("---------------------------------");

  if(q1>grado1 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90) {
    digitalWrite(dire_1, HIGH);
    for(int i1=0; i1<=pasos1; i1++){
      digitalWrite(paso_1, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso_1, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q1<grado1 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire_1, LOW);
    for(int j1=0; j1<=pasos1; j1++){
      digitalWrite(paso_1, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso_1, LOW);
      delayMicroseconds(retardo);
    }
  }


  if(q2>grado2 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire_2, HIGH);
    for(int i2=0; i2<=pasos2; i2++) {
      digitalWrite(paso_2, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso_2, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q2<grado2 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire_2, LOW);
    for(int j2=0; j2<=pasos2; j2++){
      digitalWrite(paso_2, HIGH);
```

```
      delayMicroseconds(retardo);
      digitalWrite(paso_2, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q3>grado3 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire_3, HIGH);
    for(int i3=0; i3<=pasos3; i3++){
      digitalWrite(paso_3, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso_3, LOW);
      delayMicroseconds(retardo);
    }
  }
  if(q3<grado3 && q1>=0 && q1<=90 && q2>=0 && q2<=90 &&
q3>=0 && q3<=90){
    digitalWrite(dire_3, LOW);
    for(int j3=0; j3<=pasos3; j3++){
      digitalWrite(paso_3, HIGH);
      delayMicroseconds(retardo);
      digitalWrite(paso_3, LOW);
      delayMicroseconds(retardo);
    }
  }
}
```

C. Código para encontrar el espacio de trabajo del robot delta

```
clc;
clear;
close all;

vertices = [
    -50,  0, -80;
    -25,-40, -80;
     25,-40, -80;
     50,  0, -80;
     25, 40, -80;
    -25, 40, -80;
    -50,  0, -110;
    -25,-40, -110;
     25,-40, -110;
     50,  0, -110;
     25, 40, -110;
    -25, 40, -110;
];

faces = [
    1,2,3,4,5,6;
    7,8,9,10,11,12;
];
faces2 = [
    1,2,8,7;
    2,3,9,8;
    3,4,10,9;
    4,5,11,10;
    5,6,12,11;
    6,1,7,12;
    ];

figure;
hold on;

patch('Vertices', vertices, 'Faces', faces, 'Faces',
faces2, ...
    'FaceColor', 'green', 'FaceAlpha', 0.3, ...
    'EdgeColor', 'black');

for i = 1:size(vertices, 1)
    text(vertices(i, 1), vertices(i, 2), vertices(i, 3),
...
        sprintf('%d, %d, %d', vertices(i, :)), ...
        'FontSize', 10, 'Color', 'black',
'HorizontalAlignment', 'center');
end

xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
title('Figura del Prisma Rectangular');
grid on;
axis equal;
view(3);

R = 60;
r = 35;
L1 = 60;
L2 = 90;

x_range = -250:5:250;
y_range = -200:5:200;
z_range = 0:-5:-150;

workspace_points = [];
[X, Y, Z] = meshgrid(x_range, y_range, z_range);
points = [X(:), Y(:), Z(:)];

for i = 1:size(points, 1)
    point = points(i, :);
    [q, is_reachable] = inverse_kinematics(point, R, r,
L1, L2);
    if is_reachable
        workspace_points = [workspace_points; point];
    end
end

scatter3(workspace_points(:, 1), workspace_points(:, 2),
workspace_points(:, 3), 10, 'b', 'filled');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
title('Workspace del Robot Delta');
grid on;
axis equal;

function [q, is_reachable] = inverse_kinematics(point, R,
r, L1, L2)
    x = point(1);
    y = point(2);
    z = point(3);

    a1 = -2*z*L1;
    b1 = 2*L1*(x*cosd(0)+y*sind(0)+r-R);
    c1 = (x+cosd(0)*(r-R))^2+(y+sind(0)*(r-
R))^2+(z^2)+(L1^2)-(L2^2);

    a2 = -2*z*L1;
    b2 = 2*L1*(x*cosd(120)+y*sind(120)+r-R);
    c2 = (x+cosd(120)*(r-R))^2+(y+sind(120)*(r-
R))^2+(z^2)+(L1^2)-(L2^2);

    a3 = -2*z*L1;
    b3 = 2*L1*(x*cosd(240)+y*sind(240)+r-R);
    c3 = (x+cosd(240)*(r-R))^2+(y+sind(240)*(r-
R))^2+(z^2)+(L1^2)-(L2^2);

    try
        q1 = asind(c1/sqrt((a1^2)+(b1^2)))-atand(b1/a1);
        q2 = asind(c2/sqrt((a2^2)+(b2^2)))-atand(b2/a2);
        q3 = asind(c3/sqrt((a3^2)+(b3^2)))-atand(b3/a3);

        if all(q1>0 & q1<90) && ...
           all(q2>0 & q2<90) && ...
           all(q3>0 & q3<90)
            is_reachable = true;
        else
            is_reachable = false;
        end
        q = [q1, q2, q3];
    catch
        q = [];
        is_reachable = false;
    end
end
```