UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA

DINÁMICA DE SISTEMAS Y CONTROL

TESIS

DISPOSITIVO EXPERIMENTAL DE ROBOT TIPO DELTA MECÁNICAMENTE RECONFIGURABLE

Presenta:
Ing. Raúl Ismael Serna Ruiz

Director de tesis:	Dr. Juan Manuel Jacinto Villegas
Codirector de tesis:	Dr. Juan Carlos Ávila Vilchis
Tutor adjunto:	Dr. Otniel Portillo Rodríguez

Marzo de 2024

Resumen

En este proyecto de investigación, se abordó el desafío de rediseñar un robot tipo Delta mecánicamente reconfigurable con el objetivo de mejorar su flexibilidad y adaptabilidad en diversas aplicaciones. Dejando el diseño mecánico del robot como *hardware libre* para que los investigadores puedan replicar y personalizar las características del robot según las necesidades de cada usuario o proyecto.

El problema a resolver radica en las limitaciones de los robots convencionales en términos de su falta de versatilidad. Los objetivos específicos del proyecto incluyen el diseño y fabricación de los componentes mecánicos del robot, el desarrollo del sistema eléctrico y de control, la validación del funcionamiento cinemático y la realización de pruebas y análisis de desempeño.

En la etapa uno de la metodología de diseño, se describen los diferentes pasos y etapas involucrados para el desarrollo del robot tipo Delta, desde la definición de los requerimientos del robot hasta la validación del diseño. Se discuten los requerimientos de diseño y las propuestas para los componentes del robot tipo Delta. Se evalúan tres propuestas de diseño, con énfasis en la mejora de componentes. Se evalúan las propuestas y se selecciona el diseño con mejor evaluación. Se realiza un análisis detallado de los componentes clave del robot, utilizando el criterio de Von Mises para evaluar el diseño y resistencia antes de su fabricación, considerando materiales como PLA, aluminio.

En la etapa dos de la metodología se describe cómo se construye el robot desde sus componentes individuales hasta su funcionamiento como un sistema integrado. Se describe el proceso de manufactura de los componentes, incluyendo detalles sobre configuraciones específicas para la impresión 3D y mecanizado. Además, se ilustra el ensamble del robot con un modelo CAD. Se aborda el diseño y la implementación del sistema eléctrico necesario para el funcionamiento del robot. Se presenta la validación del control de posición de los motores DC y se muestra cómo se logra implementar el algoritmo geométrico de la cinemática en un sistema embebido. También se aborda la cinemática inversa y se describen los estados de trabajo del robot que son manipulados mediante una interfaz de usuario para facilitar la interacción con el robot tipo Delta.

Por otro lado, en las pruebas realizadas al robot tipo Delta se evaluó la capacidad del robot al soportar cargas en diferentes configuraciones y posiciones, asegurando que no hubiera daño en los motores. Los resultados de las pruebas señalan que el robot tiene la capacidad de soportar una carga de 28 N dirección positiva de la gravedad en su configuración para trabajos verticales y de 16 N en la configuración para trabajos horizontales. Se validaron las posiciones alcanzadas por el robot comparándolas con las posiciones deseadas, utilizando la interfaz de usuario en Simulink® para capturar y comparar los datos. Se describen las pruebas de precisión y repetibilidad realizadas, obteniendo como resultado una precisión de 0.0037 m y una repetibilidad de 0.0024 m, lo que indica una alta fiabilidad en las operaciones del robot. Se definió el espacio de trabajo del robot tipo Delta como 260 mm x 260 mm x 135 mm, utilizando un modelo

Simulink® para visualizar los movimientos y definir las fronteras del espacio operativo del robot. En conclusión, los resultados demuestran que el robot tipo Delta diseñado posee un rendimiento comparable al de los robots comerciales, a pesar de ser fabricado en su mayoría mediante impresión 3D, lo cual lo convierte en una opción viable para investigadores y laboratorios.

Acrónimos

TRL	Technology Readiness Level
CAD	Computer Aided Design
QFD	Quality Function Deployment
HOQ	House of Quality
PLA	Polylactic Acid (Polímero)
ABS	Acrylonitrile Butadiene Styrene (Polímero)
3D	3 Dimensiones
PCB	Printed Circuit Board
DC	Direct Current
PID	Proporcional Integral Derivativo

Índice general

Resumen II

Acrónimos IV

Índice de tablas VII

Índice de figuras IX

Capítulo 1 Introducción 1
- 1.1 Planteamiento del problema . . . 2
- 1.2 Justificación . . . 3
- 1.3 Meta de Ingeniería . . . 3
- 1.4 Objetivos . . . 3
  - 1.4.1 Objetivo general . . . 3
  - 1.4.2 Objetivos específicos . . . 3
- 1.5 Alcances y limitaciones . . . 4
  - 1.5.1 Alcances . . . 4
  - 1.5.2 Limitaciones . . . 4
- 1.6 Metodología propuesta . . . 4

Capítulo 2 Antecedentes 6
- 2.1 Marco teórico . . . 6
  - 2.1.1 Tipos de robots . . . 6
  - 2.1.2 Robot Delta . . . 6
  - 2.1.3 Dispositivos comerciales . . . 8
  - 2.1.4 Precisión y repetibilidad de un robot . . . 9
  - 2.1.5 Análisis de elemento finito . . . 10
  - 2.1.6 Esfuerzo mediante criterio de Von Mises . . . 11
  - 2.1.7 Quality Function Deployment (QFD) . . . 11
- 2.2 Estado del arte . . . 12

Capítulo 3 Desarrollo de la metodología - Etapa 1 19
- 3.1 Requerimientos de diseño . . . 19
- 3.2 Diseño de componentes en modelo CAD . . . 21
  - 3.2.1 Primera propuesta . . . 22
  - 3.2.2 Segunda propuesta . . . 23
  - 3.2.3 Tercera propuesta . . . 25
  - 3.2.4 Casa de la Calidad . . . 26
- 3.3 Análisis de elemento finito a componentes del robot . . . 28
  - 3.3.1 Análisis de elemento finito del componente RD.01.135 - Capstan 29
  - 3.3.2 Análisis de elemento finito del componente RD.01.130 - Perno . 31
  - 3.3.3 Pruebas físicas de resistencia de PLA y ABS . . . 33

Capítulo 4 Desarrollo de la metodología - Etapa 2 35
4.1 Manufactura de componentes 35
4.2 Desarrollo del sistema eléctrico para el robot tipo Delta 39
4.3 Ensamble del robot Delta 40
4.4 Control de posición de Motores de DC 42
4.4.1 Validación del control de posición de los motores 44
4.5 Implementación del algoritmo geométrico de la cinemática en un sistema embebido 47
4.5.1 Dimensiones y masas de los componentes 50
4.5.2 Validación de la cinemática directa 51
4.5.3 Cinemática Inversa 52
4.5.4 Estados de trabajo del robot tipo Delta 54
4.5.5 Interfaz para el usuario 54

Capítulo 5 Resultados y discusión 56
5.1 Pruebas de carga 56
5.2 Pruebas de posición 58
5.2.1 Pruebas de precisión y repetibilidad. 60
5.3 Espacio de trabajo. 63
5.4 Discusión 65

Capítulo 6 Conclusiones y trabajo futuro 67
6.1 Trabajo futuro 68

Apéndices 74

Apéndice A Diagramas mecánicos 2D 75

Apéndice B Esquemático PCB 92

Apéndice C Dimensiones del Robot tipo Delta 95

Apéndice D Script de Matlab® para cargar dimensiones del Robot tipo Delta 99

Apéndice E Script para validación de la cinemática directa. 104

Apéndice F Software STM32CubeIDE 111

Apéndice G Modelos Simulink®. 127

Índice de tablas

2.1 Características principales de dispositivos hápticos comerciales. 17
2.2 Características principales de dispositivos hápticos diseñados por investigadores. 18

3.1 Componentes de la primera propuesta. 23
3.2 Componentes de la segunda propuesta. 24
3.3 Componentes de la tercera propuesta. 26
3.4 Configuraciones, pruebas físicas de carga y resultados de impresión con material PLA y ABS, realizadas en el perno. 34

4.1 Configuración para impresión 3D para pruebas de funcionamiento en PLA. 35
4.2 Configuración para impresión 3D de componentes del robot tipo Delta. 38
4.3 Ganancias iniciales de controladores. 44
4.4 Ganancias para controlador PD. 46
4.5 Ganancias para controlador PID. 46
4.6 Dimensiones del robot tipo Delta. 50
4.7 Masas de los componentes del robot tipo Delta. 51

5.1 Error promedio de los motores para 15 posiciones diferentes. 58
5.2 Posiciones enviadas en el estado 3 al EF. 59
5.3 Error promedio en ejes del EF para 15 posiciones diferentes 60
5.4 Posiciones alcanzadas por el efector final del robot tipo Delta en (x,y,z). 61
5.5 Gastos en material para construcción de robot tipo Delta. 65
5.6 Comparación de características del robot desarrollado en este proyecto contra robots comerciales. 66

Índice de figuras

1.1 Metodología propuesta. 5

2.1 Robot paralelo tipo Delta, editado de Alvarez [1]. 7
2.2 Representación del espacio de trabajo de un robot Delta, editada de Omron [2]. 7
2.3 Simulador de entrenamiento quirúrgico, Lee [3]. 8
2.4 Robots teleoperados aplicados a diferentes aplicaciones. (a) Teleoperación aplicada a los sistemas Quirúrgicos y Cirugía, Robot DaVinci [4]. (b) Robot tEODor, para desarme de explosivos, Siciliano [5]. 9
2.5 Dispositivos hápticos comerciales. (a) Novint Falcon, de Novint Technologies [6]. (b) Dispositivo Delta.3, Force Dimension [7]. 9
2.6 Visualización de precisión y repetibilidad, Barrientos [8]. 10
2.7 Casa de la Calidad, editada de Duffy et al [9]. 11

3.1 Modelo descriptivo para el proceso de diseño, editada de Howard et al [10]. 21
3.2 Diseño de la primera propuesta. 22
3.3 Diseño de la segunda propuesta. 24
3.4 Diseño de la tercera propuesta. 25
3.5 Casa de la calidad. 27
3.6 Nombre y No. de parte de componentes. 28
3.7 Propiedades del material PLA, revisado en Octubre de 2022 [11, 12]. . . 29
3.8 Restricciones para análisis de elemento finito a capstan. (a) Sujeción (flechas color verde) y carga en capstan (flechas color púrpura). (b) Mallado de capstan. 30
3.9 Análisis de esfuerzos - capstan. 30
3.10 Desplazamiento - capstan. 31
3.11 Restricciones para análisis de elemento finito al perno. (a) Sujeción y carga en perno. (b) Mallado de perno. 31
3.12 Análisis de esfuerzos - perno. 32
3.13 Desplazamiento - perno. 32
3.14 Banco de pruebas con sensor de fuerza. 33
3.15 Carga actuada en extremo del perno. 34

4.1 Modelo CAD para transmisión por cable. 36
4.2 Ensamble de componentes 3D para pruebas de funcionamiento. 37
4.3 Fabricación de bases del robot mediante cortadora Láser. (a) Base de robot Delta para trabajos horizontales. (b) Base de robot Delta para trabajos verticales. 38
4.4 Barras paralelas de aluminio 6061. 38
4.5 Perno de aluminio para articulación. 39

4.6 Casquillo de latón. 39
4.7 Vista 3D de PCB con tarjeta montada. 40
4.8 (a) Configuración para trabajos verticales. (b) Configuración para trabajos horizontales. 40
4.9 robot Delta, sistema electrónico y fuente de alimentación. 42
4.10 Respuesta transitoria de motores. 43
4.11 Diagrama de bloques en lazo cerrado para la sintonización de los controladores PID en tiempo continuo. 43
4.12 Sintonizado del bloque PID mediante *PIDTuner*. 44
4.13 Respuesta de controlador PID con sintonización fina para eliminar sobreimpulso. 45
4.14 Respuesta del controlador PD con sintonización fina. 45
4.15 Respuesta sintonizada del controlador PID. 46
4.16 Dimensiones del robot tipo Delta. 47
4.17 Propiedades geométricas utilizadas por Avizzano et al. para resolver las cinemáticas de un robot tipo Delta, tomada de [13] 48
4.18 Representación de posición más retraída y extendida del robot tipo Delta. 52
4.19 Simulación de los ángulos de desplazamiento del Capstan. 52
4.20 Modelo CLIK, Siciliano [14]. 53
4.21 Descripción detalla a bloques del CLIK. 53
4.22 (a) Configuración serial de la interfaz en recepción y transmisión. (b) Displays y selectores. 55

5.1 Pruebas de carga en configuración vertical. (a) Carga máxima soportada. (b) Carga máxima soportada por el robot al desplazarse a una posición. 57
5.2 Pruebas de carga en configuración horizontal. (a) Pruebas en posición retraída. (b) Pruebas en posición intermedia. (c) Pruebas en posición extendida. 58
5.3 Pruebas de posición en el estado 2 del robot tipo Delta. 59
5.4 Pruebas de posición en estado 3 del robot tipo Delta. 60
5.5 Pruebas de precisión y repetibilidad en ejes *XY*. 61
5.6 Pruebas de precisión y repetibilidad en ejes *XZ*. 62
5.7 Espacio de trabajo en *XYZ*. 63
5.8 (a) Espacio de trabajo en plano *XY*. (b) Espacio de trabajo en plano *XZ*. 64

Capítulo 1

Introducción

Los robots son dispositivos utilizados para el desarrollo y beneficio de la humanidad. En la actualidad la mayoría de la población tiene una idea de lo que es un robot y conoce algunas de sus aplicaciones.

Por otro lado, la robótica como concepto puede incluir la creación, planificación y programación de robots. Actualmente, los robots están en contacto directo con el mundo y para el caso del robot paralelo tipo Delta, su uso común se encuentra en el sector industrial realizando tareas repetitivas y con alto grado de precisión, conocidas como *pick and place*. En la telerobótica los robots se controlan a distancia mediante el método maestro-esclavo y en ambientes virtuales donde se requiera utilizar la retroalimentación de fuerza en el efector final (EF) cuando existe una colisión con objetos virtuales. Sin embargo, el elevado costo de esta tecnología implica que no se encuentre al alcance de universidades públicas y laboratorios con recursos limitados.

Este proyecto presenta el desarrollo del diseño mecánico para la fabricación de un robot paralelo, también conocido como robot Delta. Estos robots se conforman por una base fija conectada por una serie de eslabones que se unen mediante articulaciones siguiendo una sola dirección y conectándose a una base móvil donde se ensambla un EF.

Por lo tanto, el principal objetivo del proyecto es rediseñar la mecánica de un dispositivo háptico tipo Delta que permita reconfigurarlo para realizar trabajos de forma vertical y horizontal, que sea hardware libre *(open-source hardware)* y que pueda ser implementado en múltiples aplicaciones de teleróbotica y ambientes virtuales.

La tesis se encuentra dividida de la siguiente forma. En este Capítulo se presenta el planteamiento del problema, la justificación, meta de ingeniería, objetivos, los alcances, limitaciones y la metodología propuesta para el desarrollo de este proyecto. El Capítulo 2 contiene los antecedentes del proyecto, el marco teórico donde se presentan los fundamentos de los robots, sus aplicaciones y las características principales de los dispositivos comerciales que existen en la actualidad. Además, se presenta el estado del arte, que contiene la literatura que se ha tomado como base para el desarrollo de este proyecto.

En el Capítulo 3 se describe la primera etapa del desarrollo de la metodología empleada para este proyecto, esta etapa contiene los requerimientos de diseño, el diseño de componentes en modelo CAD y el análisis de elemento finito a componentes del robot. El Capítulo 4 contiene la segunda etapa en el desarrollo de la metodología, la cual contiene la manufactura de componentes, donde se mencionan las características de fabricación, los materiales y sus configuraciones; el desarrollo del sistema eléctrico, el

ensamble del robot tipo Delta, el control de posición de los motores DC y su validación; la implementación del algoritmo geométrico de la cinemática en un sistema embebido, la validación de la cinemática directa e inversa; la descripción de los estados operativos del robot y su interfaz para el usuario. En el Capítulo 5 se describen las pruebas de carga, posición, precisión y repetibilidad; los resultados de las pruebas, su espacio de trabajo del robot y una breve discusión de los resultados. En el Capítulo 6, se mencionan las conclusiones de la investigación y el trabajo futuro.

1.1. Planteamiento del problema

De acuerdo con lo que se presenta en el estado del arte (Capítulo 2.2), el único dispositivo háptico que puede ser utilizado de forma horizontal y vertical es el delta.3; sin embargo, tiene un costo de 34,858.97 USD [7], siendo este uno de los mayores inconvenientes debido a que es un precio que no es accesible para escuelas públicas o investigadores con recursos limitados, de manera que solo es accesible para un grupo reducido de escuelas e investigadores en México.

Adicionalmente, otra de las problemáticas que presentan estos dispositivos hápticos son que sus efectores finales están diseñados para aplicaciones específicas. Por lo tanto, se requiere adquirir aditamentos especiales para acoplar un nuevo EF como lo mencionan en [15]. Así mismo, si se diseña un EF que agregue más grados de libertad (máximo tres), éste EF, al ser acoplado al dispositivo no sería considerado en la cinemática del controlador debido a que los dispositivos comerciales solo proporcionan el control de fuerza y posición. Por lo tanto, para que este EF sea considerado en la cinemática se debe contactar con el distribuidor del dispositivo háptico para que se realicen las modificaciones por un costo adicional.

Así también, los dispositivos hápticos comerciales tienen espacios de trabajo predefinidos para ciertas actividades; como se menciona en [16], estos espacios de trabajo se encuentran en un rango entre 160 x 160 x 110 mm y 400 x 400 x 260 mm en traslación.

Por otro lado, en el caso del dispositivo comercial Novint Falcon que proporciona una retroalimentación de fuerza máxima de 8.8 N, cuando la aplicación requiere de una fuerza mayor o constante durante un tiempo determinado, por ejemplo en colisiones con objetos virtuales rígidos, si la fuerza es superada dentro del ambiente virtual es posible que se pierda la inmersión en el mismo y que el dispositivo háptico no proporcione la retroalimentación adecuada, esto debido a la seguridad del dispositivo que desactiva los motores para evitar que sufran daños.

Por el contrario, considerando los dispositivos hápticos tipo Delta no comerciales que son desarrollados por investigadores, estos dispositivos son implementados para una aplicación específica, lo que dificulta la implementación de estos dispositivos en aplicaciones diferentes a la inicial.

Por lo tanto, se propone el rediseño de un Robot tipo Delta, que se pueda reconfigurar para trabajos de forma horizontal y vertical, que pueda ser implementado en múltiples aplicaciones de telerobótica y ambientes virtuales, además, que sea hardware libre *(open-source hardware)* para que pueda ser fabricado por universidades o laboratorios que no cuenten con los recursos necesarios para adquirir un robot comercial.

1.2. Justificación

De acuerdo a lo presentado en el planteamiento del problema, se propone desarrollar en este proyecto el rediseño de un dispositivo háptico tipo Delta que permita su reconfiguración para que pueda realizar trabajos de forma vertical y horizontal, según se requiera en la aplicación.

El dispositivo podrá ser implementado en múltiples aplicaciones de telerobótica y ambientes virtuales. Así mismo, el diseño mecánico será *hardware libre*, permitiendo que investigadores, universidades y laboratorios que no cuentan con los recursos necesarios para adquirir un robot comercial, puedan fabricar este dispositivo y mantenerlo de forma funcional. Además, al ser *hardware libre* el usuario tendrá la libertad de diseñar su propio EF y manipular la cinemática del mismo de acuerdo a su aplicación, para posteriormente agregarlo a la cinemática del dispositivo en caso de requerir más grados de libertad actuados en el EF.

Por otra parte, el dispositivo proporciona un espacio de trabajo superior al promedio de los dispositivos comerciales, siendo este de 260 x 260 x 140 mm. Así también, proporciona una fuerza superior al promedio de los dispositivos comerciales (15 N), y que ha sido utilizada en el desarrollo de las siguientes investigaciones [15, 17–21].

1.3. Meta de Ingeniería

Rediseñar la mecánica de un dispositivo háptico tipo Delta reconfigurable para realizar tanto trabajos verticales como horizontales, implementar la electrónica, el sistema de control y de comunicaciones, que sea hardware libre *(open-source hardware)*, que soporte una carga de 15 N en cualquier dirección, carga suficiente para que pueda ser implementado en aplicaciones de telerobótica y ambientes virtuales.

1.4. Objetivos

1.4.1. Objetivo general

Rediseñar la mecánica de un dispositivo háptico tipo Delta que permita reconfigurarlo de forma vertical y horizontal, implementar la electronica, el sistema de control y comunicaciones, que sea *hardware libre* y que pueda ser implementado en múltiples aplicaciones de teleróbotica y ambientes virtuales.

1.4.2. Objetivos específicos

- Definir requisitos del diseño (selección de motores y material para complementos, GDL, torques, fuerzas, tolerancias, cableado y mecanismo para efectores finales creados).
- Diseñar componentes del dispositivo háptico en modelo CAD.
- Permitir el libre acceso al diseño mecánico.
- Realizar análisis de elemento finito a componentes del robot.
- Implementar la electrónica necesaria para el control de actuadores del robot Delta.

- Programar la cinemática del dispositivo háptico en un sistema embebido basado en el modelo geométrico de Avizzano et al. [13].
- Validar la posibilidad de agregar diferentes tipos de efectores finales al mecanismo de la base móvil del robot Delta.
- Ampliar el espacio de trabajo y fuerza que proporciona el dispositivo háptico con respecto a un promedio de los dispositivos comerciales y no comerciales.
- Validar de forma experimental los movimientos del dispositivo háptico propuesto (carga, posición, precisión, repetibilidad y espacio de trabajo).

1.5. Alcances y limitaciones

1.5.1. Alcances

- Un dispositivo háptico tipo Delta reconfigurable para implementación en trabajos horizontales y verticales.
- Contar con un dispositivo háptico que proporciona una fuerza y espacio de trabajo promedio a los dispositivos comerciales y no comerciales.
- Un dispositivo háptico tipo Delta para implementación en múltiples aplicaciones (teleoperación y ambientes virtuales).
- Un dispositivo háptico de hardware libre que permita el diseño de su propio efector final y manipular la cinemática del mismo de acuerdo a su aplicación.

1.5.2. Limitaciones

- El dispositivo solo se llegará a un TRL 4 (método que mide el grado de madurez de las tecnologías) como producto para pruebas de laboratorio.
- El funcionamiento del dispositivo se validará implementando la cinemática basada en el algoritmo geométrico propuesto en [13].

1.6. Metodología propuesta

En la Figura 1.1 se muestra la metodología con la que se desarrolló este proyecto de investigación. No obstante, es preciso mencionar que no es parte del objetivo desarrollar el algoritmo cinemático de control del dispositivo propuesto, por lo tanto, solo se validará su funcionamiento implementando la cinemática basada en el algoritmo geométrico propuesto en [13].

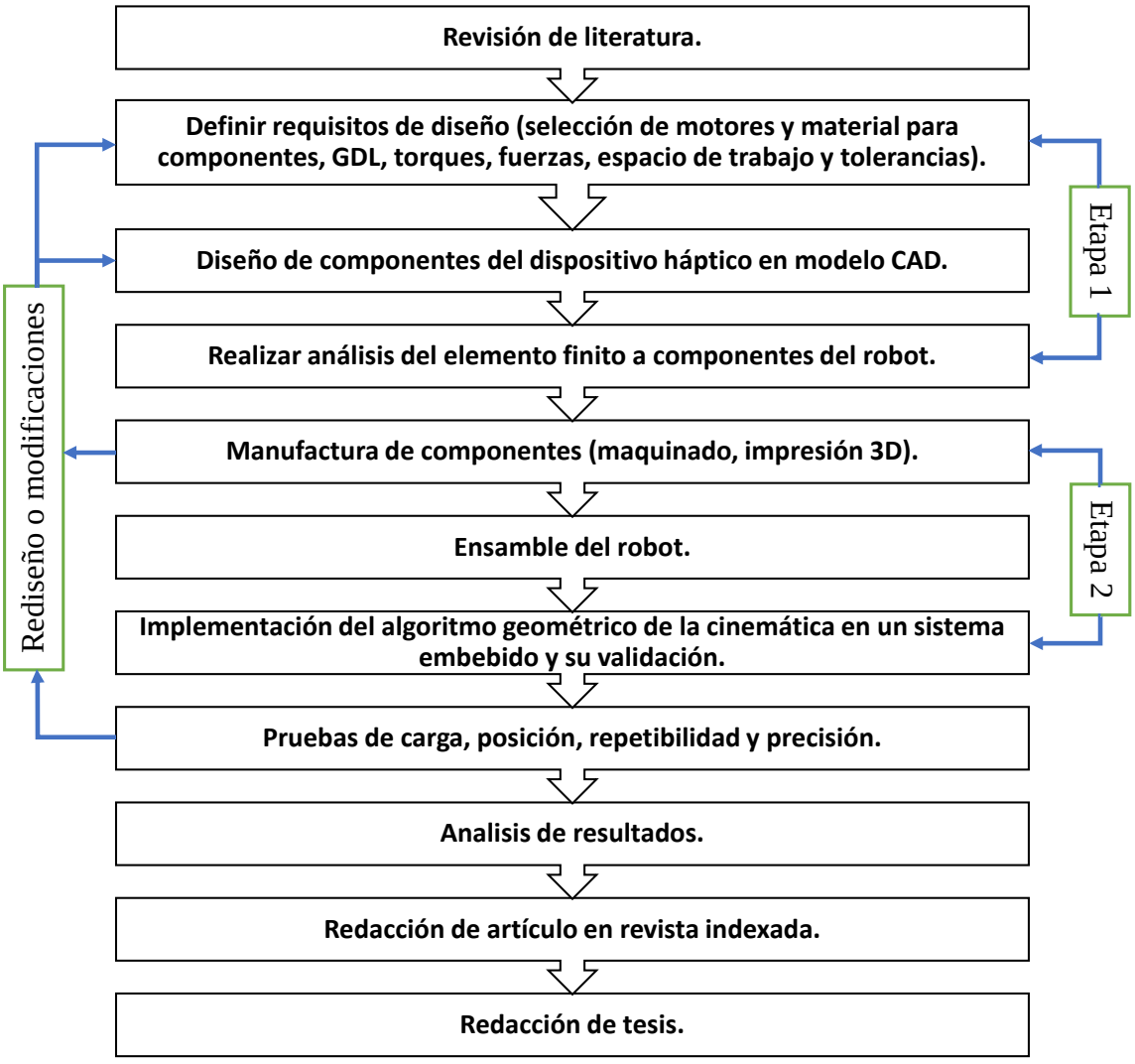


Figura 1.1: Metodología propuesta.

Capítulo 2

Antecedentes

2.1. Marco teórico

2.1.1. Tipos de robots

De acuerdo con su estructura cinemática, los robots o manipuladores se clasifican como robots serie y robots paralelos. Un robot serie es aquel que tiene una cadena cinemática abierta, es decir, que su conjunto de eslabones están conectados por articulaciones siguiendo una sola dirección. Por otro lado, los robots paralelos se denominan de cadena cinemática cerrada si cada uno de sus eslabones están conectados a otros por dos o mas articulaciones, formando una o dos cadenas cerradas y que a su vez conectan su base con el efector final (EF). Así también, estos robots paralelos pueden dar como resultado una mayor rigidez estructural y fuerzas, por lo tanto, mayor precisión de posicionamiento, mayores velocidades y aceleraciones que los robots de cadena cinemática abierta. Además, los robots que están compuestos por cadena cinemática abierta y cerrada se les conoce como robots híbridos [14, 22, 23].

Sin embargo, esta tesis se enfoca en el robot tipo Delta que se explica a detalle en la siguiente sección.

2.1.2. Robot Delta

El robot Delta (Figura 2.1) se encuentra en la clasificación de los robots paralelos, tiene tres grados de libertad (GDL) y está formado por los siguientes componentes [14, 23]:

- Base fija.
- Base móvil.
- Eslabones.
- Articulaciones.
- Efector final (ensamblado en la base móvil).

Cabe destacar que debido a su estructura y a las fuerzas que puede generar el robot tipo Delta, éste puede ser utilizado para su implementación en aplicaciones donde se requiera retroalimentación de fuerza para colisión con objetos virtuales rígidos o deformables [24, 25].

En cuanto a el espacio de trabajo del robot tipo Delta, este es menor comparado con el de un robot serie, lo que es una desventaja a considerar al seleccionar este tipo de

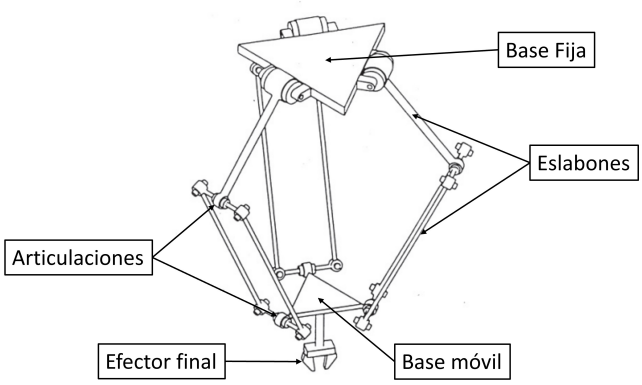


Figura 2.1: Robot paralelo tipo Delta, editado de Alvarez [1].

robot [14,26]. Otra desventaja es que la descripción cinemática de los robots paralelos es diferente a los de cadena cinemática abierta, por lo tanto, requieren diferentes métodos de análisis [23].

El espacio de trabajo de un robot manipulador representa el volumen total que es alcanzable por el EF del mismo al momento de ejecutar los movimientos deseados durante una actividad. Además, éste se encuentra limitado por la geometría del manipulador [23]. El espacio de trabajo de un robot tipo Delta se presenta en la Figura 2.2, donde los ejes $(\boldsymbol{X}, \boldsymbol{Y})$ representan los puntos dentro del alcance del robot y también pueden ser representados como diámetro $(\varnothing)$. En cuanto al eje $\boldsymbol{Z}$, este representa la distancia máxima en la que puede desplazarse el EF, por ejemplo, cuando se lleva el EF a la posición más extendida o retraída.

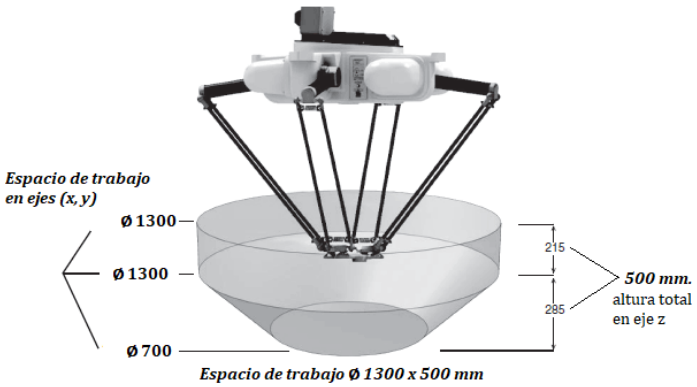


Figura 2.2: Representación del espacio de trabajo de un robot Delta, editada de Omron [2].

Aplicaciones

Existen varias aplicaciones para el robot tipo Delta, sin embargo, en este proyecto se deja abierto para su implementación en las siguientes aplicaciones:

- Ambientes virtuales (AV).
- Háptica.
- Telerobótica.

Los ambientes virtuales (AV) son aquellos en los que se proporciona la ilusión de presencia en un lugar diferente al del entorno físico real. La percepción en un AV se basa en estímulos generados por computadora. Estos estímulos suelen ser de naturaleza visual y auditiva, pero ocasionalmente pueden ser hápticos. En otras palabras, los estímulos artificiales generados por computadora tratan de crear la sensación de inmersión en el AV [27].

En la literatura sobre robótica y realidad virtual, la háptica se define en [5] como "interacciones táctiles reales y simuladas entre robots, humanos y entornos reales". La háptica también es conocida como la tecnología para experimentar y crear sensaciones táctiles en operadores humanos.

Un dispositivo háptico común lo podemos encontrar en los videojuegos, por ejemplo, el joystick es un dispositivo que genera vibraciones o produce resistencia cuando retroalimenta fuerza mientras el operador juega un videojuego. Otro dispositivo háptico es el que permite simular procedimientos médicos prácticos (ver Figura 2.3); estos simuladores tienen como objetivo entrenar al usuario y reemplazar el aprendizaje supervisado directamente en pacientes humanos o en animales [5]. El teléfono celular es otro ejemplo de dispositivo háptico, este genera retroalimentación vibrotáctil al momento de que el usuario entra en contacto con la pantalla del celular, este percibe pequeñas vibraciones en la piel [5].

Figura 2.3: Simulador de entrenamiento quirúrgico, Lee [3].

La telerobótica es un área dentro de la robótica dedicada al control de robots a distancia, donde una parte importante es la interfaz hombre-máquina, que consiste en el diseño de interfaces donde la comunicación debe ser lo más natural posible para que los movimientos del hombre sean capturados y reproducidos por el robot [28]. El operador humano puede interactuar en un ambiente virtual utilizando tres componentes principales: un manipulador maestro, un manipulador esclavo y un medio de comunicación interactivo entre los manipuladores maestro y esclavo [3]. Algunas aplicaciones prácticas de la telerobótica se encuentran en entornos inaccesibles o peligrosos, aplicaciones médicas que incluyen telecirugía y cirugía mínimamente invasiva (ver Figura 2.4a), controles remotos de ultrasonido, operaciones espaciales y submarinas, tareas militares (ver Figura 2.4b) y operaciones en entornos nucleares donde estén en contacto con desechos radiactivos [3].

2.1.3. Dispositivos comerciales

Existen dispositivos hápticos fabricados por empresas comerciales (ver Figura 2.5), sin embargo, también se encuentran los dispositivos desarrollados por investigadores para fines especiales. En la Figura 2.5a se observa el Novint Falcon, que en el año

(a) (b)

Figura 2.4: Robots teleoperados aplicados a diferentes aplicaciones. (a) Teleoperación aplicada a los sistemas Quirúrgicos y Cirugía, Robot DaVinci [4]. (b) Robot tEODor, para desarme de explosivos, Siciliano [5].

2007 fue uno de los dispositivos hápticos de cadena cinemática cerrada más populares y accesibles económicamente, no obstante en la actualidad ya no es fabricado. Este dispositivo cuenta con transmisión por cable, proporciona una fuerza de 8.8 N y un espacio de trabajo de 101.6 x 101.6 x 101.6 mm. Así mismo, en la actualidad se encuentran disponibles los dispositivos fabricados por la empresa Force Dimension (ver Figura 2.5b). Esta empresa cuenta con un dispositivo que es capaz de realizar trabajos tanto en forma vertical como horizontal. Además, sus dispositivos proporcionan una fuerza igual o mayor a los 12 N, dependiendo el modelo que se elija, ya que cuenta con dispositivos que pueden implementarse en tareas más especializadas.

(a) (b)

Figura 2.5: Dispositivos hápticos comerciales. (a) Novint Falcon, de Novint Technologies [6]. (b) Dispositivo Delta.3, Force Dimension [7].

2.1.4. Precisión y repetibilidad de un robot

La precisión puede definirse como la capacidad del robot de llevar su EF a una misma posición y orientación, una y otra vez de acuerdo a las posiciones programadas [8], es decir, es la distancia entre un punto programado y el promedio de los puntos alcanzados al reiterar el movimiento en varias ocasiones bajo las mismas condiciones. Sin embargo, la precisión se puede ver afectada por los siguientes factores: errores de calibración en el robot, imperfecciones causadas por factores térmicos y dinámicos, error en el cálculo de la cinemática, y discrepancias entre las dimensiones teóricas y reales del robot [8].

De acuerdo con Lasheras en [29] y la Norma ISO 9283:1998, una posible métrica para la precisión puede ser la siguiente:

- Lanzar el robot desde diferentes configuraciones a un destino (x_d, y_d, z_d).
- Posiciones alcanzadas (x_i, y_i, z_i).
- Calcular el error medio cuadratico (RMS) en x,y,z.
- Integrar los RMS como suma euclídea.
- La precisión podrá ser la media o la del peor caso.

Por otra parte, la repetibilidad es definida como la capacidad de un robot para regresar a una posición previamente alcanzada desde el mismo origen y bajo las mismas condiciones [5,14]. También, en [8] se menciona que "la repetibilidad de posicionamiento expresa la dispersión de las posiciones alcanzadas tras visitar n veces en la misma dirección una posición programada".

Una alternativa para obtener la repetibilidad es la que se menciona en [29], donde se consideran las siguientes métricas:

- Lanzar el robot desde diferentes configuraciones a un destino (x_d, y_d, z_d).
- Posiciones alcanzadas (x_i, y_i, z_i).
- Calcular la desviación estándar en x,y,z.
- Integrar la desviación estándar como suma euclídea.

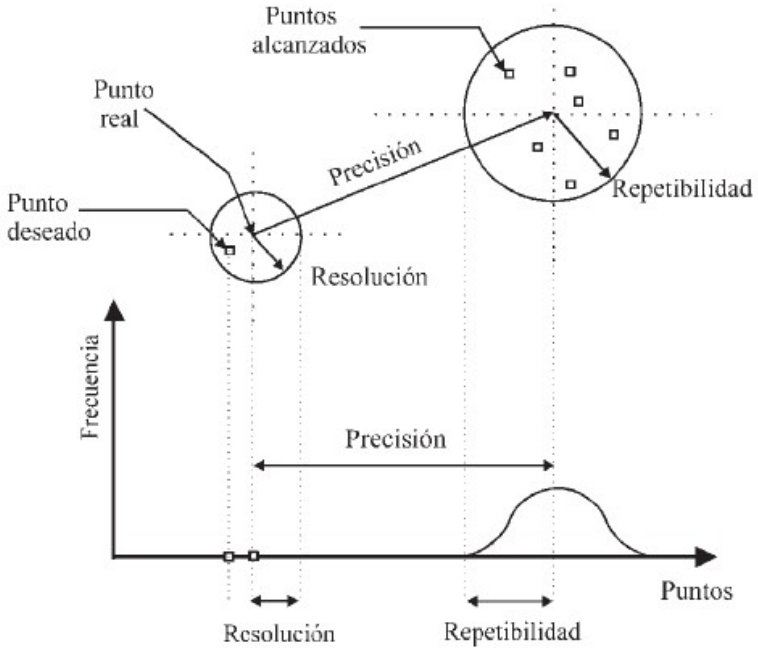


Figura 2.6: Visualización de precisión y repetibilidad, Barrientos [8].

Adicionalmente, se ha encontrado que el error de repetibilidad en robots industriales oscila entre 0.5 y 0.01 milímetros [8].

2.1.5. Análisis de elemento finito

El análisis de elemento finito es una simulación que sirve para reducir la cantidad de prototipos y experimentos físicos realizados al diseñar un componente, gracias a que

la simulación nos permite someterlo a esfuerzos para validar la resistencia del diseño y el material seleccionado. Además, permite modificar los componentes al desarrollar productos sin gastos de fabricación [30].

De modo que, para este proyecto utilizaremos el programa Solidworks® para realizar el análisis de elemento finito, el cual permite determinar si el diseño de los componentes del dispositivo soportará las cargas a las que será sometido.

2.1.6. Esfuerzo mediante criterio de Von Mises

El esfuerzo de Von Mises es un valor utilizado para determinar si un material dado experimentará deformación o fractura. El criterio de Von Mises indica que si la tensión de Von Mises en un material bajo una carga es igual o superior al límite de fluencia del mismo material bajo tensión simple, entonces el material se deformará. [31]. El esfuerzo de Von Mises lo definen en [32] "como aquel esfuerzo a tensión uniaxial que generaría la misma energía de distorsión que la que se produciría por la combinación real de los esfuerzos aplicados (tensión y cortante)".

2.1.7. Quality Function Deployment (QFD)

La QFD es una herramienta utilizada por empresas de diseño que sirve para identificar y satisfacer las necesidades del cliente o usuario, transforma las necesidades en características para una etapa de desarrollo del concepto hasta llegar al diseño y producción del mismo. Además, permite priorizar dichas necesidades y generar soluciones innovadoras que permiten diseñar o mejorar el proceso de entrega, sobrepasando las expectativas del cliente [9, 33].

Su principal forma de implementación de la QFD es conocida como la casa de la calidad (House Of Quality - HOQ), la cual consiste en una serie de matrices que están divididas en secciones que dan forma a una casa (ver Figura 2.7).

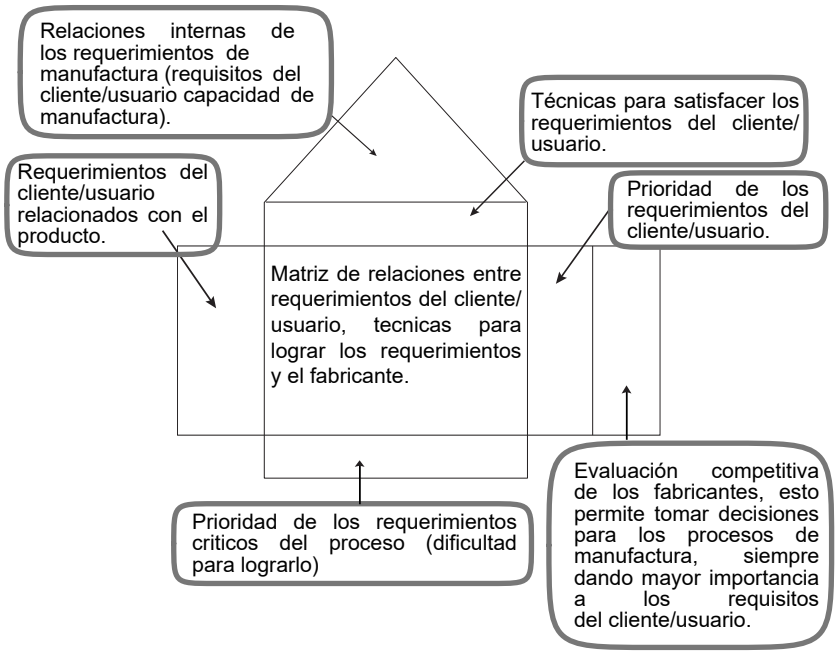


Figura 2.7: Casa de la Calidad, editada de Duffy et al [9].

Por lo tanto, para esta tesis se utiliza la casa de la calidad como la principal herramienta para evaluar cada propuesta de diseño que se genere y seleccionar la que mejor cumpla con los requerimientos de diseño.

2.2. Estado del arte

La empresa Novint Technologies fabricó uno de los dispositivos hápticos más accesibles económicamente que estuvieron disponibles comercialmente desde 2007 hasta 2013 y fue nombrado "Novint Falcon". Dicho dispositivo háptico actualmente ya no se fabrica porque la empresa se retiró del mercado; tuvo un costo de alrededor de los USD $1000. En la actualidad, este dispositivo solo puede ser adquirido mediante un proveedor externo llamado HapticsHouse a un precio de USD $499 [34]. El dispositivo es de cadena cinemática cerrada, proporciona una fuerza de 8.8 N, consta de tres GDL de tal manera que su EF puede desplazarse en un espacio de trabajo de 101.6 milímetros en los tres ejes (x,y,z), proporciona una resolución de 1 mm en el EF; así también, utiliza tres motores DC como actuadores. La empresa proporcionaba un Kit de Desarrollo de Software (SDK, acrónimo por su traducción en inglés, Software Development Kit), que permite el control del dispositivo a través de comandos que se envían por el protocolo de comunicación USB [19, 35, 36].

En la literatura se encuentran diversas aplicaciones en cuanto al Novint Falcon, Scalona et al. presentan en [36], donde utilizan el dispositivo háptico Novint Falcon para evaluar el aprendizaje motor en personas durante la realización de tareas de alcance en un espacio con perturbaciones; las tareas fueron realizadas en un escenario de realidad virtual donde se evaluaron tareas solicitadas en ocho direcciones con perturbaciones y campos de fuerza incrementales, sus resultados les permitieron concluir que sí es posible implementar este dispositivo háptico como una alternativa práctica para rehabilitación en personas.

En [19] los autores desarrollan un sistema de simulación de entrenamiento periodontal háptico, llamado Haptodont, para simular un sondeo periodontal, donde las interacciones de una sonda y un espejo son simuladas por el dispositivo háptico Geomagic Touch, mientras que el dispositivo Novint Falcon se utilizó para implementar el soporte para los dedos.

Moo-Young et al. en [6] describen el diseño y desarrollo de un sistema de entrenamiento epidural virtual. Este entrenamiento también fue desarrollado aprovechando la retroalimentación visual y la háptica que ofrece el dispositivo Novint Falcon. Finalmente, este proceso demostró que es posible desarrollar sistemas de una manera que presente soluciones potenciales que pueden ayudar a practicar las habilidades médicas.

Menges et al. en [37], presentan una aplicación en un entorno de fundición restringido, donde el operador debe realizar tareas de limpieza como cepillado y raspado en condiciones peligrosas; para esto, implementan la teleoperación mediante el dispositivo háptico Novint Falcon utilizado como maestro, y el brazo robótico Franka como esclavo, que es el encargado de reproducir los movimientos para ejecutar la tarea de limpieza sin exponer al usuario.

En 2021 Park et al. en [38], propusieron el uso del dispositivo háptico Novint Falcon para enseñanza de la escritura a mano en niños con retrasos cognitivos y motricidad fina en función de la complejidad de la tarea de escritura a mano, diseñaron su propio agarre personalizado en forma de lápiz que cuenta con 3 grados de libertad para

proporcionar retroalimentación de fuerza. Realizaron dos grupos para evaluar el funcionamiento, donde los resultados demostraron que el grupo objetivo se desempeñó significativamente mejor que el grupo de control en tareas de escritura que son visualmente familiares pero hápticamente difíciles. También se observó una mejora en el desempeño de tareas no entrenadas, así como de tareas entrenadas.

Gutiérrez et al. en [20], muestran una plataforma de realidad virtual para la rehabilitación de miembros superiores con monitorización basada en electromiografía, que proporciona retroalimentación de las contracciones musculares del paciente. Para este proceso de rehabilitación se proporcionó retroalimentación háptica a los pacientes para fortalecer los músculos del brazo mediante el dispositivo Novint Falcon. Los resultados demostraron que la plataforma de rehabilitación es adaptable a las necesidades de los pacientes, generando una buena respuesta háptica y un seguimiento fiable de las contracciones musculares.

Por otra parte, los robots paralelos fabricados por Force Dimension [7] son utilizados para diferentes aplicaciones, estos dispositivos son los siguientes: el delta.3, omega.3, omega.6, omega.7, sigma.7 y lambda.7. Estos dispositivos hápticos son de cadena cinemática cerrada, sus actuadores son motores de CD y su transmisión es por cable, el más económico tiene un costo de 22,879.57 USD, hasta un costo de 89,143.98 USD del más especializado. Además, proporcionan una interfaz háptica flexible donde cada parámetro se puede personalizar o reprogramar para cumplir los requisitos de la aplicación; también cuenta con un controlador USB 2.0 independiente, lo que permite la integración de efectores finales personalizados con múltiples grados de libertad, donde Force Dimension proporciona soporte técnico avanzado para aplicaciones que requieran efectores finales personalizados. Force Dimension proporciona un SDK que permite el control total sobre cada elemento, como son las lecturas del encoder y los comandos del motor.

El delta.3 cuenta con un espacio de trabajo de $400 \times 400 \times 260$ mm, tiene una resolución < 0.02 mm y proporciona una fuerza de 20 N en traslación, puede orientar su placa base de 0 a 90 grados para adaptarse aún más a diversas aplicaciones y entornos.

Además, en cuanto a la serie de los robots omega, el omega.3 es el dispositivo más sencillo que ofrece Force Dimension, cuenta con un espacio de trabajo de $160 \times 160 \times 110$ mm, con una resolución < 0.01 mm y una fuerza de 12 N en traslación.

El omega.6 agrega un EF diferente al de los dispositivos omega.3 y delta.3, este efector es en forma de bolígrafo que permite proporcionar un rango de movimiento en la muñeca, su espacio de trabajo en traslación es de $160 \times 160 \times 110$ mm con una resolución de < 0.01 mm y en rotación de $240 \times 140 \times 320$ grados con resolución de 0.09 grados; también proporciona una fuerza de 12 N.

El omega.7 tiene su propio EF en forma de gripper que ofrece una fuerza de 8 N, un rango de movimiento de 25 mm para grasping y una resolución de 0.006 mm. Su espacio de trabajo en traslación es de $160 \times 160 \times 110$ mm, una resolución < 0.01 mm y una fuerza de 12 N; su espacio de trabajo en rotación es de $240 \times 140 \times 320$ grados con una resolución de 0.09 grados.

En 2010, Force Dimension presenta el sigma.7 que fue el primer dispositivo háptico comercial en incorporar fuerzas, torsiones y capacidades de agarre elevadas. También, puede integrarse en una configuración de estación de trabajo dual para operación bimanual, su espacio de trabajo es $190 \times 190 \times 130$ mm con resolución de 0.0015 mm y una fuerza de 20 N en traslación, en rotación cuenta con $235 \times 140 \times 200$ grados,

resolución de 0.013 grados y una fuerza de 400 mNm; el gripper ofrece una fuerza de 8 N, un rango de movimiento de 25 mm y resolución de 0.006 mm para grasping.

Force Dimension presenta el dispositivo lambda.7 en el año 2017, este dispositivo es diseñado para integrarlo en aplicaciones más complejas como las quirúrgicas o espaciales, integrando el robot en consolas de estaciones de trabajo bimanuales, este diseño ofrece un espacio de trabajo en traslación de $240 \times 240 \times 170$ mm, una resolución de 0.0015 mm y una fuerza de 20 N, en rotación su espacio de trabajo es de $180 \times 140 \times 290$ grados, resolución de 0.0067, 0.0067, 0.0135 grados y una fuerza de 200, 400, 100 mNm. El gripper permite un grasping de 15 grados, una resolución de 0.02 mm y proporciona 8 N de fuerza.

Delimitando el tema a los robots comerciales de Force Dimension, se encuentra el trabajo presentado por Oiwa et al. en [15], donde describen un estudio para evaluar la operatividad de un robot quirúrgico. El robot opera bajo el control maestro-esclavo implementado por la interfaz háptica del omega.7 y proporciona retroalimentación de fuerza. Sus resultados verificaron la efectividad del sistema del robot para implementación quirúrgica.

En [21], presentan un sistema robótico para intervención cerebrovascular, en donde su objetivo principal fue brindar experiencia y habilidades quirúrgicas al médico, así como evitar su exposición a rayos X, logrando controlar este sistema robótico mediante el dispositivo háptico omega.3 que opera como manipulador maestro cuando es controlado por el cirujano, y al mismo tiempo mediante retroalimentación háptica. Esta información es captada por el manipulador esclavo que está en contacto con el paciente. Los resultados del sistema robótico fueron exitosos debido a que durante la prueba del modelo vascular, el operador no estuvo expuesto a los rayos X ya que nunca entró en el lugar de la operación.

Mohammadi et al. en [39], proponen una técnica para la estimación de la fuerza de contacto de la yema del dedo al agarrar objetos deformables sin utilizar sensores de fuerza. En esta simulación utilizaron el dispositivo háptico omega.3 para representar las fuerzas de contacto del usuario con el objeto y medir con precisión las fuerzas ejercidas durante la interacción. Los resultados demostraron que con su técnica es posible estimar la fuerza de contacto de los dedos con objetos que conozcan su matriz de rigidez.

Chen et al. en [40], desarrollaron un sistema de intervención cardiovascular de operación maestro-esclavo con retroalimentación de fuerza integrando el dispositivo omega.3, el cual es operado por el médico como manipulador maestro, que a su vez, opera un robot manipulador esclavo en tiempo real. Sus resultados validaron la operación remota para el proceso de empujar, retirar y girar la guía que ingresa al lugar de la lesión del modelo cardiovascular bajo la guía de imágenes de la máquina de rayos X.

En [41] proponen un método de telemanipulación bilateral para comunicación entre dispositivos maestros y esclavos. Para este método integraron dos dispositivos omega.3 que realizan la retroalimentación háptica representando los movimientos y las fuerzas ejercidas por el dedo índice y pulgar del lado maestro y enviándolas a una mano de múltiples dedos como EF en el lado esclavo para lograr una mejor interacción con la tarea asignada.

Por otro lado, se reportan robots paralelos para interacción hombre-robot no comerciales como en [13] donde se propone rediseñar el modelo CAD propuesto por Avizzano et al. con una solución alternativa al modelado cinemático/dinámico del robot Delta de Clavel, basándose en un cálculo geométrico que permite reducir la carga computacional. Dicho modelo puede ser implementado en un microcontrolador sin introducir

cambios al diseño de control. Este robot tiene una base de 150 mm de radio y una altura de 340 mm, proporciona 3 grados de libertad (GDL) y un error medio cuadrado (RMS) en todo su espacio de trabajo de 15.24 nm.

Zhang et al. en [42] presentan el control de gestos naturales de un robot paralelo Delta mediante la implementación del dispositivo Leap Motion (este es un sistema de visión) que sirve para el seguimiento de los movimientos de las manos sin tener contacto con el usuario. Este método de control sin contacto se implementa al robot paralelo Delta donde definen ocho gestos manuales para interactuar con el robot. El dispositivo cuenta con 3 GDL, una resolución de hasta 1 mm, un torque nominal de 1.27 Nm y un espacio de trabajo de $400 \times 400 \times 150$ mm. En sus resultados demuestran que es posible manipular un robot Delta con precisión utilizando el dispositivo Leap Motion aun sin tener conocimiento previo en control de robots.

En [43] presentan una cinemática novedosa para un dispositivo háptico que cuenta con 4 GDL, propuesto para uso médico. La cinemática de este dispositivo se basa en la estructura del robot Delta para generar un movimiento en traslación y tres rotaciones alrededor de un centro remoto de movimiento que es la base inferior. Este dispositivo tiene un espacio de trabajo en traslación de 112 mm y en rotación de $\pm 180°$. Como resultado desarrollaron un prototipo impreso en 3D donde validaron la cinemática del dispositivo maestro.

Guanyang et al. en [44] presentan el diseño mecánico de dos dispositivos hápticos, un robot delta y un 3R paralelo rotacional, los cuales son implementados para teleoperación, control de posición y orientación de un robot esclavo para exploración espacial. El robot delta desarrollado está constituido por 3 GDL, proporciona un espacio de trabajo de $140 \times 140 \times 140$ mm, tiene una resolución de 0.1 mm, y brinda una fuerza máxima de 4 N. Realizaron experimentos utilizando un entorno virtual para validar la efectividad del enfoque propuesto.

Vulliez et al. en [16] presentan el diseño de un robot paralelo emparejado que ofrece 6 GDL para aplicaciones hápticas. Este dispositivo lo nombraron Delthaptic; está compuesto por dos robots Delta conectados a las dos extremidades del EF. Esta estructura paralela acoplada les permitió alcanzar un espacio de trabajo de $200 \times 200 \times 200$ mm mostrando una alta precisión de posicionamiento, que es un beneficio de los dispositivos paralelos como se vio en la Sección 2.1.2. El dispositivo brinda una fuerza de 20 N y una precisión ≤ 1 mm; en rotación un área de trabajo de ± 90 ± 90 ± 90 grados, una fuerza de 0.5 Nm y precisión ≤ 2 grados. Fabricaron un prototipo de laboratorio que utilizaron para validar el funcionamiento de la estructura mecánica de su diseño, logrando realizar pruebas donde integraron el dispositivo para teleoperar robots industriales en tareas remotas con retroalimentación de fuerza.

Somló et al. en [45] presentaron el diseño de la parte mecánica de un robot paralelo al cual nombraron "PHANTOM", su diseño se basó principalmente en bases triangulares y brazos unidos en forma de paralelogramo. En esta investigación se basaron solo en la parte del diseño mecánico, dejándolo como prototipo para una futura fabricación y despreciando el sistema de control.

En la Tabla 2.1 se resumen las características principales de los Robots tipo Delta comerciales, considerando que todos estos dispositivos utilizan transmisión por cable y como actuadores 3 motores DC. Además, sus principales implementaciones se encuentran en la háptica, teleoperación y ambientes virtuales. También, en la Tabla 2.2 se presentan las características de los dispositivos que fueron diseñados por diferentes investigadores, resaltando que ninguno de estos dispositivos considera la reconfiguración

para que pueda utilizarse tanto en trabajos horizontales como verticales y todos utilizan la transmisión por cable. Por lo tanto, a partir de estas características se definieron los requisitos de diseño para el desarrollo del robot tipo Delta que se propone en este proyecto de investigación.

Cabe señalar que de todos los dispositivos hápticos comerciales revisados en el estado del arte, solo se encontró uno que considera la configuración tanto para trabajos horizontales como para trabajos verticales, motivo por el cual se propone un dispositivo que pueda ser reconfigurado mecánicamente para realizar tanto trabajos verticales como trabajos horizontales. De acuerdo a los dispositivos comerciales y no comerciales analizados, se propone que pueda retroalimentar una fuerza de 15 N en su EF, suficiente para ser implementado en el promedio de los proyectos descritos en el estado del arte. Además, en la mayoría de las investigaciones donde se utiliza un dispositivo comercial, no requirieron de modificar el EF. Sin embargo, algunos investigadores que requerían de un EF diferente al de fabrica, desarrollaron su propio dispositivo háptico y su EF para que este se adaptara a la aplicación requerida; en estos casos los autores hacen mención del costo adicional que representaría si se utilizara un dispositivo comercial, debido a que la misma empresa se encargaría de realizar el diseño del EF para que posteriormente lo agreguen a la cinemática del dispositivo háptico.

Tabla 2.1: Características principales de dispositivos hápticos comerciales.

Robot comercial	**Costo (USD)**	**Reconfigurable mecánicamente**	**Resolución**	**Fuerza**	**GDL**	**Espacio de trabajo**
Novint Falcon [19, 35, 36] 2007	499	No	1 mm - efector final	8.8 N	3	101.6 x 101.6 x 101.6 mm
Delta.3 [7] 2009	34,858.97	Sí	<0.02 mm	20 N	3	Traslación ⌀ 400 x 260 mm
Omega.3 [7] 2003	22,879.57	No	<0.01 mm	12 N	3	Traslación ⌀ 160 x 110 mm
Omega.6 [7] 2004	29,462.84	No	Traslación <0.01 mm Rotación 0.09 grados	12 N	3	Traslación ⌀ 160 x 110 mm Rotación 240 x 140 x 320 grados
Omega.7 [7] 2007	37,125.34	No	Traslación <0.01 mm Rotación 0.09 grados Grasping 0.006 mm	Traslación 12 N Grasping ± 8.0 N	3	Traslación ⌀ 160 x 110 mm Rotación 240 x 140 x 180 grados Grasping 25 mm
Sigma.7 [7] 2010	84,827.08	No	Traslación 0.0015 mm Rotación 0.013 grados Grasping 0.006 mm	Traslación 20 N Rotación 400 mNm Grasping ± 8.0 N	7	Traslación ⌀ 190 x 130 mm Rotación 235 x 140 x 200 grados Grasping 25 mm
Lambda.7 [7] 2017	89,143.98	No	Traslación 0.0015 mm Rotación 0.0067, 0.0067 0.0135 grados, Grasping 0.02 mm	Traslación 20 N Rotación 200, 400, 100 mNm, Grasping 8 N	7	Traslación ⌀ 240 x 170 mm Rotación 180 x 140 x 290 grados Grasping 15 mm

Tabla 2.2: Características principales de dispositivos hápticos diseñados por investigadores.

Robots diseñados por investigadores	**Resolución**	**Fuerza**	**GDL**	**Actuadores**	**Espacio de trabajo**	**Tipo de implementación**
Avizzano et al. [13] 2015	-	-	3	3 motores CD	-	Telemedicina, ultrasonido.
Zhang et al. [42] 2019	1 mm	-	3	3 servomotor CA	400 x 400 x 150 mm	Teleoperación.
Saafi et al. [43] 2017	-	-	4	3 motores CD	Traslación de 112 mm Rotación de ±180 grados.	Teleoperación, médico.
Guanyang et al. [44] 2019	0.1 mm	4 N	3	3 motores CD	140 x 140 x 140 mm	Teleoperación, exploración espacial.
Vulliez et al. [16] 2018	Traslación $\leq$ 1 mm Rotación $\leq$ 2 grados	20 N	6	3 motores CD	Traslación 200 $\times$ 200 $\times$ 200 mm Rotación $\pm$ 90 $\pm$ 90 $\pm$ 90 grados	Háptica.
Somló et al. [45] 2018	-	-	3	Manual	Traslación 530.12 x 530.12 x 273.72 mm	Prototipo.

Capítulo 3

Desarrollo de la metodología - Etapa 1

En este capítulo se presenta la etapa 1 del desarrollo de la metodología propuesta donde se describen los requisitos de diseño y los criterios de selección. Así también, se hace mención del diseño de componentes en modelo CAD, las propuestas de diseño y la casa de la calidad como método de evaluación de cada propuesta de diseño. De igual forma, se muestra el análisis de elemento finito realizado para obtener las cargas estáticas a los componentes del diseño seleccionado del robot tipo Delta, donde se evalúa su funcionamiento y la resistencia del material seleccionado.

3.1. Requerimientos de diseño

Mediante un estudio de los dispositivos comerciales en la actualidad y de los fabricados por investigadores que se presentan en el estado del arte, se considera la ingeniería inversa como base para el diseño del dispositivo háptico. La ingeniería inversa a grandes rasgos se basa en comprender el funcionamiento interno de los dispositivos actuales, analizando sus componentes con el fin de obtener información sobre su diseño o cualquier característica relevante y así proponer un diseño diferente que permita mejorar las características seleccionadas de los dispositivos actuales.

Por lo tanto, de acuerdo a los dispositivos actuales analizados se seleccionaron los siguientes requerimientos de diseño para el desarrollo de este robot tipo Delta:

1. **Reconfigurable mecánicamente (trabajos verticales y horizontales):** El dispositivo se propone para que permita la reconfiguración mecánica para realizar trabajos tanto horizontales como verticales según la aplicación que requiera el usuario.
2. **Fuerza:** Se propone que el dispositivo soporte una carga de 15 N en el efector final para representar colisiones con objetos rígidos en ambientes virtuales. Esta carga se determinó de acuerdo a los trabajos revisadas en el estado del arte donde se utilizaron robots tipo Delta y en promedio la fuerza proporcionada por el dispositivo era de 15 N, determinando que es una carga suficiente para ser implementado en múltiples aplicaciones, ya que si se aumenta la fuerza en el dispositivo se requeriría de motores más grandes que comprometerían la portabilidad del dispositivo. [15, 17–21].

3. **Espacio de trabajo:** Se determinó un espacio de trabajo para el robot tipo Delta de $200 \times 200 \times 150$ mm, este se eligió de acuerdo al promedio de los dispositivos comerciales y no comerciales que son implementados en las investigaciones que se mencionan en el estado del arte [6, 7, 13, 15, 16, 36, 37, 39, 40, 44, 45].
4. **Diseñar un dispositivo portable:** El robot tipo Delta debe poder ser trasladado por el usuario de un lugar a otro.
5. **Rigidez en los componentes del dispositivo:** Los componentes del robot tipo Delta deberán soportar una carga de 15 N sin presentar alguna deformación o fractura. Por lo tanto, se eligieron para las pruebas los polímeros PLA y ABS para impresión en 3D, aluminio para las barras paralelas y pernos para unión de capstan (conocido también como cabrestante), articulaciones y base móvil.
6. **Disminuir juego mecánico en articulaciones:** Se diseñaron diferentes articulaciones que permiten disminuir el juego mecánico. Además, se propone utilizar transmisión por cable, que en comparación de una transmisión por engranes, esta ayuda a disminuir el juego mecánico en el dispositivo.
7. **Adaptable a diferentes tipos de efectores finales:** Este robot tipo Delta se diseñará de tal forma que facilite el ensamble para diferentes efectores finales, según la aplicación que se trabaje.
8. **Fácil ensamble:** Se considera la cantidad total de componentes que conformen el robot tipo Delta, para este proyecto son un total de 230 piezas que pueden ensamblarse con herramienta básica (desarmadores, pinzas) que permiten que el ensamble final pueda ser realizado por personas con un nivel de estudio de licenciatura en un tiempo máximo de 4 horas.
9. **Fácil mantenimiento:** El mantenimiento del robot se propone que sea correctivo debido a la disponibilidad de materiales para venta al público (PLA, barras paralelas de aluminio, rodamientos, seguros, resortes y barra de latón), estos materiales permiten que el usuario adquiera los elementos con distribuidores locales para realizar el mantenimiento en el que solo se requiera sustituir el componente dañado.
10. **Accesible para diferentes grupos de investigadores:** La materia prima y manufactura de los componentes del robot tipo Delta, permiten que el dispositivo sea accesible para su fabricación y mantenimiento en la mayoría de grupos de investigadores y universidades.

Por otra parte, se proponen los siguientes materiales para la fabricación de componentes del robot tipo Delta:

1. Rollo de PLA para impresión 3D de los componentes.
2. Hoja de MDF (Medium Density Fibreboard, en inglés), base para trabajos verticales de 9 mm de espesor.
3. Hoja de triplay (base general para trabajos horizontales) de 9 mm de espesor.
4. Cable de acero de 0.9 mm de diámetro.
5. Barra de Aluminio (para fabricación de barras paralelas y pernos).
6. Barra de latón (para prensado de cable de acero).

Respecto a los actuadores, se escogieron motores de corriente continua teniendo en cuenta que el robot tipo Delta pueda soportar una carga de 15 N en su EF. Considerando las dimensiones para preservar la portabilidad del dispositivo y que este incluyera una caja de engranajes para incrementar el torque de salida. Por lo tanto, se seleccionó

el motor de la marca Pololu, #4861, Gearbox 4.4:1, 25D x 63L mm, MP 12 V con encoder.

3.2. Diseño de componentes en modelo CAD

Para iniciar el proceso de diseño de los componentes en modelo CAD, definimos como proceso de diseño al desarrollo intensivo por el cual se obtiene la propuesta de un producto especifico [10]. En la Figura 3.1 se detalla el proceso en que se fundamenta el modelo descriptivo para la formulación de un diseño.

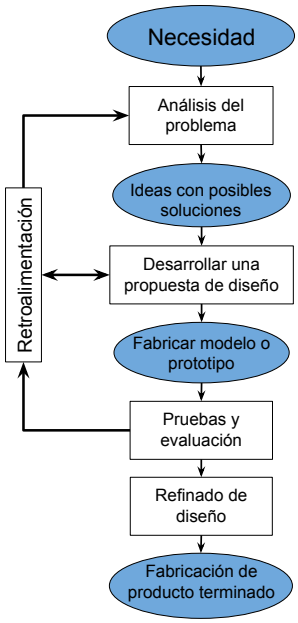


Figura 3.1: Modelo descriptivo para el proceso de diseño, editada de Howard et al [10].

A continuación, se describe el modelo de la figura anterior:

1. **Necesidad:** Surge a través de un problema que requiere ser resuelto.
2. **Análisis del problema:** En esta etapa se determinan los requisitos y las limitaciones, para quien se diseñará el producto.
3. **Ideas con posibles soluciones:** Antes de comenzar con el diseño se plantea una serie de posibles soluciones, estas no deben de estar limitadas.
4. **Desarrollar una propuesta de diseño:** Una vez claras las posibles soluciones, estas se plantean en una propuesta de diseño. Esta etapa puede continuar a lo largo del proyecto y repetirse las veces que sean necesarias hasta encontrar un diseño funcional.
5. **Fabricar modelo o prototipo:** En esta etapa se utiliza la propuesta de diseño para construir un prototipo.
6. **Pruebas y evaluación:** Cada prototipo se tiene que evaluar mediante pruebas de funcionamiento, estas pruebas permiten validar dónde se necesitan mejoras en el diseño.
7. **Refinado de diseño:** Una vez que se han completado las pruebas, puede evaluarse el diseño para mejorarlo. Esta etapa puede repetirse varias veces a medida que se crean y evalúan más prototipos.

8. **Fabricación de producto terminado:** La etapa final comienza una vez que se hayan completado las mejoras y validado su funcionamiento, se puede comenzar la fabricación del producto con los estándares de calidad requeridos (dimensiones, tolerancias, textura, etc.).

Por lo tanto, con base a este modelo descriptivo del proceso de diseño y considerando los requerimientos de diseño mencionados anteriormente, se realizó una primera propuesta de diseño para el robot tipo Delta (Figura 3.2), listando los componentes de esta primera propuesta en la Tabla 3.1.

3.2.1. Primera propuesta

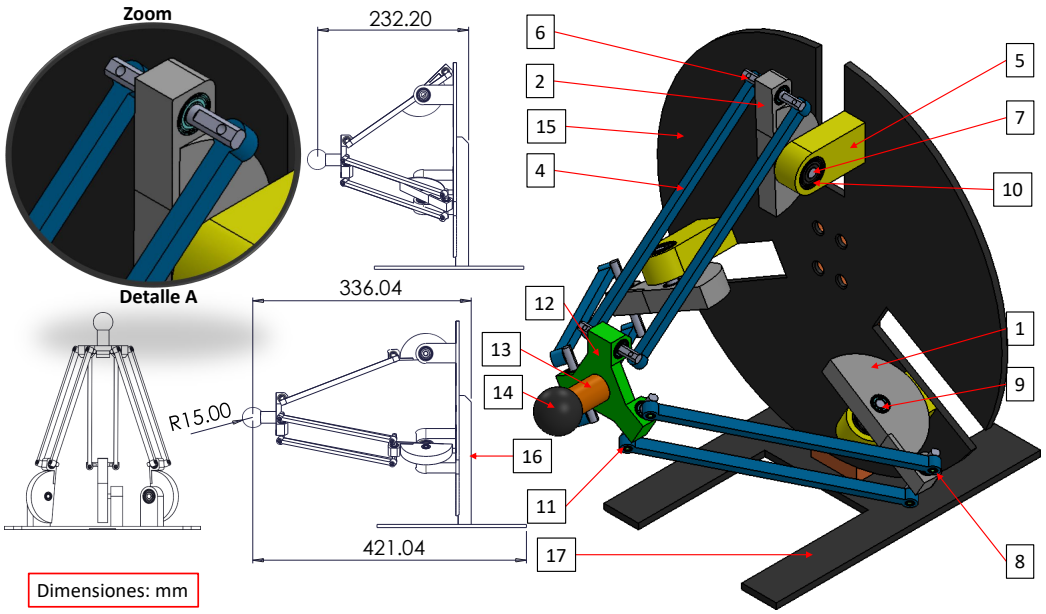


Figura 3.2: Diseño de la primera propuesta.

Este diseño se evaluó con base en los requerimientos de diseño del robot tipo Delta y como se menciona en el modelo del proceso de diseño descriptivo (Figura 3.1), al desarrollar una propuesta de diseño debe de considerarse una serie de posibles soluciones para el problema y el desarrollo de una propuesta puede repetirse las veces que sea necesario hasta encontrar el diseño que cumpla con los requerimientos de diseño. Por consiguiente, al desarrollar la primera propuesta se evaluaron algunos de sus componentes que influyen en la mecánica del diseño, considerando los siguientes criterios para realizar modificaciones en el diseño de la primera propuesta.

1. **El capstan:** En el diseño de esta primera propuesta el capstan esta compuesto por dos piezas (medio círculo y extensión para unión con barras paralelas), de modo que se plantea modificar el diseño para que sea de una sola pieza y eliminar los componentes para ensamble.
2. **La base del capstan:** Se propone un diseño donde se soporte al capstan por ambos lados debido a que ofrece mayor estabilidad.
3. **La base circular:** Es posible reducir el diámetro de la base sin afectar la funcionalidad y disminuyendo el tiempo de fabricación en la cortadora láser.

Tabla 3.1: Componentes de la primera propuesta.

No.	Descripción de componente	Cantidad
1	Capstan de 80 mm de diámetro	3
2	Brazo complemento para dar mayor alcance a capstan	3
3	Polea de 19.8 mm de diámetro	3
4	Barras paralelas cuadradas	6
5	Base soporte para capstan y motor	3
6	Eje para ensamble de barras paralelas con capstan y unión con base móvil	6
7	Perno para ensamble de capstan y base soporte de capstan	3
8	Perno para ensamble de barras paralelas y eje	12
9	Rodamientos 628/6-2z - diam. ext=13 mm, diam. int=6 mm	18
10	Rodamientos 627-2z - diam. ext=22 mm, diam. int=7 mm	6
11	Rodamientos W637/3-2z - diam. ext=6 mm, diam. int=3 mm	24
12	Base móvil del robot	1
13	Soporte para efector final	1
14	Efector final	1
15	Base circular para ensamble de robot (para realizar trabajos verticales)	1
16	Soporte para ensamble de base general y base para trabajos verticales	1
17	Base general del robot (para realizar trabajos horizontales)	1
18	Motor Pololu, 4.4:1, 25Dx63L mm, 12V con Encoder	3
19	Resorte de tensión de 2 cm de largo (extremo a extremo)	3
20	Tornillo para mecanismo de tensión de cable	3
	Total de piezas	**102**

4. **El eje para unión entre las barras paralelas y el capstan:** El ensamble entre el eje del capstan y las barras paralelas puede mejorarse diseñando una articulación que sea concéntrica en ambos componentes, evitando así un desplazamiento que requiera un método de análisis diferente en la cinemática del robot, como seria el caso para esta propuesta por la forma de ensamble de los componentes (ver Detalle A de Figura 3.2).

De acuerdo con los criterios mencionados y los cambios propuestos, se generó una segunda propuesta que tiene como objetivo mejorar los componentes evaluados, mejorando en este diseño el soporte del capstan y el tipo de articulación que mejora ensamble de las barras paralelas (Figura 3.3). En la Tabla 3.2 se describen los componentes presentes en la segunda propuesta.

3.2.2. Segunda propuesta

Se evaluó esta propuesta de diseño con base en los requerimientos de diseño, se consideraron algunos de sus componentes principales para su análisis (el capstan, las bases que soportan el capstan, la articulación y las barras paralelas) y se determinó que pueden ser modificados para mejorar su funcionamiento o forma de ensamble. Los criterios que se consideraron para modificar los componentes de la segunda propuesta son los siguientes:

1. **El capstan:** Se considero diseñar un capstan diferente a las propuestas anteriores, tomando el medio círculo de la primera propuesta y que fuera de una sola pieza de la segunda propuesta. Además, se propone reducir el diámetro del capstan y representar la relación de transmisión con la polea.

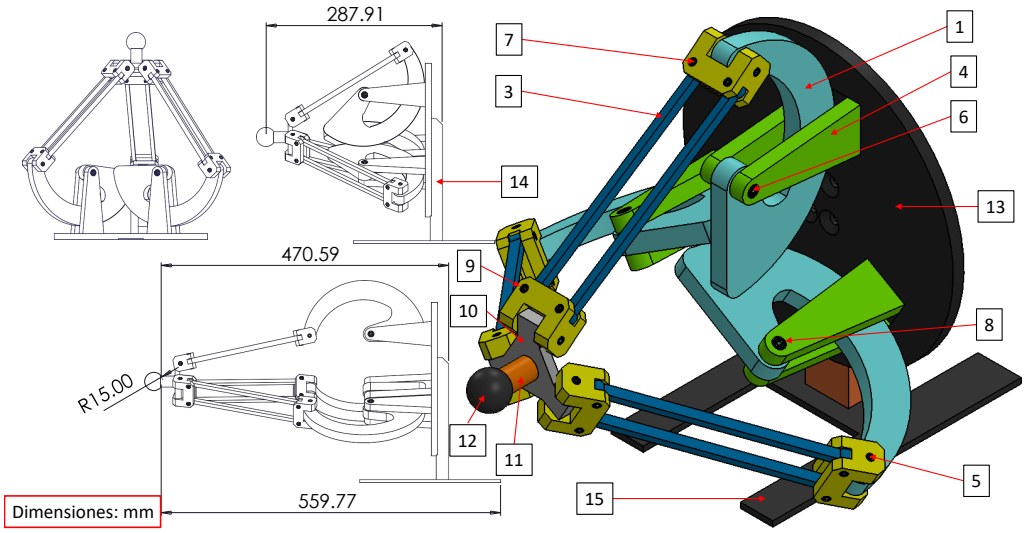


Figura 3.3: Diseño de la segunda propuesta.

Tabla 3.2: Componentes de la segunda propuesta.

No.	Descripción de componente	Cantidad
1	Capstan de 200 mm de diámetro	3
2	Polea de 19.8 mm de diámetro	3
3	Barras paralelas cuadradas	6
4	Base soporte para capstan y motor	6
5	Eje para ensamble de barras paralelas con capstan y unión con base móvil	6
6	Perno para ensamble de capstan y base soporte de capstan	3
7	Perno para ensamble de barras paralelas y eje	12
8	Rodamientos 623-2z - diam. ext=10 mm, diam. int=3 mm	12
9	Rodamientos W637/3-2z - diam. ext=6 mm, diam. int=3 mm	48
10	Base móvil del robot	1
11	Soporte para efector final	1
12	Efector final	1
13	Base circular para ensamble de robot (trabajos verticales)	1
14	Soporte para ensamble de base general y base para trabajos verticales	1
15	Base general del robot, (trabajos horizontales)	1
16	Motor Pololu, 4.4:1, 25Dx63L mm, 12V con Encoder	3
17	Resorte de tensión de 2 cm de largo (extremo a extremo)	3
18	Tornillo para mecanismo de tensión de cable	3
	Total de piezas	**114**

2. **Base soporte del capstan:** Las 2 bases para soportar el capstan proporcionan una buena estabilidad en el capstan, pero dificultan el ensamble con la polea y el cable de acero. Además, aumenta el consumo de material PLA al ser 2 piezas, por lo que nuevamente se probará con una pieza pero con soportes adicionales.
3. **Barras paralelas:** Si se consideran barras paralelas circulares de aluminio, es posible adquirirlas en medida estándar y eliminar el mecanizado.
4. **La articulación:** El diseño de la articulación para unir el capstan con las barras paralelas puede dificultar el ensamble y producir juego mecánico.
5. **El número total de piezas:** Se incrementa el número de componentes que se requiere fabricar o adquirir para esta propuesta, se analizará la posibilidad de

reducir la cantidad de componentes.

Con base en los criterios mencionados, se propone una tercera propuesta que mejore las características de los diseños anteriores (ver Figura 3.4), considerando una reducción en el número de componentes para fabricación, cambio en la geometría de las barras paralelas, diferentes las articulaciones que faciliten el ensamble, modificación al capstan y su base. Los componentes que conforman la tercera propuesta se mencionan en la Tabla 3.3.

3.2.3. Tercera propuesta

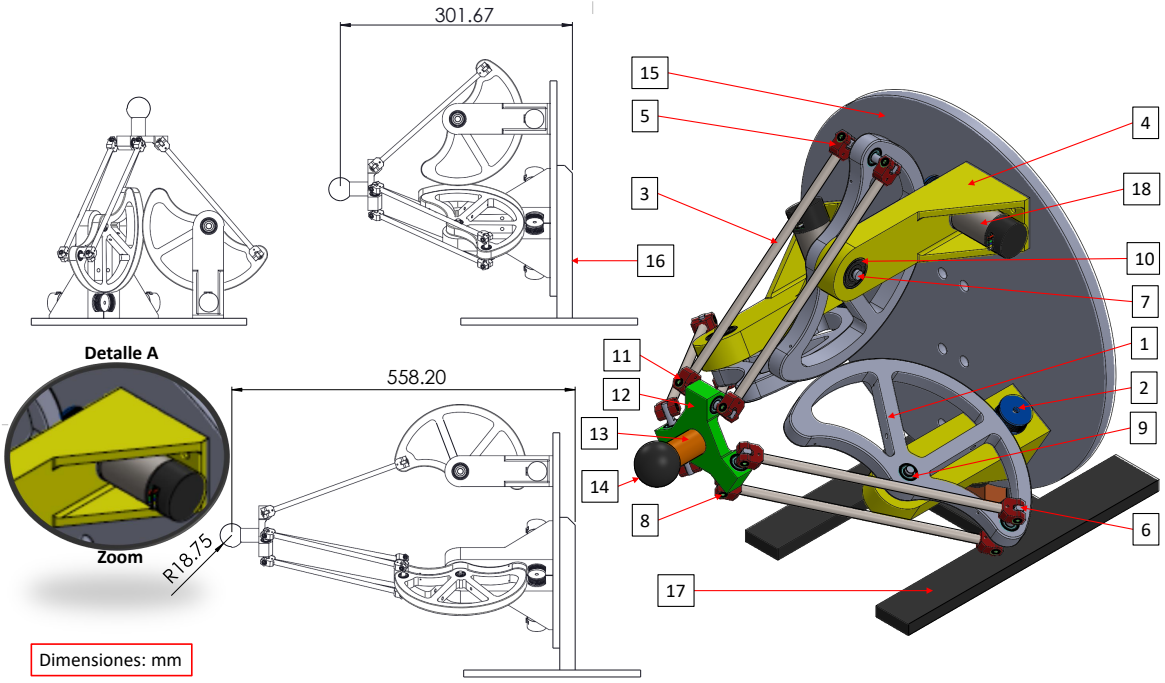


Figura 3.4: Diseño de la tercera propuesta.

A continuación se describen las modificaciones realizadas a los componentes y las características principales de esta tercera propuesta:

1. **Articulación:** Se rediseña este componente para ensamblar las barras paralelas con el capstan, disminuyendo la complejidad de su fabricación (tamaño, tiempo) y considerando que sea concéntrica al eje.
2. **Base soporte de capstan:** Se agregan refuerzos tipo nervio a la base que soporta el capstan para evitar deformaciones cuando se aplique una carga al efector final (Ver Detalle A de Figura 3.4).
3. **El capstan:** Se modificó el diseño y el diámetro del capstan sin afectar su funcionamiento ni resistencia, logrando reducir el material requerido para su fabricación y el tiempo impresión en 3D.
4. **Polea:** Se agregan muescas a la polea para dirigir el cable de acero y evitar que se enrede, se consideraron 3 vueltas del cable a la polea.
5. **Número de componentes:** En comparación con la segunda propuesta, esta propuesta reduce la cantidad total de piezas, que disminuye tiempos de fabricación, material y costo.

Tabla 3.3: Componentes de la tercera propuesta.

No.	Descripción del componente	Cantidad
1	Capstan de 172 mm de diámetro	3
2	Polea de 23 mm de diámetro	3
3	Barras paralelas circulares	6
4	Base soporte para capstan y motor	3
5	Pieza para ensamble de barras paralelas con capstan y unión con base móvil	12
6	Eje para ensamble de barras paralelas con capstan y unión con base móvil	6
7	Perno para ensamble de capstan y base soporte de capstan	3
8	Perno para ensamble de barras paralelas y elemento de unión	12
9	Rodamientos 628/6-2z - diam. ext = 13 mm, diam. int = 6 mm	18
10	Rodamientos 627-2z - diam. ext = 22 mm, diam. int = 7 mm	6
11	Rodamientos W637/3-2z - diam. ext = 6 mm, diam. int = 3 mm	24
12	Base móvil del robot	1
13	Soporte para efector final	1
14	Efector final	1
15	Base circular para ensamble de robot (trabajos verticales)	1
16	Soporte para ensamble de base general y base para trabajos verticales	1
17	Base general del robot	1
18	Motor Pololu, 4.4:1, 25Dx63L mm 12V con Encoder	3
19	Resorte de tensión de 2 cm de largo (extremo a extremo)	3
20	Tornillo para mecanismo de tensión de cable	3
	Total de piezas	**111**

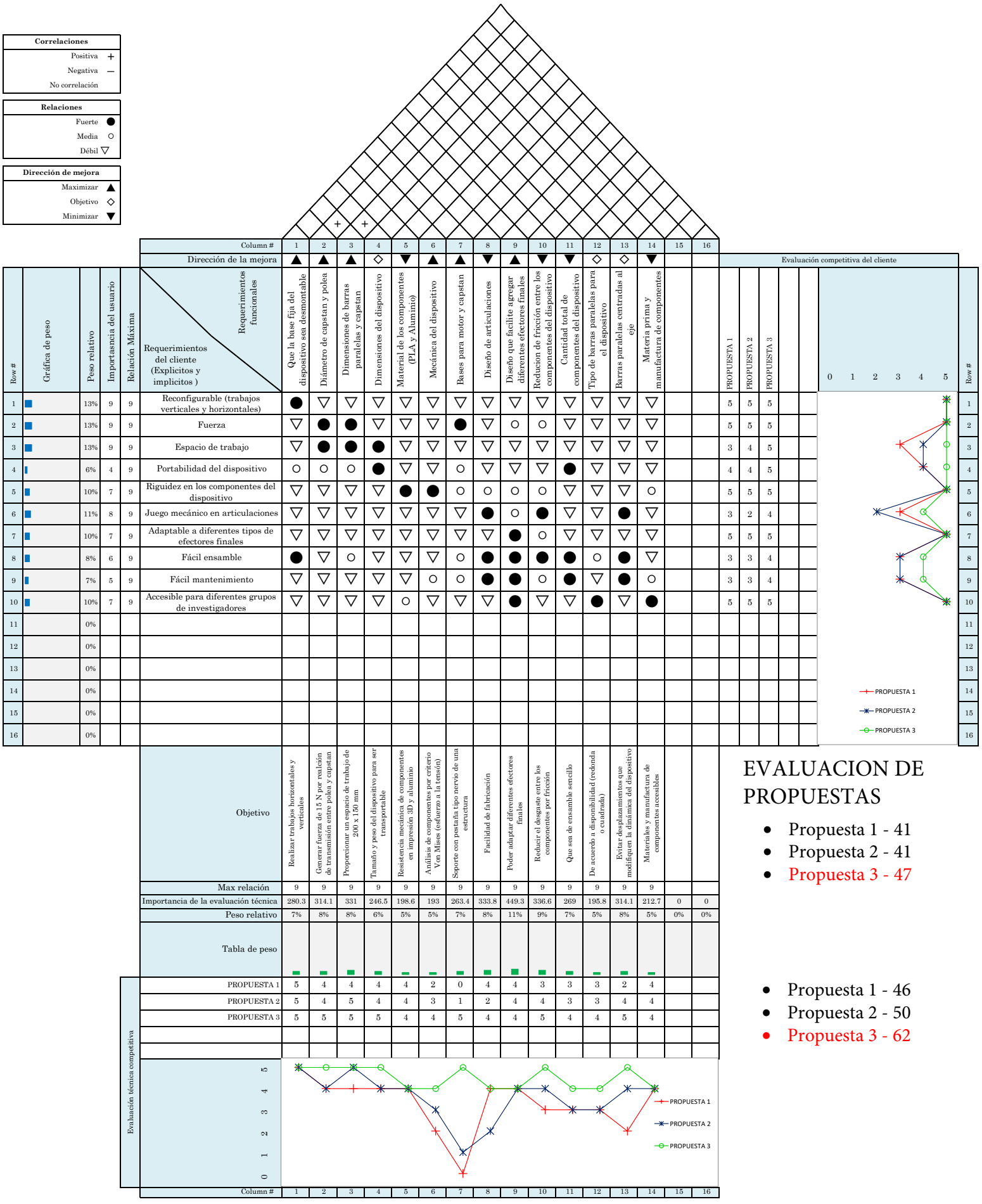
6. **Barras paralelas:** Al considerar que las barras paralelas sean circulares y en medida estándar, se facilita su adquisición y selección de material (aluminio, acero inoxidable, etc.).
7. El dispositivo puede reconfigurarse mecánicamente para ser utilizado tanto para trabajos horizontales como para verticales.

Considerando las tres propuestas presentadas anteriormente, se realizó la evaluación de estas para seleccionar el diseño que mejor cumple con los requerimientos de diseño. La evaluación de las propuestas de diseño se realizó mediante la casa de la calidad, que como se mencionó en el Capítulo 2.1.7, es una herramienta que permite evaluar las características de cada propuesta de diseño con base en los requerimientos establecidos para el robot tipo Delta y mediante una serie de matrices internas que componen la casa de la calidad es posible seleccionar el diseño que cumple de mejor forma con los requerimientos de diseño (ver Figura 3.5).

3.2.4. Casa de la Calidad

En la Figura 3.5 se presenta la casa de la calidad que contiene los datos de cada una de las propuestas de diseño para el robot tipo Delta. En la parte izquierda se presentan los requerimientos de diseño; las técnicas para satisfacer dichos requerimientos se colocan en la parte superior; en la parte central una matriz que relaciona los requerimientos de diseño contra las técnicas para satisfacerlos; en la parte inferior se coloca el objetivo o prioridad de los requerimientos. La evaluación para cada una de las propuestas de diseño se divide en dos partes, en el lado derecho se muestra la evaluación del cumplimiento de los requerimientos de diseño; y en la parte inferior se evalúa la dificultad que representa conseguir el objetivo de cada requerimiento de diseño.

QFD: Despliegue de la función calidad
PROYECTO: Dispositivo experimental de robot tipo Delta mecánicamente reconfigurable.



EVALUACION DE PROPUESTAS

- Propuesta 1 - 41
- Propuesta 2 - 41
- Propuesta 3 - 47

- Propuesta 1 - 46
- Propuesta 2 - 50
- Propuesta 3 - 62

Figura 3.5: Casa de la calidad.

Los resultados en la evaluación de las propuestas de diseño para resolver los requerimientos del robot tipo Delta se observan en la gráfica derecha de la Figura 3.5, siendo en este apartado los 47 puntos correspondientes a la propuesta 3 los que mejor cumplen con los requerimientos, por encima de la propuesta 1 y 2.

Además, en cuanto a la evaluación sobre la dificultad que representa lograr el objetivo de los requerimientos de diseño, se muestra la gráfica inferior con los resultados, donde la propuesta 3 obtiene 62 puntos, siendo superiores de nueva cuenta a los que obtiene la propuesta 1 y 2 respectivamente.

Por lo tanto, a partir de este punto se considera a la propuesta 3 como el diseño seleccionado para continuar con su desarrollo, análisis y fabricación de sus componentes para validación y pruebas físicas de funcionamiento.

Adicionalmente, para un mejor control de los componentes de diseño se le asignó nombre y número de parte a cada elemento que integra la propuesta seleccionada. Por lo tanto, la Figura 3.6 presenta el diseño final del robot tipo Delta, considerando los componentes que se agregaron o modificaron durante el desarrollo del proyecto y se describen en las siguientes secciones.

No.	Número de parte	Descripción	Cant.
1	RD.01-110	Base para trabajos horizontales	1
2	RD.01-115	Barra soporte para ensamble de base general y base para trabajos verticales	1
3	RD.01-120	Base circular para trabajos verticales	1
4	RD.01-125	Base Soporte para capstan y motor	3
5		Tuerca de seguridad para tornillo M4	12
6		Motor 25D mm, 4.4:1, 12 V	3
7		encoder_400PPR	3
8	Rodamiento SKF 627-2z	Rodamientos diam. ext. = 22 mm, diam. int. = 7 mm	6
9		Seguro para perno de 6 mm de diámetro	3
10	RD.01-130	Perno para ensamble de capstan y base de motor	3
11		Tornillo M8 x 40, Cabeza plana	6
12	RD.01-135	Capstan	3
13	RD.01-140	Polea	3
14		Tuerca para tornillo M4	3
15		Tornillo milimetrico M4	3
16		Resorte de tensión	3
17	Rodamiento SKF 628/2z	Rodamientos diam. ext = 13 mm, diam. int. = 6 mm	18
18		Cable de acero de 0.9 mm de diámetro	3
19	RD.01-145	Barras paralelas circulares	6
20	RD.01-150	Perno para ensamble de barras paralelas con articulación	12
21	RD.01-155	Articulación para ensamble en barras paralelas	12
22	RD.01-160	Eje para ensamble de paralelas con capstan y base móvil	6
23	RD.01-165	Base móviil	1
24	RD.01-170	Soporte para efector final	1
25	RD.01-175	Efector final	1
26		Seguro para perno de 5 mm de diámetro	15
27	Rodamiento ZEN SMR 63-2z	Rodamientos diam. ext. = 6 mm, diam. int. = 3 mm	24
28		Seguro para perno de 2 mm de diámetro	12
29	RD.01-180	Cople para encoder	3
30	RD.01-185	Base para encoder	3
31		Tornillo M5 x 25, Cabeza plana	4
32		Tuerca de seguridad para tornillo M5	4
33		Tuerca de seguridad para tornillo M8	6
34		Tornillo M4 x 20, Cabeza plana	12

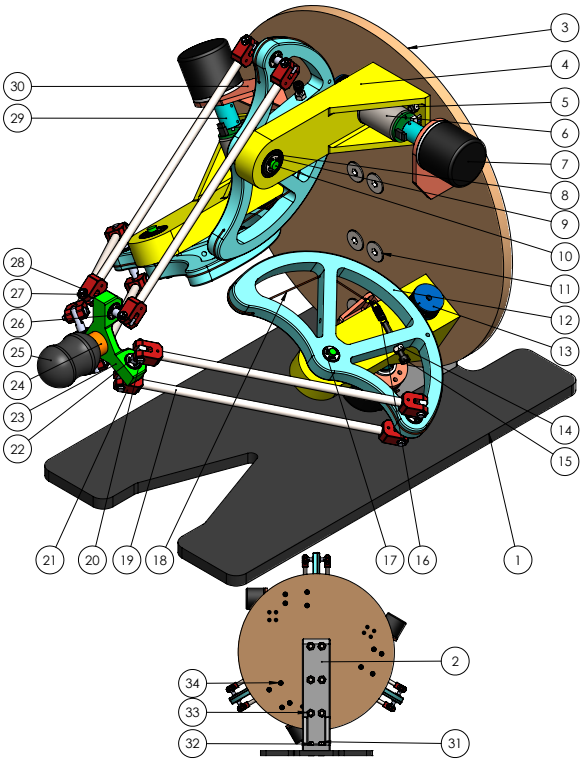


Figura 3.6: Nombre y No. de parte de componentes.

3.3. Análisis de elemento finito a componentes del robot

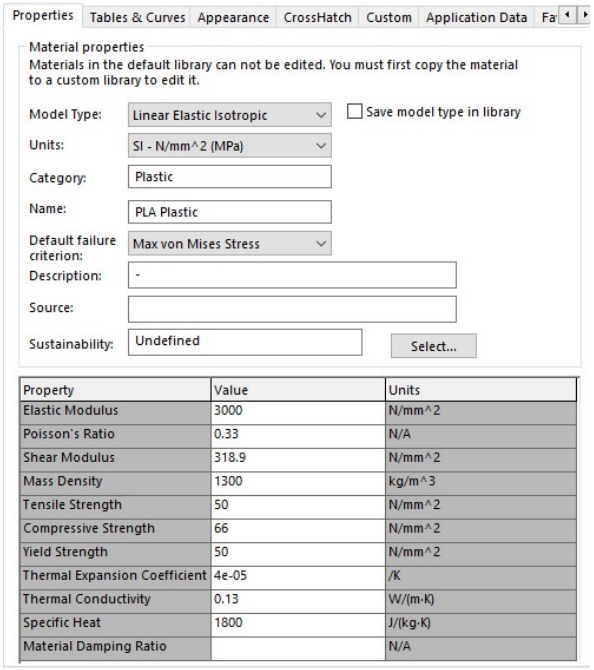
En esta sección, se llevó a cabo el análisis de cargas estáticas utilizando el método de elemento finito para definir si el diseño de los componentes y el material seleccionado

pueden soportar una carga de 15 N en el EF del robot tipo Delta.

Sin embargo, el análisis de elemento finito solo se realizó a dos componentes que pudieran ser afectados al soportar la carga de 15 N y que pudieran sufrir deformaciones o rupturas cuando se retroalimente una carga en el EF por colisiones con objetos virtuales.

- RD.01-135 - Capstan (Figura 3.8).
- RD.01-130 - Perno para ensamble con capstan y la base de motor (Figura 3.11).

Por otro lado, se eligió el polímero PLA como el material para este análisis. No obstante, debido a que el programa Solidworks® no cuenta con las propiedades del PLA, se ingresaron manualmente en su librería de materiales (ver Figura 3.7), considerando que estos valores son para simulación debido a que presentan pequeñas variaciones entre las diferentes fuentes consultadas.



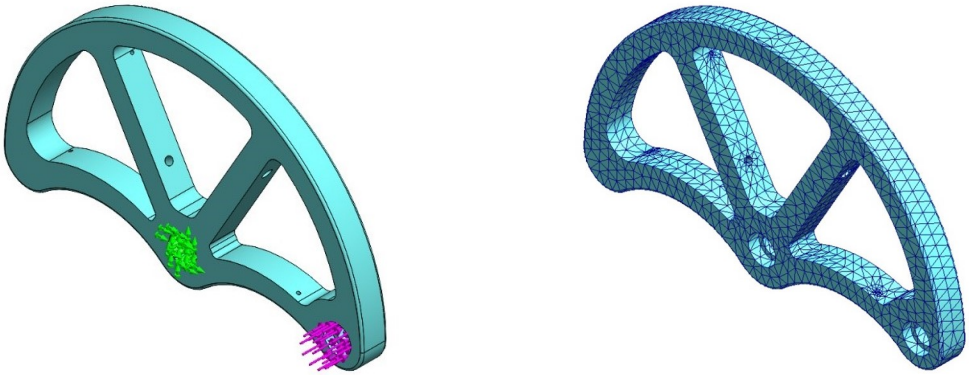
Property	Value	Units
Elastic Modulus	3000	N/mm^2
Poisson's Ratio	0.33	N/A
Shear Modulus	318.9	N/mm^2
Mass Density	1300	kg/m^3
Tensile Strength	50	N/mm^2
Compressive Strength	66	N/mm^2
Yield Strength	50	N/mm^2
Thermal Expansion Coefficient	4e-05	/K
Thermal Conductivity	0.13	W/(m·K)
Specific Heat	1800	J/(kg·K)
Material Damping Ratio		N/A

Figura 3.7: Propiedades del material PLA, revisado en Octubre de 2022 [11, 12].

3.3.1. Análisis de elemento finito del componente RD.01.135 - Capstan

Para el análisis de este componente se consideraron las siguientes restricciones (ver Figura 3.8):

1. Se colocó geometría fija o de sujeción en el área central, donde estarán los rodamientos y se ensamblará con el perno para unión a la base (Zona verde de la Figura 3.8a).
2. Se consideró para el análisis una carga de 15 N, que es la carga máxima que el robot tipo Delta soportará en su EF (Zona púrpura de la Figura 3.8a).
3. Se creó un mallado estándar para el análisis del componente, con distancia entre nodos de 4.151 mm con una tolerancia entre ellos de ± 0.201 mm, dando un total de 18316 nodos (Figura 3.8b).



(a) (b)

Figura 3.8: Restricciones para análisis de elemento finito a capstan. (a) Sujeción (flechas color verde) y carga en capstan (flechas color púrpura). (b) Mallado de capstan.

Los resultados del análisis de elemento finito muestran la tensión del componente y está representada en una escala de deformación de 1:1, este resultado viene dado bajo el criterio de Von Mises, donde de acuerdo a las características del PLA su esfuerzo máximo a la tensión (antes de deformación) = 50 MPa.

Por lo tanto, para interpretar el resultado de la simulación, se observa que el capstan está representado por colores, estos colores representan diferentes valores de la carga que estaría actuando en dicha zona, siendo la zona de color rojo la más crítica y que requiere de mayor atención.

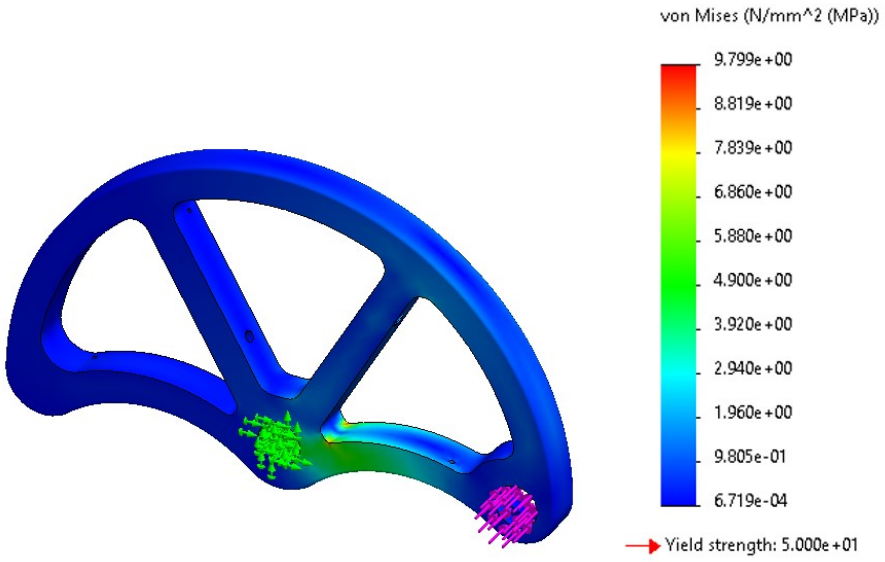


Figura 3.9: Análisis de esfuerzos - capstan.

Por consiguiente, en la Figura 3.9 se observa que el capstan no presenta zonas de color rojo. De modo que las pequeñas zonas color amarillo y verde del componente soportarían una fuerza de 9.799 MPa. Por lo tanto, el componente no se vería afectado al soportar una carga de 15 N.

Además, el análisis proporciona el desplazamiento que sufrirá el componente (Figura 3.10); donde se observa de color rojo en el capstan la zona que sufriría un mayor

desplazamiento y validado con la gráfica del lado derecho, indica que sería un valor de 0.06442 mm, lo cual es un desplazamiento aceptable que no afecta al componente.

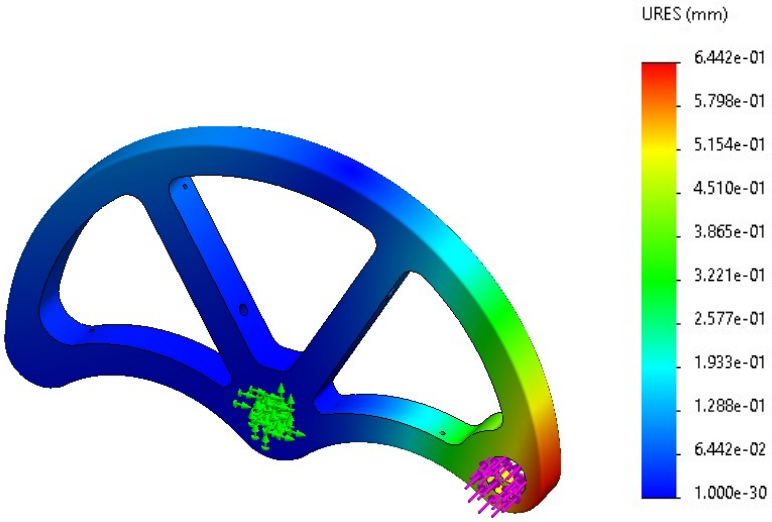


Figura 3.10: Desplazamiento - capstan.

3.3.2. Análisis de elemento finito del componente RD.01.130 - Perno

Para el análisis del perno se consideran condiciones similares a las del capstan, una geometría fija y una carga de 15 N (Figura 3.11).

(a) (b)

Figura 3.11: Restricciones para análisis de elemento finito al perno. (a) Sujeción y carga en perno. (b) Mallado de perno.

1. La geometría fija o de sujeción se colocó en la parte más gruesa del perno, donde estarán los rodamientos que se ensamblan a la base soporte del capstan (Flechas de color verde de la Figura 3.11a).
2. Carga de 15 N (carga máxima que soportará el robot tipo Delta en su EF), perpendicular al perno y aplicada en la zona donde se ensambla el capstan (Flechas de color púrpura de la Figura 3.11).
3. Se creó una malla estándar con distancia entre nodos de 1.116 mm y una tolerancia entre ellos de ± 0.055 mm, dando un total de 13407 nodos (Figura 3.11b).

En la Figura 3.12 se muestran los resultados del análisis de esfuerzos, que al igual que el análisis del capstan y al ser del mismo material (PLA), tiene el mismo esfuerzo máximo a la tensión (antes de deformación) = 50 Mpa, del cual para la carga de 15 N se tiene un esfuerzo a tensión de 5.061 MPa representado en color rojo en el perno; siendo un valor que en simulación no genera mayor atención por posible ruptura o deformación.

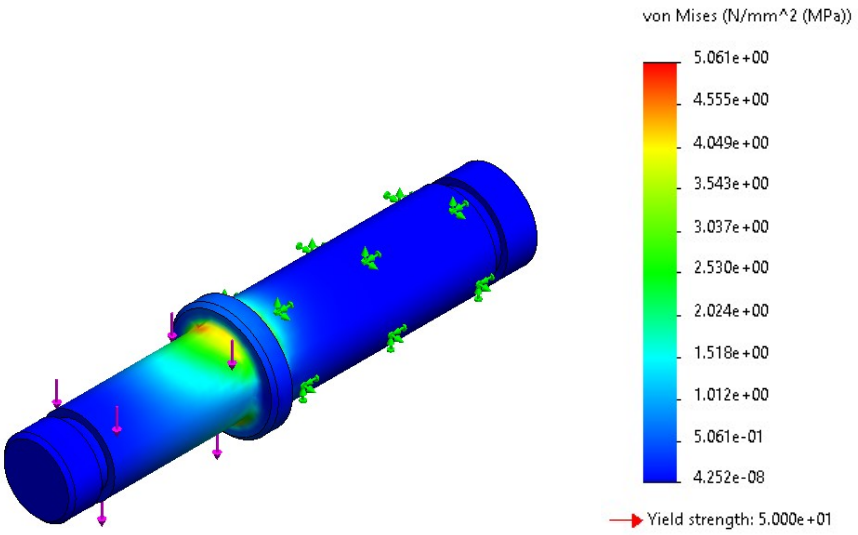


Figura 3.12: Análisis de esfuerzos - perno.

En cuanto al desplazamiento se observa que la zona roja tendría un desplazamiento de 0.03644 mm si se aplica una carga de 15 N con respecto de su posición inicial (ver Figura 3.13), una distancia que al igual que el capstan no afecta en el funcionamiento del componente.

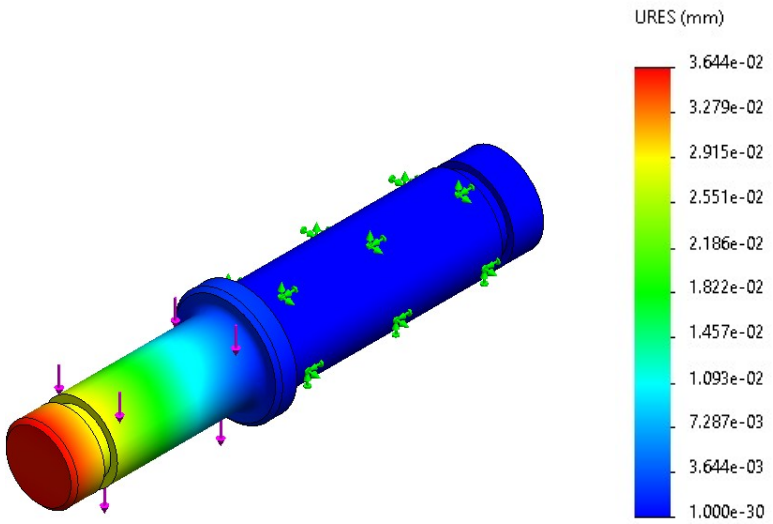


Figura 3.13: Desplazamiento - perno.

En conclusión, en el análisis de elemento finito se logró determinar que los diseños

propuestos para RD.01-135-Capstan y para RD.01-130-Perno, son capaces de soportar las cargas que actuarán en el robot tipo Delta. Cabe mencionar que adicional a los resultados del análisis de elemento finito para el perno, se realizaron pruebas físicas de este componente con el objetivo de evaluar su resistencia debido a que en la simulación se considera una pieza completamente sólida; sin embargo, la forma de fabricación de los componentes en impresión 3D depende de la temperatura del extrusor para lograr una correcta adhesión del material entre cada capa.

3.3.3. Pruebas físicas de resistencia de PLA y ABS

A continuación, se describen las pruebas realizadas al perno en material PLA y ABS. Estas pruebas tienen como objetivo comparar la resistencia de cada material y seleccionar el que ofrezca mejores resultados.

Las pruebas se realizaron a utilizando un banco de pruebas manual que tiene integrado un sensor de fuerza marca Mecmesin AFG 100N y que soporta una fuerza máxima de 100 N (Figura 3.14).

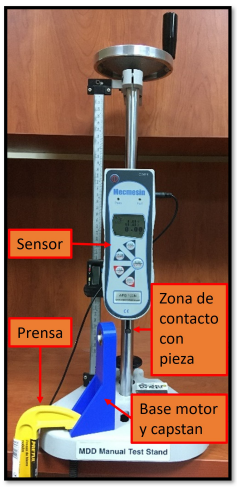


Figura 3.14: Banco de pruebas con sensor de fuerza.

El banco de pruebas permite ejercer una carga a cualquier pieza que pueda sujetarse a su base, el sensor se encuentra ensamblado a un tornillo sinfín que se encuentra en la parte trasera del banco de pruebas y permite desplazar el sensor hacia arriba y abajo. El sensor tiene una punta en la parte inferior que cuando entra en contacto con la pieza a la que se quiere aplicar la carga, muestra en la pantalla la fuerza que se está aplicando a la pieza.

Por lo tanto, para esta prueba se ensambló la base soporte para capstan y motor (RD.01-125 - BM.C) con sus respectivos rodamientos y el perno (RD.01-130 - Perno.Capstan) para fijarlos al banco de pruebas (ver Figura 3.14), de tal forma que permitió ejercer la carga en el extremo del perno para representar la carga que estaría actuando en este componente al retroalimentar fuerza por colisión con objetos virtuales (ver Figura 3.15).

Resultados

La Tabla 3.4 muestra los resultados de las pruebas físicas de resistencia y la configuración para impresión 3D de cada material (PLA y ABS). Cabe señalar que todos

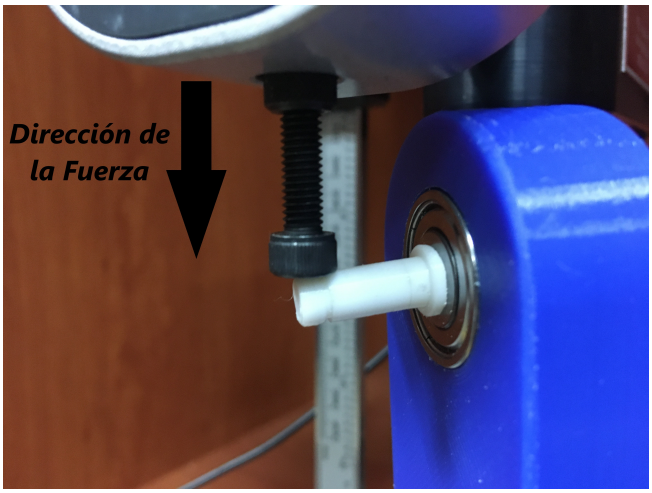


Figura 3.15: Carga actuada en extremo del perno.

los componentes para estas pruebas se imprimieron con una calidad fina de 0.1 mm de altura por cada capa, con un 100 % de llenado para ambos materiales y a una temperatura de 60°C en el plato, resaltando que para el ABS era necesario agregar un pegamento especial que incrementa su adherencia al aumentar la temperatura y viceversa, permitía retirar la pieza terminada fácilmente al enfriarse el plato (temperatura ambiente).

Tabla 3.4: Configuraciones, pruebas físicas de carga y resultados de impresión con material PLA y ABS, realizadas en el perno.

Material		**Temperatura del extrusor (°C)**	**Fuerza soportada (N)**
PLA	**ABS**		
	x	250	85.36
	x	250	71.68
	x	250	58.86
x		205	100
x		205	*242.99
x		205	*176.68
* Estos resultados se obtuvieron con un sensor de carga que soporta 50 $kg \approx 490.5\ N$, este sensor se integro al banco de pruebas debido a que el sensor Mecmesin AFG 100N solo soporta 100 N.			

Estas pruebas permitieron concluir sobre el diseño del perno, siendo este capaz de soportar sin problema una carga de 15 N en el EF. Además, las pruebas físicas de resistencia de material permitieron elegir el polímero PLA para el resto de los componentes, haciendo notar que esta elección es debido a que las impresoras 3D con las que se cuenta en la institución no alcanzan la temperatura necesaria para trabajar correctamente el ABS.

Capítulo 4

Desarrollo de la metodología - Etapa 2

En este capítulo se presenta la etapa 2 del desarrollo de la metodología propuesta donde se describe la manufactura de los componentes, el desarrollo del sistema eléctrico para el robot y el ensamble del final robot tipo Delta. También, se describe el método utilizado para realizar el control de posición de los motores DC y su validación. Además, se describe la implementación del algoritmo geométrico de la cinemática en un sistema embebido y su validación, con base en lo propuesto en [13]. Adicionalmente, se describen los estados de trabajo del robot tipo Delta y la interfaz generada para el usuario.

4.1. Manufactura de componentes

La manufactura de componentes se realizó mediante prototipado rápido 3D, haciendo uso de la impresora 3D modelo ENDER-3 V2 y mediante el software slicer llamado Ultimaker Cura se generó el código G para la impresión.

Inicialmente, se fabricaron solo los componentes que intervienen en la transmisión por cable, con la finalidad de validar el funcionamiento de estos componentes.

En la Tabla 4.1 se muestran los componentes que se imprimieron en 3D para las pruebas de funcionamiento y su configuración en el programa Cura. Cabe mencionar que los componentes se utilizaron para validar el ensamble y el funcionamiento del mecanismo de tensión para la transmisión por cable, por lo que no se requirió de la máxima calidad de impresión en los componentes y se limitó el porcentaje de llenado.

Tabla 4.1: Configuración para impresión 3D para pruebas de funcionamiento en PLA.

Componente	**Material**	**Porcentaje de llenado (%)**	**Calidad - Altura de la capa (mm)**		
			0.1	0.12	0.16
RD.01-135 - Capstan.	PLA	40			**x**
RD.01-140 . Polea.	PLA	100		**x**	
RD.01-130 - Perno.	PLA	100	**x**		
RD.01-125 - Base motor.	PLA	40		**x**	
Para todos los componentes la temperatura de extrusor y del plato son de 200 y 60 °C respectivamente.					

En la Figura 4.1 se presenta el modelo CAD donde se describen los componentes y la forma de realizar el ensamble, haciendo notar la trayectoria del cable al rodear el capstan y el numero de vueltas necesarias en la polea para posteriormente engancharlo al resorte de tensión.

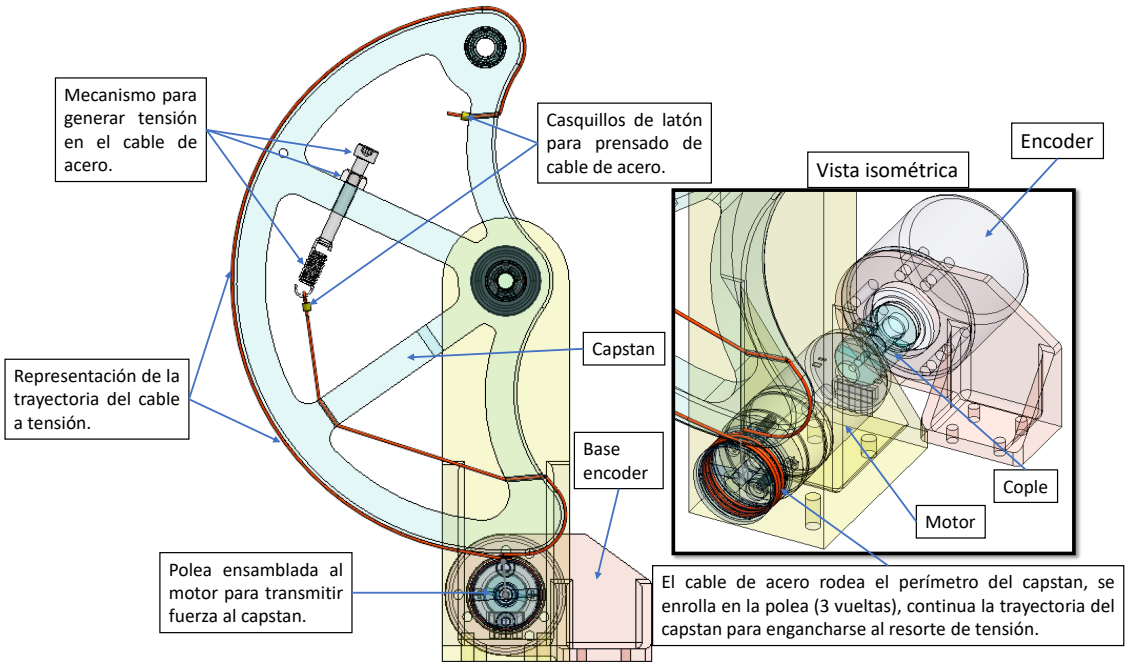


Figura 4.1: Modelo CAD para transmisión por cable.

Además, para las pruebas de funcionamiento y el ensamble se consideraron los siguientes componentes comerciales:

1. Motor DC marca Pololu #4861, 12 V con las siguientes características principales:
   Motor sin carga
   - Velocidad Angular (ω) = 1800 rpm
   - Corriente (I) = 100 mA = 0.1 A

   Máxima eficiencia del motor
   - Torque Nominal (M_N) = 0.93 kg mm = 0.00093 kg m = 0.0091 Nm
   - Corriente Nominal (I_N) = 0.26 A
   - Torque de paro (Stall Torque, M_H) = 7.1 kg mm = 0.0071kg m = 0.0696 Nm
2. Cable de acero de 0.9 mm.
3. Varilla de latón de 3 mm de diámetro (fabricación de casquillos para prensado de cable de acero).
4. Tornillo M4.
5. Tuerca de seguridad.
6. Resorte de tensión.
7. Rodamientos con diámetro externo = 22 mm y diámetro interno = 7 mm.
8. Rodamientos con diámetro externo = 13 mm y diámetro interno = 6 mm.

Posteriormente, se realizó el ensamble de los componentes mencionados anteriormente (Figura 4.2) y se realizaron pruebas de movimiento manuales (sin voltaje en el

motor), donde se probó el funcionamiento de los componentes que están presentes en la transmisión por cable; se validó que el mecanismo de tensión formado por el resorte, tuerca y tornillo si proporciona la suficiente tensión para mantener el cable alrededor del capstan sin perder su posición. Así también, el cable que rodea la polea se desplazó correctamente por el ancho de la misma, manteniendo su posición dentro de las muescas que dirigen su desplazamiento y el movimiento de la polea fue uniforme al movimiento del capstan.

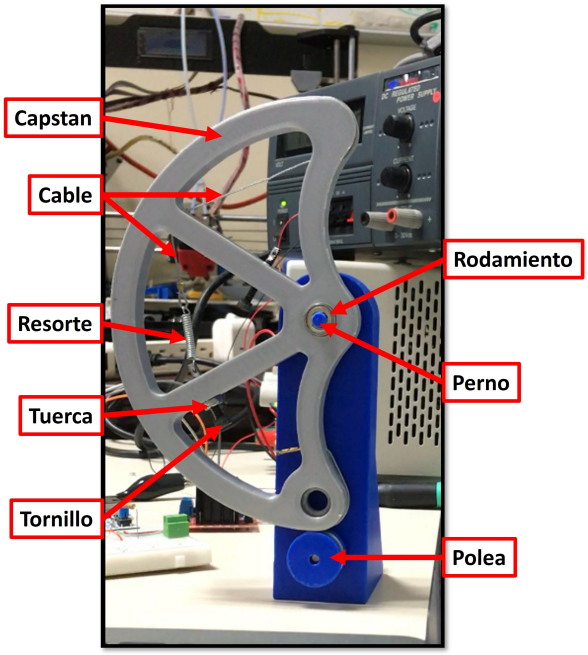


Figura 4.2: Ensamble de componentes 3D para pruebas de funcionamiento.

Por otra parte, se generó un código para validar el funcionamiento de componentes con voltaje en el motor. El control del motor se realizó a través del microcontrolador de la tarjeta *STM32F767ZI*, se utilizó el programa *STM32CUBEIde* y mediante PWM (Pulse Width Modulation) se controló el motor.

Adicionalmente, el siguiente link contiene evidencias de las pruebas de funcionamiento de los componentes para la transmisión por cable con voltaje en el motor, (ver [videos_pruebas_1]).

Finalizadas las pruebas de funcionamiento de los componentes presentes en la transmisión por cable, se inició la impresión de los componentes en su versión final, con la calidad y porcentaje de llenado correspondiente.

Impresión 3D de componentes

La fabricación de componentes se realizó a través de prototipado rápido. Cabe mencionar, que el material para la impresión de los componentes es el polímero PLA. En la Tabla 4.2 se muestra la configuración utilizada para la impresión 3D de cada componente, el porcentaje de llenado, la calidad de impresión y la cantidad de piezas a fabricar. El porcentaje de llenado para cada componente fue determinado con base en la función de cada uno, es decir, si el componente se ve comprometido por cargas que afecten su función o estructura, se imprimió una pieza completamente sólida, de lo contrario se disminuyó el porcentaje de llenado en la fabricación del componente.

Tabla 4.2: Configuración para impresión 3D de componentes del robot tipo Delta.

Item	No. Parte/Componente	Porcentaje de llenado (%)	Altura de capa (mm)		No. piezas
			0.10	0.12	
1	RD.01-115 - Soporte para ensamble de bases.	40		x	1
2	RD.01-125 - Base Soporte para capstan y motor.	60		x	3
3	RD.01-130 - Perno para ensamble de capstan y base de motor.	100	x		3
4	RD.01-135 - Capstan.	100		x	3
5	RD.01-140 - Polea.	100	x		3
6	RD.01-150 - Perno para ensamble de barras paralelas con articulación.	100	x		12
7	RD.01-155 - Articulación para ensamble en barras paralelas.	100	x		12
8	RD.01-160 - Eje para ensamble de paralelas con capstan y base móvil.	100	x		6
9	RD.01-165 - Base móvil.	100		x	1
10	RD.01-170 - Soporte para efector final.	60		x	1
11	RD.01-175 - Efector final.	20	x		1
La temperatura del extrusor y del plato son de 205 °C y 60 °C respectivamente.					

Mecanizado de componentes

Las bases del robot requirieron de un material diferente al polímero PLA, ya que por sus dimensiones no fue posible imprimirlas en 3D. Para la fabricación de ambas bases del robot se utilizó la cortadora láser CO_2 - GUTSTARK NT-4040. En la base para trabajos horizontales RD.01-110 se utilizó una hoja de Triplay de 9 mm de espesor (Figura 4.3a) y en la base para trabajos verticales RD.01-120 se utilizó una hoja de MDF de 9 mm de espesor (Figura 4.3b).

(a) (b)

Figura 4.3: Fabricación de bases del robot mediante cortadora Láser. (a) Base de robot Delta para trabajos horizontales. (b) Base de robot Delta para trabajos verticales.

Por otro lado, Las barras paralelas RD.01-145 (Figura 4.4) son un componente crucial en las cargas que soportará el robot, estas se adquirieron de aluminio 6061 de un diámetro de 1/4 de pulgada y se ajustaron a una longitud de 20 cm.

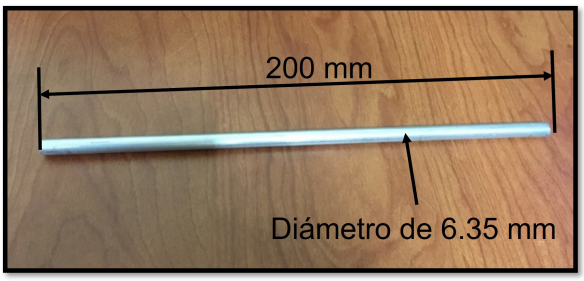


Figura 4.4: Barras paralelas de aluminio 6061.

En el caso de los pernos RD.01-150 (Figura 4.5), al ser un componente con un diámetro de 3 mm, utilizado como unión entre las articulaciones de las barras paralelas, el capstan y la base móvil, pudiera sufrir una fractura si el material no es lo suficientemente resistente cuando el robot soporte cargas. Por lo tanto, se fabricaron de aluminio 6061 mediante un Torno CNC modelo Resource Kit Unimat CNC 166074RP, debido a que el aluminio en comparación con el PLA proporciona mayor resistencia. Los diagramas mecánicos 2D están disponibles en el Apéndice A.

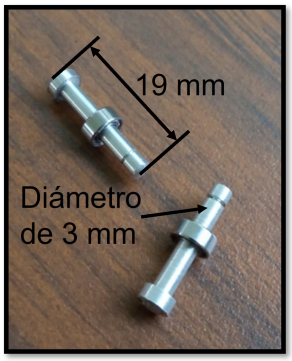


Figura 4.5: Perno de aluminio para articulación.

Los casquillos de latón se utilizaron para prensar el cable de acero que rodea el capstan en la transmisión por cable (Figura 4.6). Estos casquillos se cortaron a una longitud de 4 mm de una barra de latón de 4 mm de diámetro, perforando el centro a 1 mm de diámetro para pasar el cable y prensarlo.

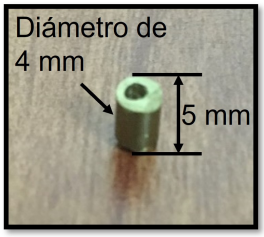


Figura 4.6: Casquillo de latón.

4.2. Desarrollo del sistema eléctrico para el robot tipo Delta

Finalizada la fabricación de los componentes del robot tipo Delta se inició con el desarrollo del sistema eléctrico para alimentar los motores DC. Estas conexiones se realizaron con base en el diagrama electrónico que se muestra en el Apéndice B. Cabe mencionar que para controlar el sentido de giro y la velocidad de los motores se utilizó el driver DRV8874 de Texas Instruments, debido a que puede suministrar una corriente de 2.1 A (con un limite de corriente hasta 4.4 A) a un motor DC bidireccional, no requiere disipadores, cuenta con detección y regulación de corriente, con proteccion contra sobrevoltaje y sobrecorriente.

Además, con el objetivo de facilitar las conexiones en el robot tipo Delta se diseñó una PCB (Printed Circuit Board) en el programa Altium designer. Durante el diseño

de la PCB se determinó colocar la tarjeta STM32 NUCLEO F767ZI sobre la PCB, separando la parte digital de la parte de potencia, la vista 3D de la PCB se presenta en la Figura 4.7.

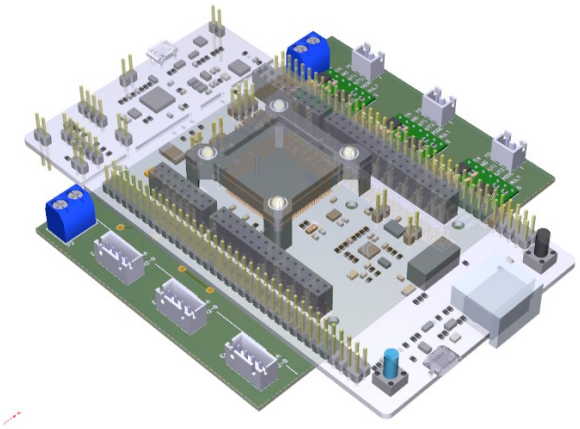
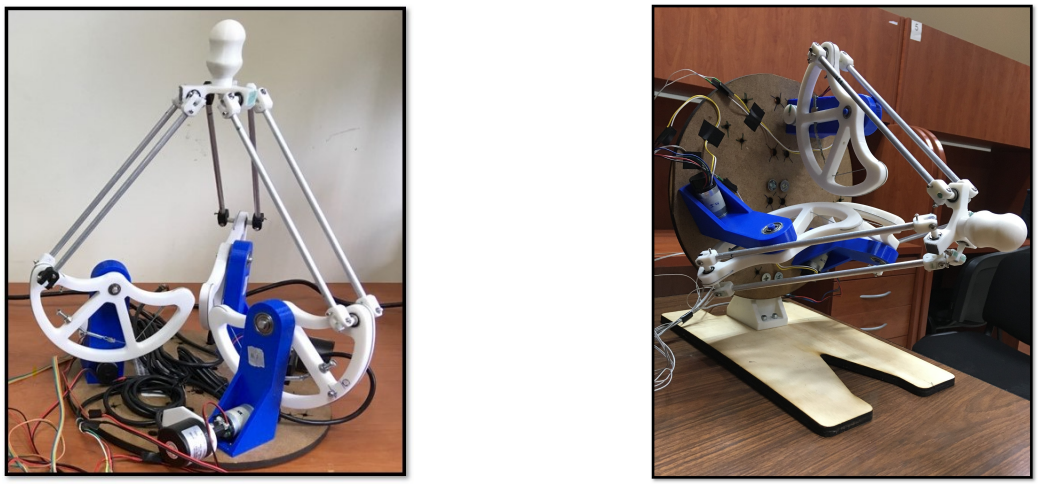


Figura 4.7: Vista 3D de PCB con tarjeta montada.

4.3. Ensamble del robot Delta

En esta sección se muestra el ensamble final del robot tipo Delta, su configuración para trabajos verticales (Figura 4.8a) y su configuración para trabajos horizontales (Figura 4.8b).



(a) (b)

Figura 4.8: (a) Configuración para trabajos verticales. (b) Configuración para trabajos horizontales.

Cabe mencionar que con el objetivo de mejorar la resolución del robot en el EF, se sustituyo el encoder que trae acoplado de fábrica, ya que este solo proporcionaba 211.2 ppr (pulsos por revolución) con la técnica de cuadratura y se reemplazó por un encoder que proporciona 1600 ppr. Por lo tanto, fue necesario diseñar una base y un cople para adaptar el encoder al motor, ambas piezas se fabricaron de PLA mediante impresión 3D.

Se realizaron pruebas de funcionamiento donde se validó el movimiento de sus articulaciones y los demás componentes que integran al robot. En las pruebas se encontraron dos restricciones que no se visualizaron en el diseño del robot. La primera restricción se encontró cuando se desplazaba el EF por los ejes (x,y) mientras en el eje z se mantenía una posición cercana a la más retraída del robot (entre 0.130 y 0.140 m), durante estos movimientos las barras paralelas podían colisionar con alguna de las bases de los motores. Así mismo, la segunda restricción se observó al llevar el robot a su posición más retraída (0,0,0.130 m), al realizar un movimiento para sacarlo de esa posición era necesario ejercer una mayor fuerza para lograr el desplazamiento. Por lo tanto, para corregir ambas restricciones se agregaron límites mecánicos al diseño, de esta forma se evitó que el capstan llegara a las zonas donde colisionaba o requería de mayor fuerza para salir de ella.

En el siguiente enlace se encuentra un video donde se aprecian algunos movimientos realizados en las pruebas de funcionamiento del robot tipo Delta en la configuración para trabajos verticales (ver [video_pruebas_2]).

A continuación se describen algunas recomendaciones de acuerdo a las modificaciones realizadas durante el proceso de validación de funcionamiento y ensamble del robot tipo Delta.

1. Considerar la fabricación de los pernos en aluminio mediante mecanizado o validar que los pernos fabricados mediante impresión 3D en PLA estén completamente uniformes, es decir, que estén completamente rectos y no presenten una ligera desviación. Esto pudiera pasar debido a la forma de impresión del componente, al ser una pieza cilíndrica debe tomarse como base una de las zonas planas al inicio o final de la pieza y considerando la longitud del cilindro puede comprometerse el detalle de impresión al final de la pieza, ocasionado por falta de una correcta adhesión a la base de la impresora 3D, generando piezas con pequeñas imperfecciones que afectan el funcionamiento del robot tipo Delta.
2. Considerar que el encoder de los motores de bajo costo no proporciona la resolución suficiente para las aplicaciones propuestas, de modo que para este robot tipo Delta se sustituyó por uno que ofrece mejores características.
3. Considerar la fabricación de los componentes pequeños en aluminio mediante mecanizado (ejes y pernos), de esta forma se proporciona mayor resistencia que al ser fabricados de PLA.
4. Considerar la relación de posición de cada componente en el diseño CAD para generar una simulación completa de los movimientos del robot y validar que no existan colisiones entre sus componentes.
5. Realizar pruebas de funcionamiento manuales para validar que el robot pueda desplazarse sin dificultades en todo su espacio de trabajo.
6. Considerar el ensamble de los motores en orden secuencial (1,2,3), tomando un plano cartesiano en el centro de la base fija del robot, ubicando el motor número 1 en la parte derecha y de forma perpendicular al eje X, de forma que el capstan se encuentre alineado al eje X, el motor 2 y 3 se deben ubicar a una separación de 120 grados respectivamente y en sentido antihorario, en el Apéndice C se muestra el ensamble final y el orden correcto de los motores.
7. Considerar que al ensamblar la base general en la configuración para trabajos horizontales, el motor número 1 debe ubicarse en la parte derecha de la base fija del robot, sin olvidar que el orden de los motores debe ser secuencial y en sentido antihorario

La Figura 4.9 muestra el ensamble del robot tipo Delta en configuración para trabajos verticales, donde se puede visualizar la conexión del sistema electrónico (Tarjeta STM32 NUCLEO F767ZI y PCB) y la fuente de alimentación BK PRECISION®.

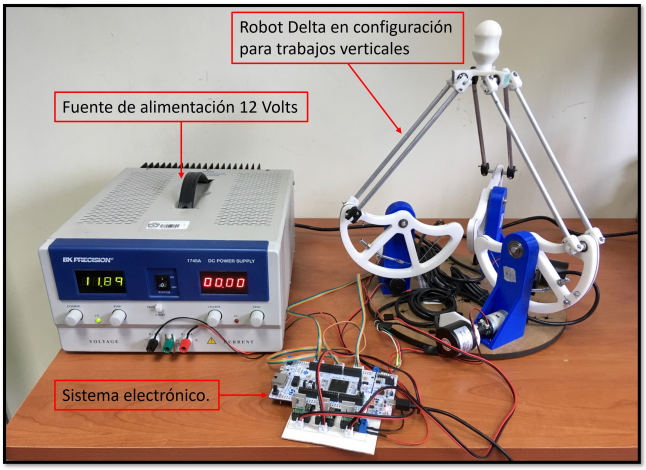


Figura 4.9: robot Delta, sistema electrónico y fuente de alimentación.

4.4. Control de posición de Motores de DC

En esta sección se describe cómo se realizó el control de posición de los motores, basado en el método Rapid Control Prototyping (RCP) [46]. Este método permite reducir tiempos de programación utilizando bloques de Simulink® para generar código C avanzado del control y enlazarlo con el programa STM32CubeIDE de la tarjeta de desarrollo STM32 NUCLEO F767ZI. De modo que se consideraron los siguientes puntos para el desarrollo del control de posición de los motores:

1. Respuesta transitoria de cada motor.
2. Función de transferencia para cada motor.
3. Valores iniciales de las ganancias PID para el controlador de cada motor.
4. Sintonización fina de controladores en tiempo discreto.

Obtención de la respuesta transitoria de los motores

La respuesta transitoria del motor se obtuvo a partir de la medición de la velocidad del eje como se menciona en [47], donde se procesan los datos a través de un modelo Simulink/Matlab®. Para el caso de la velocidad del motor se utilizó la derivada de tustin, este método se describe en [48], donde aproximan la velocidad del motor a partir de una diferenciación directa. Por consiguiente, a través de la posición angular del eje del motor que se obtiene mediante el acoplamiento de un encoder con 1600 ppr y a una frecuencia de muestreo de 1 kHz que comunica con Simulink® a una frecuencia de 100 Hz y un baud rate de 115200 Bits/s se obtuvo la respuesta transitoria de cada motor que se muestra en la Figura 4.10.

Después, mediante el comando ***systemIdentification*** en Matlab® se obtuvo la función de transferencia de cada motor. Estas funciones de transferencia se muestran en la Figura 4.11 y se emplean para desarrollar un modelo Simulink® en lazo cerrado

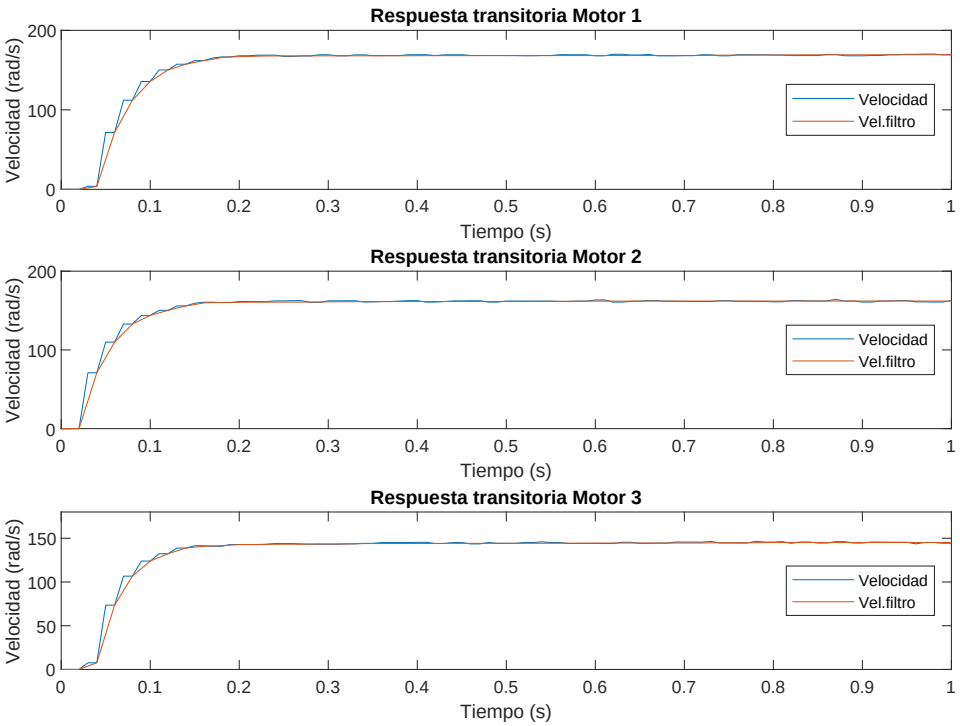


Figura 4.10: Respuesta transitoria de motores.

donde se obtienen los valores iniciales para el control PID de cada motor. Los datos requeridos en el worskpace de Matlab/systemIdentification son: la velocidad en rad/s, la entrada escalón a su máximo voltaje de alimentación y el tiempo en el que se realizó la muestra, para este caso 10 milisegundos.

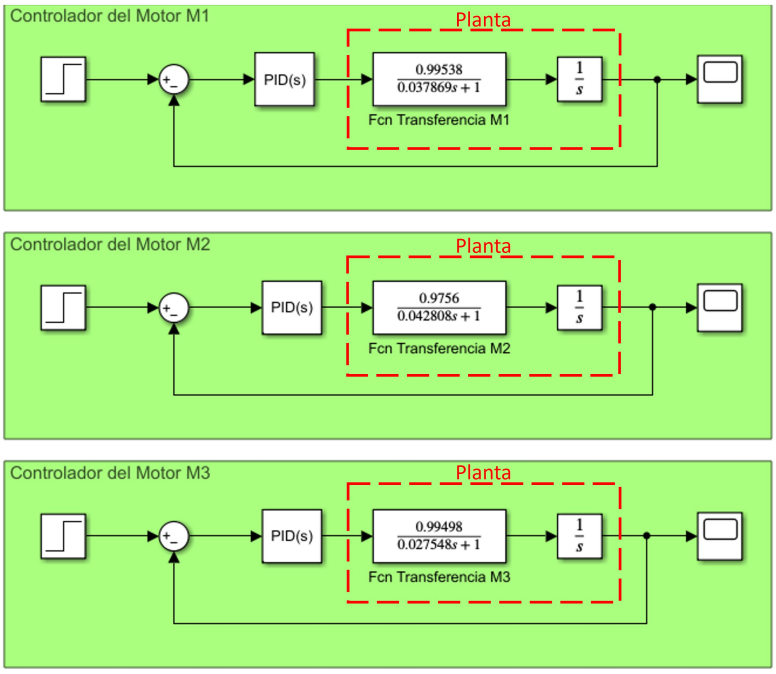


Figura 4.11: Diagrama de bloques en lazo cerrado para la sintonización de los controladores PID en tiempo continuo.

Sintonización de controladores PID en tiempo continuo

El modelo Simulink® de la Figura 4.11 muestra la planta que representa a cada motor, conformadas por la función de transferencia y un bloque que integra la velocidad del motor para obtener la posición del eje del motor. Para obtener los valores iniciales de las ganancias de los controladores PID en lazo cerrado se utilizó la herramienta ***PIDTuner***, esta herramienta permite sintonizar los valores del controlador PID basándose en el bloque que contiene la función de transferencia. Para ingresar a la herramienta ***PIDTuner*** se deben seleccionar los parámetros del bloque PID del modelo Simulink®; una vez dentro de los parámetros, la herramienta muestra la respuesta del bloque PID sin sintonizar (línea punteada de la Figura 4.12) y la respuesta del bloque sintonizado (línea continua de la Figura 4.12).

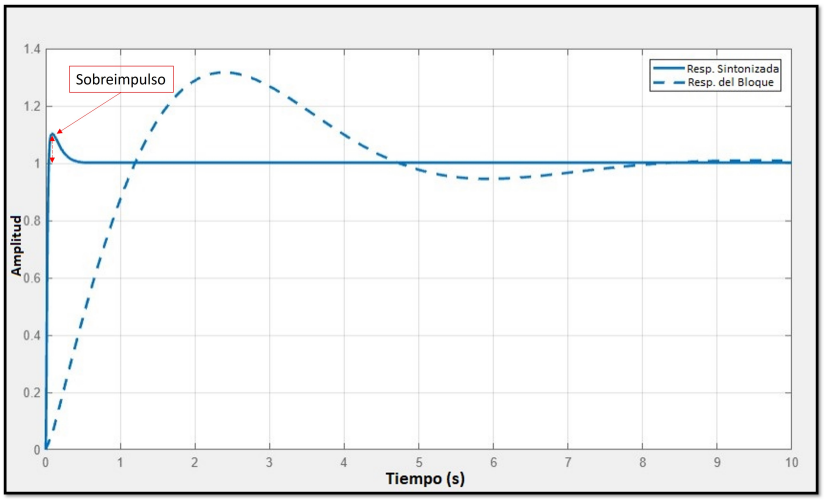


Figura 4.12: Sintonizado del bloque PID mediante *PIDTuner*.

La Tabla 4.3 muestra las ganancias generadas en ***PIDTuner*** para cada motor. Por otra parte, en la Figura 4.12 se aprecia un sobreimpulso en la respuesta del controlador PID sintonizado, por lo que se realizó una sintonización fina de los controladores para eliminar el sobreimpulso.

Tabla 4.3: Ganancias iniciales de controladores.

	Motor 1	**Motor 2**	**Motor 3**
Kp	55.3	49.9	76.1
Ki	340	271.4	642.7
Kd	1.57	1.6	1.57

Realizada la sintonización fina, la Figura 4.13 muestra la respuesta del controlador PID donde se elimina el sobreimpulso.

4.4.1. Validación del control de posición de los motores

Cabe mencionar que para poder implementar el modelo Simulink en el sistema embebido de la tarjeta NUCLEO-F767ZI, es necesario convertir el lazo de control de tiempo continuo a tiempo discreto como se menciona en [46], recordando que el tiempo continuo es utilizado para simulaciones en escenarios ideales.

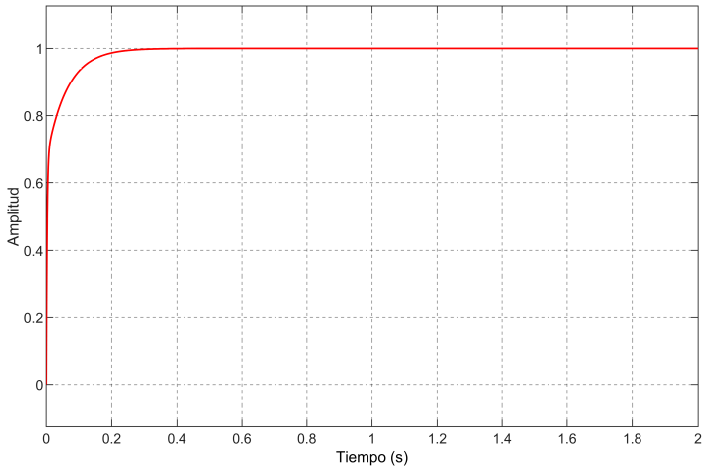


Figura 4.13: Respuesta de controlador PID con sintonización fina para eliminar sobreimpulso.

Sintonización fina de los controladores digitales en tiempo discreto

Considerando que la respuesta de los motores es muy rápida (menor a 0.3 segundos) como se puede observar en la Figura 4.13, se redujo cada ganancia en la sintonización fina debido a que para la aplicación háptica no es necesaria toda la velocidad de los motores (1800 rpm). Además, se probaron dos controladores diferentes (PD y PID), ajustando manualmente las ganancias para conseguir el menor grado de error posible y determinar el controlador que muestre mejor funcionamiento.

En primer lugar, se probó el controlador PD debido a que en la literatura se recomienda este controlador para sistemas en movimiento, ya que este controlador ayuda a reducir el tiempo de subida y los errores transitorios como el sobreimpulso y las oscilaciones en la salida [49]. En este controlador se observó que la respuesta es de 0.6 seg (Figura 4.14), una respuesta más lenta en comparación con la respuesta anterior donde todavía no se reducían las ganancias.

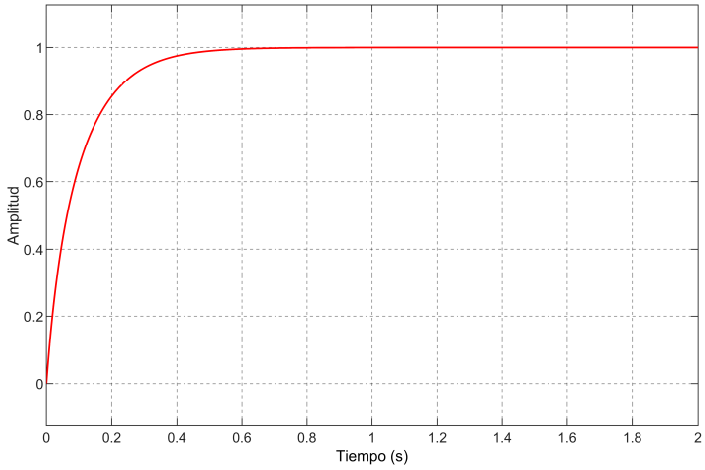


Figura 4.14: Respuesta del controlador PD con sintonización fina.

En la Tabla 4.4 se muestran las ganancias utilizadas para el controlador PD, así como el error promedio en la posición deseada de cada motor.

Tabla 4.4: Ganancias para controlador PD.

	Motor 1	**Motor 2**	**Motor 3**
Kp	10	10	11
Kd	0.1	0.1	0.1
Error	**0.35 grados**	**0.45 grados**	**0.35 grados**

En segundo lugar se realizó la prueba con el controlador PID, que es el tipo de controlador más utilizado y requiere de la sintonización de las tres ganancias, proporcional (Kp), integral (Ki) y derivativo (Kd) [23], mostrando un menor grado de error en la posición en comparación con el controlador PD. En la Figura 4.15 se muestra la simulación de la respuesta del controlador PID donde se observa que es más lenta su respuesta (0.8 segundos).

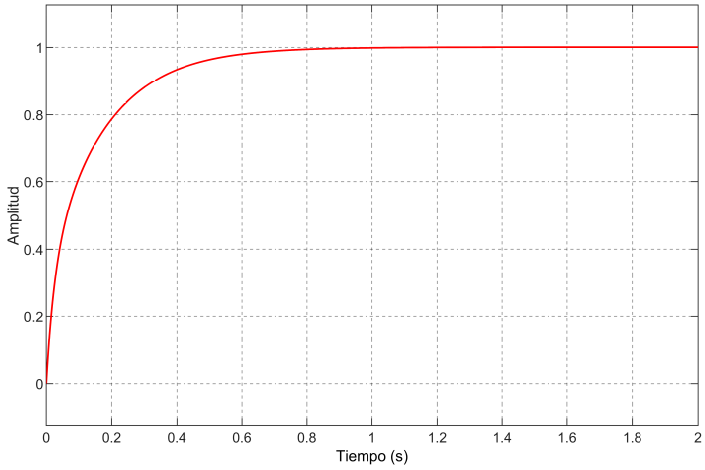


Figura 4.15: Respuesta sintonizada del controlador PID.

Las ganancias y el error promedio del controlador PID se presentan en la Tabla 4.5.

Tabla 4.5: Ganancias para controlador PID.

	Motor 1	**Motor 2**	**Motor 3**
Kp	8	8	9
Ki	0.06	0.05	0.06
Kd	0.3	0.3	0.3
Error	**0.2 grados**	**0.2 grados**	**0.2 grados**

Dado que en la practica se observó un funcionamiento similar en los dos controladores y en ningún caso se observó oscilamiento al llegar a la posición deseada, cualquiera de los dos controladores es funcional; sin embargo, con el controlador PID se obtuvo un error promedio de 0.2 grados y se percibió una mejor respuesta al ejercer fuerza en el EF, debido a las propiedades generadas por la ganancia del integrador.

4.5. Implementación del algoritmo geométrico de la cinemática en un sistema embebido

Recordando que la implementación del algoritmo geométrico en un sistema embebido está basada en el trabajo propuesto por Avizzano et al. [13], donde proponen una solución enfocada en propiedades geométricas que reducen la carga computacional al calcular las 3 cinemáticas juntas (Cinemática directa, Cinemática diferencial y Cinemática diferencial inversa), expresando sus cálculos en pocas líneas de código.

La Figura 4.16 indica las variables usadas en la implementación del algoritmo geométrico, donde R_b es el radio del círculo formado al unir los 3 centros de los ejes que conectan los brazos con la base móvil, R_m es el radio al formar un círculo hasta el centro de cada eje de ensamble de los 3 capstan, R_B es el radio máximo cuando el EF se encuentra alineado horizontalmente en la posición más extendida, E_i son los codos del dispositivo que representan los centros de la zona de ensamble del capstan con el eje que une las barras paralelas, L_F es la longitud de los antebrazos del robot y L_A es la longitud del brazo del robot, del centro del eje de ensamble del capstan al centro del eje de ensamble con el antebrazo del robot. Los ángulos articulares del capstan se representan con ϕ_1, ϕ_2 y ϕ_3.

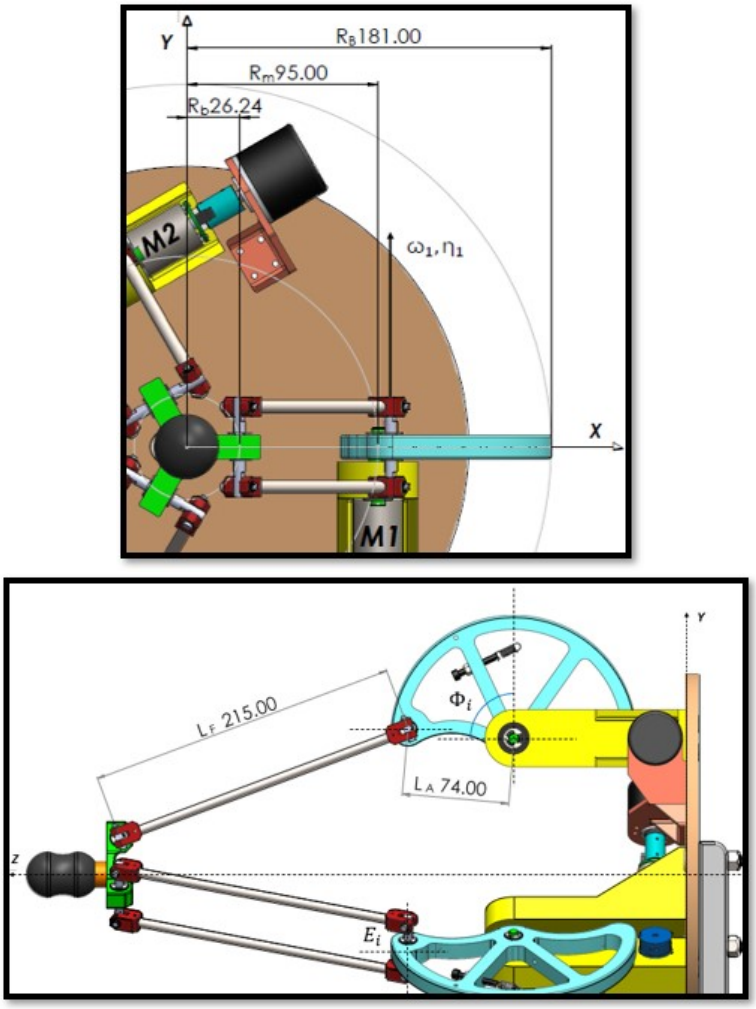


Figura 4.16: Dimensiones del robot tipo Delta.

Las técnicas tradicionales utilizadas para calcular la cinemática directa/inversa consisten en encontrar las posiciones espaciales de los codos del dispositivo (E_i) y después, encontrar el punto de intersección de 3 esferas diferentes cuyos centros están en los codos [50, 51].

Sin embargo, Avizzano et al. resolvieron la cinemática directa que proporciona la ubicación de la base móvil con respecto a la base fija dados los ángulos articulares ϕ_i, identificados por un marco base cuyo origen O_M esta alineado a lo largo del eje Z cuando $\phi_1 = \phi_2 = \phi_3 = 0$. Por lo tanto, utilizaron la información geométrica y los desplazamientos articulares para calcular las posiciones de los codos:

$$E_i = L_i + CC_i \tag{4.1}$$

$$CC_i = [R_m cos(\theta_i), R_m sin(\theta_i), 0]^T \tag{4.2}$$

$$L_i = [L_A cos(\phi_i) cos(\theta_i), L_A cos(\phi_i) sin(\theta_i), L_A sin(\phi_i)]^T \tag{4.3}$$

donde CC_i son las posiciones centrales de los capstan, $\theta_{1..3} = (0, 2\pi/3, -2\pi/3)$ es la ubicación de los codos en radianes, igual a 120° y L_i proporciona la dirección del brazo.

Cinemática Directa

En las soluciones clásicas se requiere encontrar el origen de la base móvil O_M como la intersección de tres esferas, centradas en E_1, E_2 y E_3 respectivamente y con un radio igual a la longitud del antebrazo L_F, lo que requeriría escribir y resolver un sistema completo formado por tres ecuaciones cuadráticas. No obstante, los autores resolvieron el problema considerando que los 3 codos (E_1, E_2, E_3) pertenecen a una única esfera centrada en O_M y que pasa por cada codo (ver Figura 4.17), utilizando una serie de propiedades geométricas que les permitió calcular el desplazamiento sobre el eje mediante el teorema de Pitágoras.

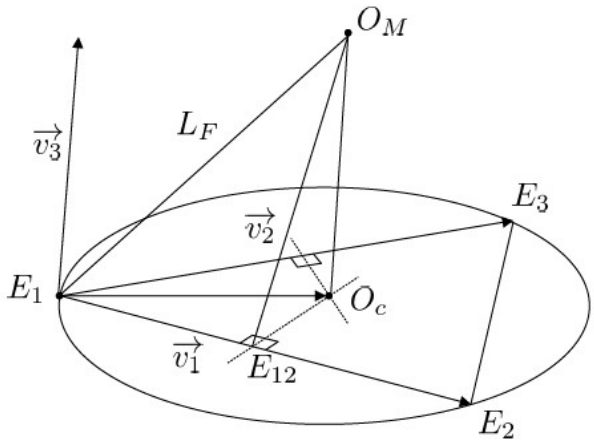


Figura 4.17: Propiedades geométricas utilizadas por Avizzano et al. para resolver las cinemáticas de un robot tipo Delta, tomada de [13]

Adicionalmente, consideraron un sistema de referencia no cartesiano formado por el triángulo que se crea al unir los vértices E_1, E_2 y E_3 con líneas, tomando a E_1 como el origen, mientras v_1, v_2 y v_3 representan sus ejes, definiéndolos como:

$$
\begin{aligned}
v_1 &= E_2 - E_1 \\
v_2 &= E_3 - E_1 \\
v_3 &= v_1 \times v_2
\end{aligned} \tag{4.4}
$$

Considerando a O_c en el centro del círculo circunscrito, estando en la intersección de las bisectrices perpendiculares de las aristas del triángulo. De modo que, mediante relaciones algebraicas y el uso de la pseudoinversa, obtuvieron:

$$
O_c = (v_1 C_{22}(C_{11} - C_{12}) + v_2 C_{11}(C_{22} - C_{12}))/(2\Delta) \tag{4.5}
$$

donde: $C_{11} = |v_1|,\ C_{22} = |v_2|,\ C_{12} = v_1^T v_2$, definiendo el determinante de la pseudoinversa como: $\Delta = C_{11}C_{22} - C_{12}C_{12}$.

Por lo tanto, determinaron el desplazamiento vertical en el eje v_3 imponiendo la identidad de Pitágoras en el triángulo formado por (O_M, O_c y E_1). Obteniendo lo siguiente:

$$
\gamma_3 = \sqrt{\frac{L_F^2 - O_c^T O_c}{v_3^T v_3}} \tag{4.6}
$$

donde γ_3 es la coordenada a lo largo del versor v_3 de la base móvil en el nuevo sistema de referencia. Encontrando la posición de la base móvil:

$$
O_M = E_1 + O_c + v_3\gamma_3 \tag{4.7}
$$

Cinemática Diferencial

En cuanto a la cinemática diferencial, utilizaron la Ecuación (4.8) y la resolvieron por *Jd* (Jacobiano directo) y para la cinemática diferencial inversa utilizaron el *Jinv* (Jacobiano inverso).

$$
\dot{O}_M = J_d\omega \tag{4.8}
$$

$$
\omega = Jinv\dot{O}_M \tag{4.9}
$$

donde $\omega = [\omega_1, \omega_2, \omega_3]^T$

La cinemática diferencial directa y diferencial inversa las calcularon juntas. Determinaron la cinemática utilizando una derivación geométrica no convencional, considerando que la longitud del antebrazo no cambia durante el movimiento y, de modo que para cada codo (i), obtuvieron:

$$
(O_M - E_i)^T(O_M - E_i) = L_F^2 \tag{4.10}
$$

llegando a la siguiente relación:

$$
(O_M - E_i)^T(\dot{O}_M - \dot{E}_i) = 0 \tag{4.11}
$$

Esto les permitió describir la cinemática diferencial inversa, reemplazando cada $\dot{E}_i$ por su representación analítica.

$$
\dot{E}_i = \eta_i \times L_i\omega_i \tag{4.12}
$$

Donde $\eta_i = [sin(\theta_i), cos(\theta_i), 0]^T$, obteniendo:

$$(O_M - E_i)^T \dot{O}_M = (O_M - E_i)^T (\eta_i \times L_i \omega_i) = 0$$
$$\omega_i = \frac{(O_M - E_i)^T \dot{O}_M}{(O_M - E_i)^T (\eta_i \times L_i)} \tag{4.13}$$

Calculando la fila del *Jinv* mediante la Ecuación (4.14)

$$Jinv_i = \frac{(O_M - E_i)^T}{(O_M - E_i)^T (\eta_i \times L_i)} \tag{4.14}$$

Por otra parte, consideraron calcular la cinemática directa por superposición de efectos. Tomando el movimiento de una articulación a la vez, conociendo la dirección del movimiento relacionada con la articulación. Es decir, si sólo se mueve la articulación 3, E_1 y E_2 permanecerán fijos, entonces el triángulo (O_M, E_1, E_2) solo puede girar a lo largo del eje descrito por v_1. Esta dirección la calculan por producto cruz entre el eje v_1, la línea que pasa por la base móvil y el punto medio entre E_1 y E_2, por lo tanto:

$$\tilde{J}d_3 = (O_M - E_{12}) \times v_1 \tag{4.15}$$

Donde

$$E_{12} = \frac{1}{2}(E_1 + E_2) \tag{4.16}$$

Donde $\tilde{J}d$ se refiere a la dirección del jacobiano. Esta ecuación la repiten para la primera y la segunda fila del jacobiano. Una vez determinadas las direcciones jacobianas, las escalaron conociendo el producto de identidad entre el jacobiano directo y el inverso.

$$Jd_i = \tilde{J}d_i / (Jinv_i \tilde{J}d_i) \tag{4.17}$$

Analizado el algoritmo geométrico desarrollado por Avizzano et al. se inicia la implementación en un sistema embebido, por lo que es necesario ingresar las dimensiones mostradas en la Figura 4.16 y las masas de algunos componentes del robot tipo Delta.

4.5.1. Dimensiones y masas de los componentes

En la Tabla 4.6 se muestra un resumen con las dimensiones de los componentes utilizados para la implementación del algoritmo geométrico de la cinemática de este robot tipo Delta. En el Apéndice C se muestra el modelo CAD con las dimensiones del robot tipo Delta.

Tabla 4.6: Dimensiones del robot tipo Delta.

Radio de acoplamiento con base móvil (R_b)	26.24 mm
Radio al eje del capstan (R_m)	95 mm
Radio máximo de cobertura en posición extendida (R_B)	181 mm
Longitud de los antebrazos del robot (L_F)	215 mm
Longitud de los brazos del robot.(L_A)	74 mm

Por otra parte, utilizando una báscula digital tipo gramera se obtuvieron las masas de los componentes para la compensación de gravedad del robot tipo Delta (Tabla 4.7).

Tabla 4.7: Masas de los componentes del robot tipo Delta.

RD.01-165 - Base móvil RD.01-170 - Soporte efector final RD.01-175 - Efector final	30 g
RD.01-160 - Eje superior para barras paralelas capstan y base móvil Rodamiento SKF 628-2z (2 pzas.)	6 g
RD.01-160 - Eje inferior para barras paralelas capstan y base móvil Rodamiento SKF 628-2z (2 pzas.)	6 g
Barras paralelas RD.01-155 - Articulación (2 pzas.) Rodamiento ZEN SMR 63-2z (4 pzas.)	44 g (22 g cada una)
RD.01-135 - Capstan Rodamiento SKF 628/2z (2 pzas.)	84 g cada uno

Después, mediante el script de Matlab® que se muestra en el Apéndice D se crearon las variables con las dimensiones del robot, para posteriormente cargarlas en el modelo Simulink® (***Cinematica_DeltaR***) utilizado para el desarrollo de la cinemática directa.

4.5.2. Validación de la cinemática directa

Recordando que la cinemática directa permite determinar la posición del EF a partir de las posiciones de las articulaciones (capstan) y, a su vez, de los motores mediante la relación de transmisión; se realizó la validación de la cinemática directa implementando el modelo Simulink® ***Cinematica_DeltaR*** donde previamente se cargaron las dimensiones del robot tipo Delta. Este modelo Simulink® lee la posición del motor y calcula las posiciones de los capstan y del EF, obtiene la matriz del Jacobiano inverso, del Jacobiano directo, la velocidad del motor y del EF. De manera que, mediante este modelo se guardaron las posiciones del robot tipo Delta en su configuración para trabajos verticales y a través de los scripts de Matlab® que se muestran en el Apéndice E se representaron los parámetros físicos para visualizar la posición totalmente retraída y totalmente extendida, ambas alineadas verticalmente con el eje Z (ver Figura 4.18); donde los brazos del robot se representan con líneas punteadas color azul, la línea punteada verticalmente de color negro representa la posición (0,0) en los ejes (***x,y***), la línea punteada horizontal color negro, representa la posición 0 grados del capstan (0 grados), la posición del efector final se representa con las iniciales **EF** y para representar cada capstan en el centro de ensamble entre el eje y las barras paralelas se utilizaron **C1, C2 y C3** respectivamente.

También, se imprimieron las posiciones en una terminal serial, donde se validó que el desplazamiento en el eje Z correspondiera al recorrido físico del capstan. Así mismo, en la terminal serial se imprimió la posición del capstan en grados y se validó que correspondiera con los 85° de desplazamiento positivo y 18° de desplazamiento negativo que muestran en el diseño CAD (Figura 4.19).

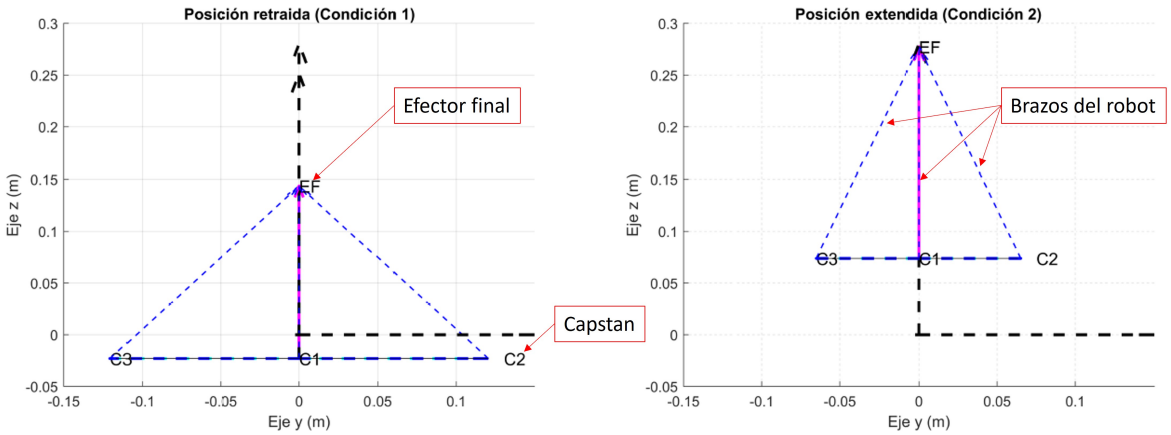


Figura 4.18: Representación de posición más retraída y extendida del robot tipo Delta.

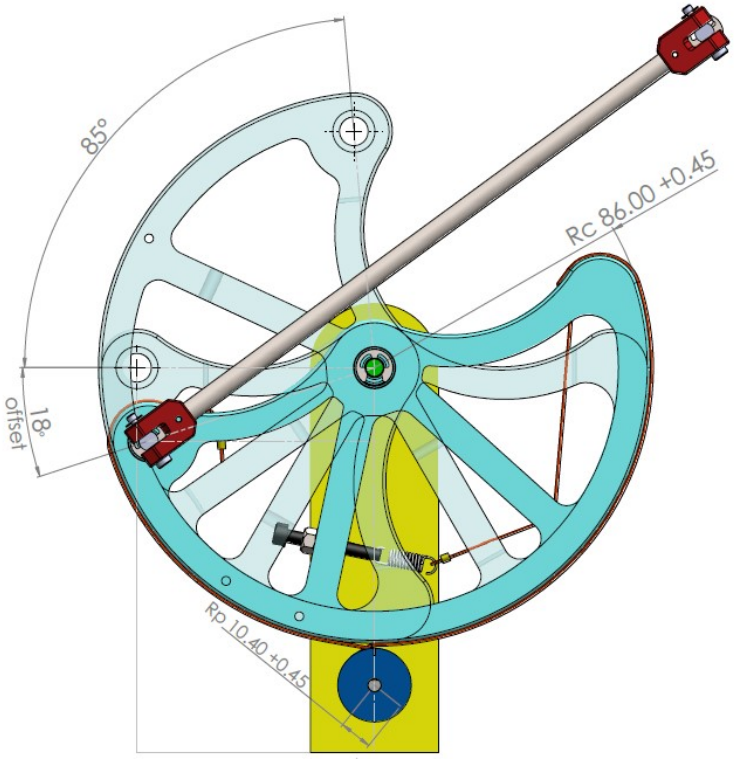


Figura 4.19: Simulación de los ángulos de desplazamiento del Capstan.

4.5.3. Cinemática Inversa

Considerando que el usuario entra en contacto principalmente con el EF, se desarrolló un modelo Simulink® que permita conocer la cinemática inversa del robot tipo Delta. En la cinemática inversa se envían las coordenadas (x,y,z) de la posición deseada del EF para obtener la posición de sus articulaciones (capstan), que dependen directamente de las poleas acopladas a los motores. Por lo tanto, con base en el modelo CLIK (Control Loop Inverse Kinematic) que se muestra en la Figura 4.20, se desarrolló la cinemática inversa; donde $\boldsymbol{K}$ representa una ganancia proporcional, $\dot{x}_d$ refiere a la velocidad deseada del EF, x_d es la posición deseada del EF, x_e es la posición actual del EF, e corresponde al error en la posición, $k(\cdot)$ corresponde a la matriz del modelo

geométrico directo, J_A^{-1} es la matriz del Jacobiano analítico inverso, $\dot{q}$ es la velocidad de la articulación que al integrarse se obtiene la posición de la articulación (q).

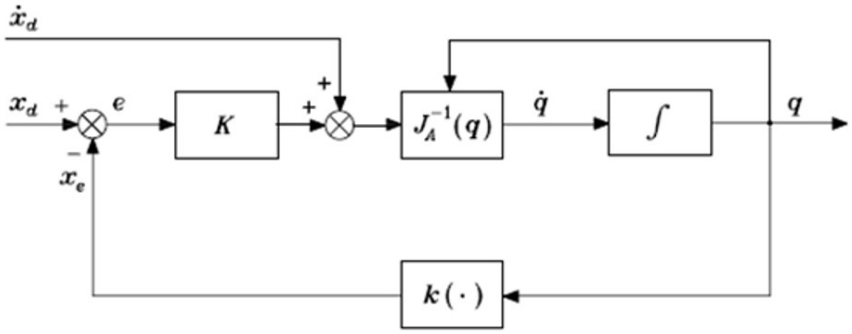


Figura 4.20: Modelo CLIK, Siciliano [14].

Se realizó la simulación del modelo CLIK en Simulink® validando su funcionamiento y determinando algunas modificaciones para implementarlo en la cinemática inversa de este robot. De esta forma, el bloque de la integral que se muestra en el modelo CLIK es descartado debido a que se consideró utilizar un control de velocidad en donde se hace la conversión de torque a corriente y mediante el mapeo de esta señal, se obtiene la posición deseada del EF del robot (ver Figura 4.21).

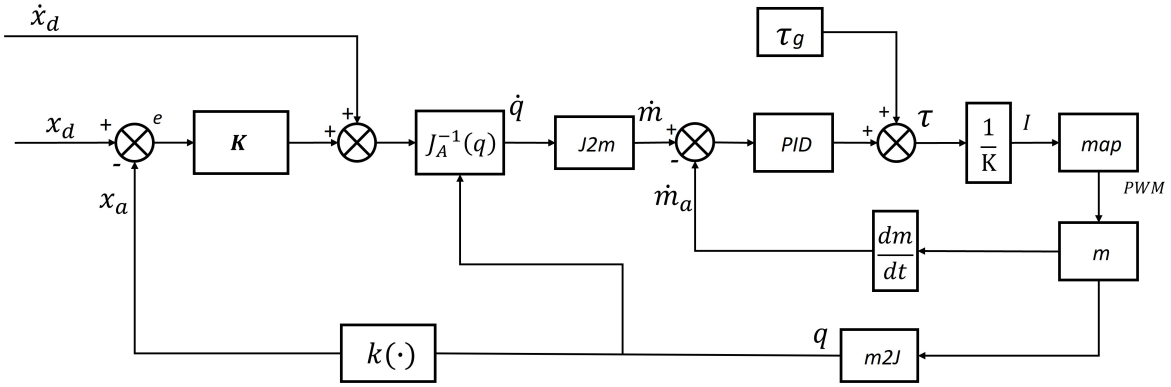


Figura 4.21: Descripción detalla a bloques del CLIK.

Donde $\dot{x}_d$ es la velocidad deseada del EF, que no fue considerada para este caso, x_d es la posición deseada del EF, x_a es la posición actual del EF, e hace referencia al error en la posición, $\boldsymbol{K}$ representa una ganancia proporcional, $k(\cdot)$ es la matriz del modelo geométrico directo que se obtiene del modelo Simulink® ***Cinematica_DeltaR*** previamente implementado y validado, del mismo modelo se obtiene J_A^{-1} que es la matriz del Jacobiano analítico inverso, $\dot{q}$ es la velocidad de la articulación (capstan), el bloque *J2m* realiza la conversión de la velocidad de capstan a motor mediante relación de transmisión, $\dot{m}$ es la velocidad del motor, τ_g corresponde a la compensación por gravedad que obtenemos del modelo Simulink® ***Cinematica_DeltaR*** que para este modelo fue descartado, τ es el torque, K representa la constante del motor que divide al torque y da como resultado una señal en términos de corriente (I). Así también, la señal obtenida de la corriente entra a un bloque donde se realiza un mapeo de corriente a PWM para determinar la cantidad de PWM que se envía a los motores (ver líneas 547-629 del Apéndice F), de modo que m corresponde a la posición de los motores,

esta se envía a un bloque donde se deriva para retroalimentarla como velocidad actual de los motores ($\dot{m}_a$), Además, la posición de los motores se envía al bloque *m2J* para convertirla en posición de la articulación (q) y a su vez retroalimentar el bloque donde se realiza la multiplicación con J_A^{-1}.

Por otra parte, como consecuencia del desarrollo de los diferentes tipos de control para el robot tipo Delta, se crearon diferentes ***estados*** para controlar el funcionamiento del robot, en la siguiente sección se describe cada uno de los estados.

4.5.4. Estados de trabajo del robot tipo Delta

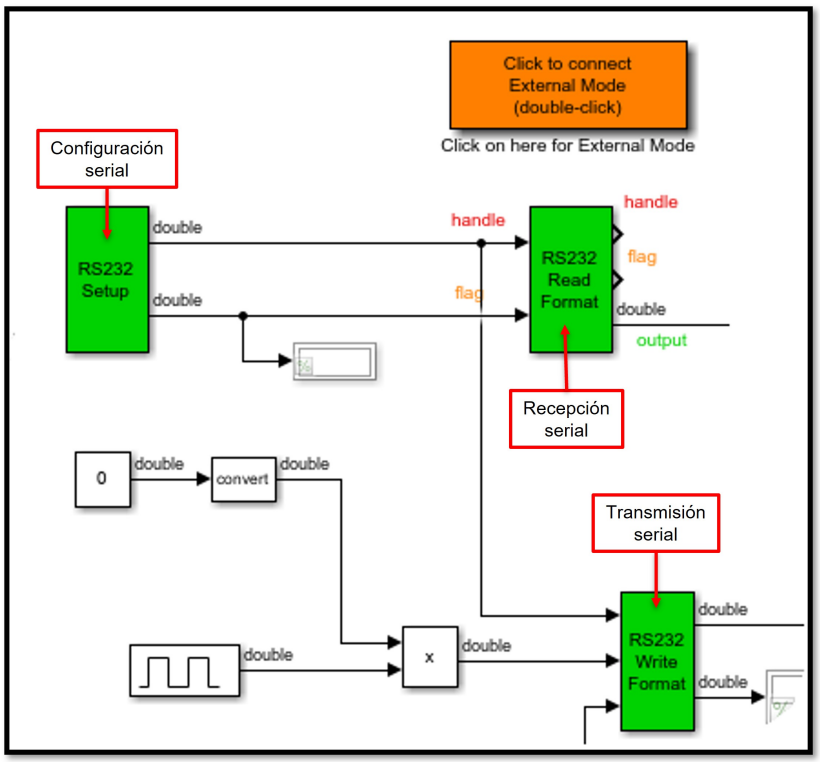
Se definieron 4 estados para las diferentes tareas que puede realizar el robot tipo Delta y que debe ser seleccionado por el usuario. Las tareas específicas que se realizan en cada estado del robot tipo Delta se describen a continuación:

- **Estado 0:** Estado en el que el robot se puede mover libremente sin presentar ninguna resistencia.
- **Estado 1:** Este estado corresponde a la calibración del robot para la posición *Home*, es decir, el robot es dirigido a su posición más retraída donde se reinicia la posición de los motores a cero grados. Cabe mencionar que este estado es importante que se active antes de iniciar los estados 2 o 3 para evitar alguna colisión.
- **Estado 2:** Este estado permite realizar el control del robot mediante la posición de los motores. Es decir, se introducen las posiciones deseadas de los motores y mediante la cinemática directa, se determina la posición del EF.
- **Estado 3:** Este estado permite realizar el control del robot mediante la cinemática inversa. Es decir, en este estado se ingresan las coordenadas (x,y,z) donde se desea que se posicione el efector final.

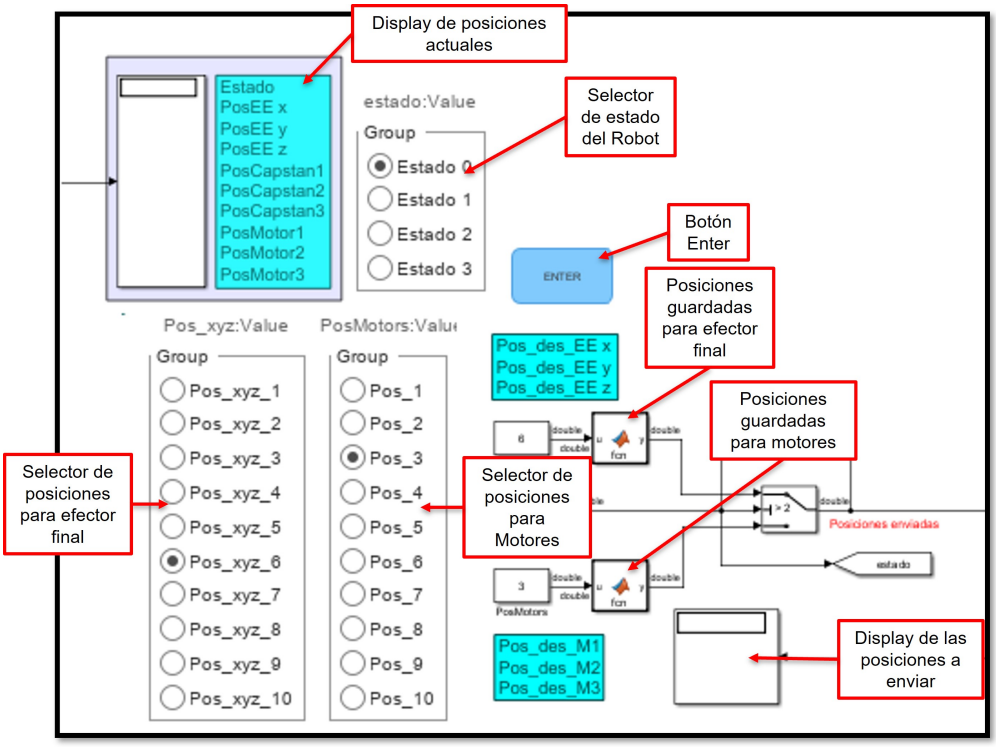
Por otro lado, se creó una interfaz para el usuario a partir de bloques de Simulink® donde se envían las posiciones deseadas de los motores y del EF según el estado seleccionado. En esta interfaz se pueden visualizar las posiciones actuales de los motores, de los capstan y del EF.

4.5.5. Interfaz para el usuario

La interfaz para el usuario permite seleccionar el estado con el cual se desea manipular el robot tipo Delta. Por ejemplo, en el estado 2 es posible enviar las posiciones en grados para los motores y en el estado 3 se ingresan las coordenadas (x,y,z) del efector final. La Figura 4.22a resalta los bloques para la configuración serial tanto en recepción como en transmisión. En la Figura 4.22b se muestra la sección de la interfaz que permite la selección del estado de control del robot, los selectores de la posición deseada, displays con las posiciones actuales de los motores, capstan y coordenadas del EF. En el Apéndice G se presenta el modelo completo en Simulink® de la interfaz del usuario.



(a)



(b)

Figura 4.22: (a) Configuración serial de la interfaz en recepción y transmisión. (b) Displays y selectores.

Capítulo 5

Resultados y discusión

Es importante destacar que las pruebas de posición, repetibilidad y precisión del robot se llevaron a cabo únicamente en la configuración para trabajos verticales. En cuanto a las pruebas de carga, se tomaron en cuenta ambas configuraciones mecánicas del robot, tanto para trabajos verticales como horizontales, ya que resulta relevante identificar la capacidad de carga que el EF puede soportar en distintas direcciones.

Pruebas de carga, posición, repetibilidad, precisión y espacio de trabajo

Las pruebas de carga en la configuración para trabajos verticales se validó la carga máxima que soporta el robot en su EF en una posición central alineada verticalmente y considerando que las cargas se encuentran distribuidas equitativamente entre los tres motores. Por otro lado, en las pruebas de carga para la configuración de trabajos horizontales se realizaron 3 pruebas diferentes, enviando el EF a una posición totalmente retraída, una posición intermedia y una posición extendida; en esas posiciones se les aplicó una carga al EF en dirección positiva de la gravedad.

Las pruebas de posición se realizaron para los estados 2 y 3, donde se envía la posición de los motores y las coordenadas del EF respectivamente, se guardaron las posiciones alcanzadas por los motores y el EF, de manera que se obtuvo el error en cada posición y se calculó el error promedio para cada estado.

Las pruebas de repetibilidad y precisión solo se realizaron para el estado 3. En las pruebas de repetibilidad, se dirigió el EF hacia el mismo punto bajo condiciones idénticas de carga y temperatura, calculando la repetibilidad del robot a través de la desviación estándar. Para las pruebas de precisión, se envió el EF a una posición predefinida, repitiendo este desplazamiento varias veces bajo carga y temperaturas nominales. Utilizando el error medio cuadrático (RMS) para calcular la precisión del robot.

Por otro lado, para determinar el espacio de trabajo se validaron las fronteras del robot en cada eje y se representaron utilizando Matlab®.

5.1. Pruebas de carga

Las pruebas de carga en la configuración para trabajos verticales se iniciaron colocando una carga equivalente a 15 N (1.5296 kg) sobre el EF en una posición alineada verticalmente con el eje Z (Figura 5.1), después se realizaron con el EF del robot en

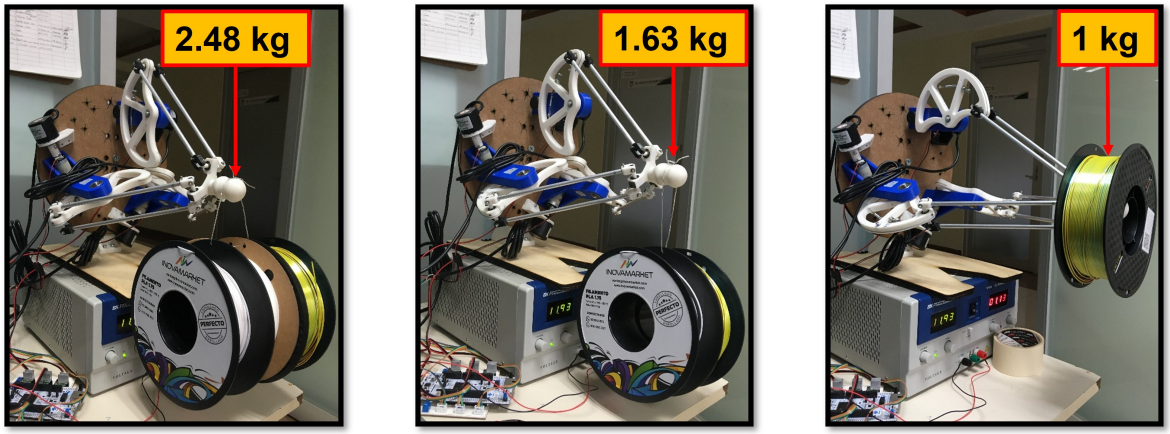
posiciones aleatorias dentro de su espacio de trabajo, donde se validó que el robot soporta carga establecida. También, en una posición alineada verticalmente se aumentó el peso hasta obtener la carga máxima que soporta el robot en su EF (ver Figura 5.1a), se establecieron 28 N (2.87 kg) como el máximo de carga que soporta el robot sin riesgo de dañar los motores debido a que estos motores soportan una corriente máxima de 2.1 A y el consumo de corriente por los 3 motores para esa carga fue de 5.9 A, considerando que el EF se encuentra alineado verticalmente y suponiendo que la carga está distribuida de igual forma entre los motores por la configuración triangular de los robots tipo Delta, esta seria la corriente máxima que soportarían los motores para evitar que se dañen.

Adicionalmente, se realizó una prueba donde se identificó la carga máxima que puede levantar el robot en su EF al desplazarse para llegar a una nueva posición (Figura 5.1b), logrando desplazarse con una carga en su EF de 11 N (aproximadamente 1.18 kg).

(a) (b)

Figura 5.1: Pruebas de carga en configuración vertical. (a) Carga máxima soportada. (b) Carga máxima soportada por el robot al desplazarse a una posición.

Las pruebas de carga en la configuración para trabajos horizontales se realizaron en la posición más retraída del efector final (0,0,0.1437) m, soportando una carga de 2.48 kg con una corriente de 1.52 A por los 3 motores (Figura 5.2a), en la posición intermedia del efector final (0,0,0.182) m soportó una carga de 1.63 kg con un consumo de corriente por los 3 motores de 0.5 A (Figura 5.2b) y para la posición más extendida durante las pruebas, el efector final se localizaba en la posición (0,0,0.251) m, soportando una carga de 1.06 kg con una corriente de 1.3 A por los 3 motores (Figura 5.2c). Por lo tanto, se puede concluir que la carga máxima que el robot puede soportar en su EF para configuraciones de trabajo horizontal depende principalmente de la posición del EF. Teniendo en cuenta que la corriente máxima que un motor puede soportar es de 2.1 A y basándose en la lectura de corriente de las pruebas, el consumo de corriente de los motores no representaría ningún riesgo de daño para los mismos. Esto es así incluso considerando que en esta configuración la distribución de la carga no es la misma para los tres motores.



(a) (b) (c)

Figura 5.2: Pruebas de carga en configuración horizontal. (a) Pruebas en posición retraída. (b) Pruebas en posición intermedia. (c) Pruebas en posición extendida.

5.2. Pruebas de posición

Las pruebas de posición se realizaron para los estados 2 (control de posición de los motores) y 3 (control de posición del EF). La validación de las posiciones se realizó de la siguiente manera: se capturaron 15 posiciones conocidas por cada estado, que representan las posiciones deseadas que envía el usuario, posteriormente se compararon con las posiciones alcanzadas, obtenido el error promedio para ambos estados. Cabe mencionar que se utilizó una interfaz de usuario en Simulink® la cual envía y recibe datos seriales, y guarda los datos deseados en tiempo real. A continuación, se explica de forma más detallada las pruebas de ambos estados.

En el estado 2, se envía la posición deseada de los motores y mediante la interfaz de usuario en Simulink® se guardan los valores y se generan gráficas para 3 posiciones diferentes. Para la primera posición que se muestra en la parte superior de la Figura 5.3 se enviaron los 3 motores a la misma posición, es decir, M1 = M2 = M3 = 144°. En la gráfica central de la Figura 5.3 se presenta la segunda prueba, realizada a partir de las posiciones anteriores, de manera que los 3 motores se encuentran en la posición de 144° y se envían a las siguientes posiciones: M1 = 728°, M2 = 85° y M3 = 292°. Para la tercera prueba (ver gráfica inferior de la Figura 5.3), se enviaron las siguientes posiciones: M1 = 715°, M2 = 314° y M3 = 36°. En cada prueba se muestra la posición deseada contra la posición alcanzada por cada motor.

Estas pruebas continuaron para un total de 15 posiciones diferentes, todas iniciando desde su última ubicación alcanzada, cada posición se mantuvo un tiempo aproximado de 8 segundos hasta enviar la siguiente posición, obteniendo el error promedio de cada motor que se presenta en la Tabla 5.1.

Tabla 5.1: Error promedio de los motores para 15 posiciones diferentes.

	Error promedio
Motor 1	1.572 grados
Motor 2	1.226 grados
Motor 3	1.092 grados

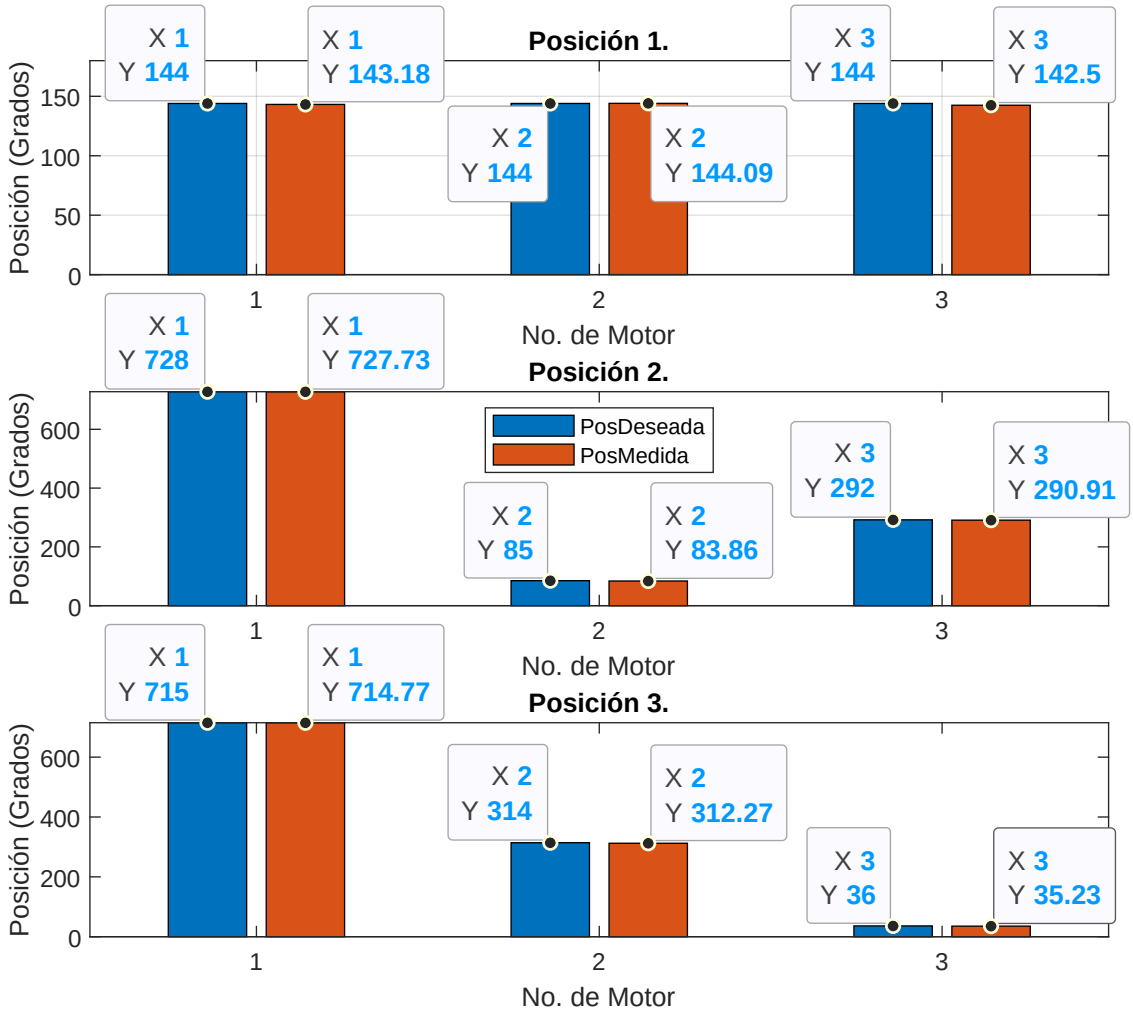


Figura 5.3: Pruebas de posición en el estado 2 del robot tipo Delta.

Para el estado 3 se realizaron las pruebas enviando las coordenadas (x,y,z) del EF y mediante la cinemática diferencial se obtienen las velocidades deseadas de las articulaciones que, a su vez son transformadas al torque deseado para controlar la posición de los motores. De manera que en la Figura 5.4 se representan las posiciones enviadas de la Tabla 5.2.

Tabla 5.2: Posiciones enviadas en el estado 3 al EF.

	Eje X (m)	Eje Y (m)	Eje Z (m)
Posición 1	0.0.541	0.0588	0.1942
Posición 2	0	0	0.26
Posición 3	-0.1063	-0.0645	0.1846

Las pruebas continuaron para un total de 15 posiciones diferentes. Por lo tanto, el error promedio de los ejes (x,y,z) del EF se muestran en la Tabla 5.3

Adicionalmente, se agrega un enlace con las pruebas de posición para el estado 2 y 3 en la configuración para trabajos verticales, en estas pruebas se puede visualizar el funcionamiento del robot y el uso de la interfaz del usuario en Simulink®, donde

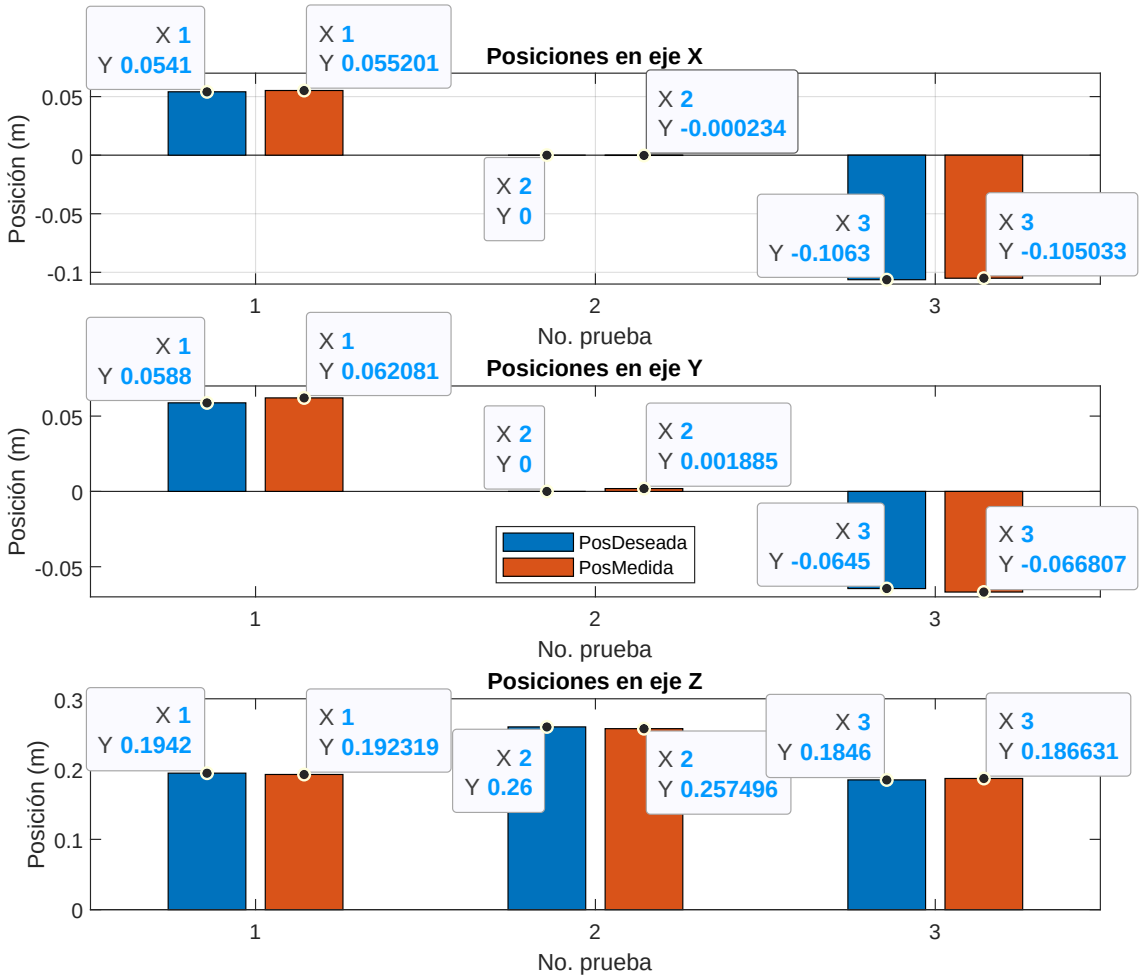


Figura 5.4: Pruebas de posición en estado 3 del robot tipo Delta.

Tabla 5.3: Error promedio en ejes del EF para 15 posiciones diferentes

	Error promedio
Eje x	0.0013 m
Eje y	0.0017 m
Eje z	0.0025 m

se realiza la selección de estado del robot, la selección de posiciones para motores y EF respectivamente, se muestra un display con las posiciones a enviar y uno con las posiciones actuales del EF, motores y capstan (video [video_pruebas_3]).

5.2.1. Pruebas de precisión y repetibilidad.

Como se mencionó anteriormente, estas pruebas se realizaron solo en el estado 3, se enviaron las siguientes coordenadas del EF y se repitieron en 10 ocasiones bajo las mismas condiciones: x = 0, y = 0, z = 0.25 m.

En la Tabla 5.4 se presentan las posiciones registradas del EF para cada prueba, obteniendo un error promedio en el eje $x = 0.0009$ m, en $y = 0.0019$ m, en $z = 0.0025$ m.

Tabla 5.4: Posiciones alcanzadas por el efector final del robot tipo Delta en (x,y,z).

	Eje X (m)	**Eje Y (m)**	**Eje Z (m)**
Pos deseada	**0**	**0**	**0.25**
Pos. prueba 1	-0.0005	0.0031	0.2471
Pos. prueba 2	-0.0004	0.0032	0.2471
Pos. prueba 3	-0.0005	0.0030	0.2470
Pos. prueba 4	0.0017	-0.0014	0.2482
Pos. prueba 5	0.0006	-0.0004	0.2476
Pos. prueba 6	0.0006	-0.0003	0.2475
Pos. prueba 7	0.0006	-0.0003	0.2475
Pos. prueba 8	0.0007	-0.0003	0.2474
Pos. prueba 9	0.0029	0.0038	0.2485
Pos. prueba 10	-0.0005	0.0034	0.2470

En la Figura 5.5 se representan las posiciones del EF en los ejes XY de cada prueba realizada. En esta misma Figura se muestra un zoom de las posiciones alcanzadas donde se aprecia que la posición más lejana donde se posicionó el EF muestra una separación de la posición deseada de aproximadamente 0.003 m en el eje X y de 0.004 m en el eje Y.

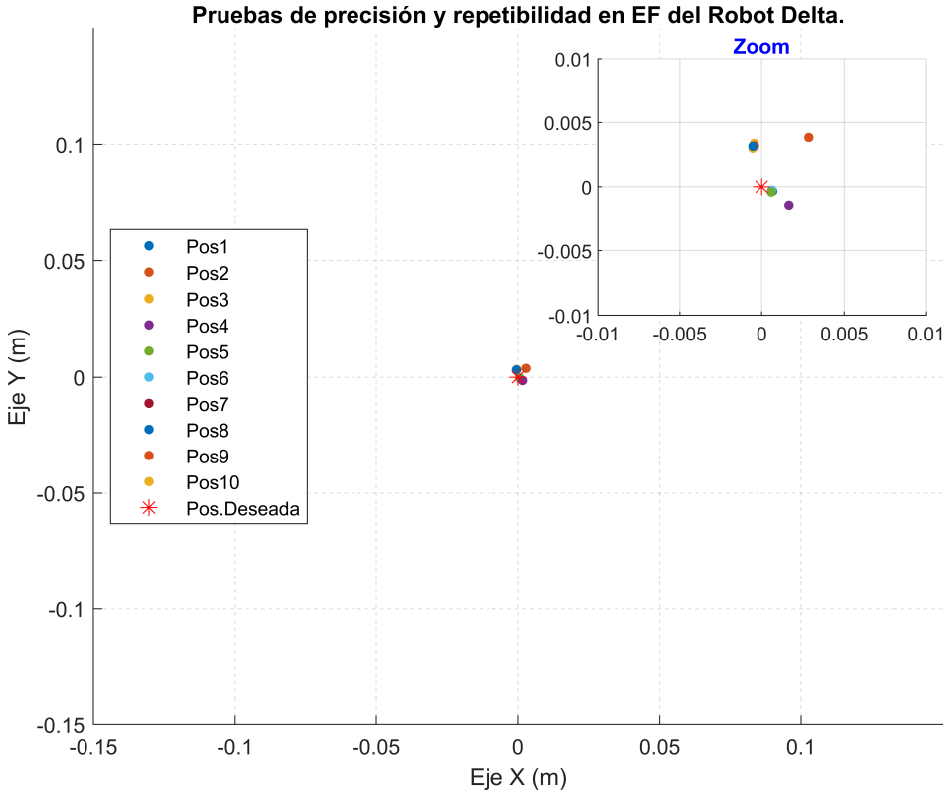


Figura 5.5: Pruebas de precisión y repetibilidad en ejes XY.

La Figura 5.6 muestra las posiciones alcanzadas en cada prueba por el EF desde los ejes XZ. En la misma Figura se agrega un zoom donde observa que la posición que presenta más diferencia de la posición deseada en el eje z está a 0.003 m de distancia.

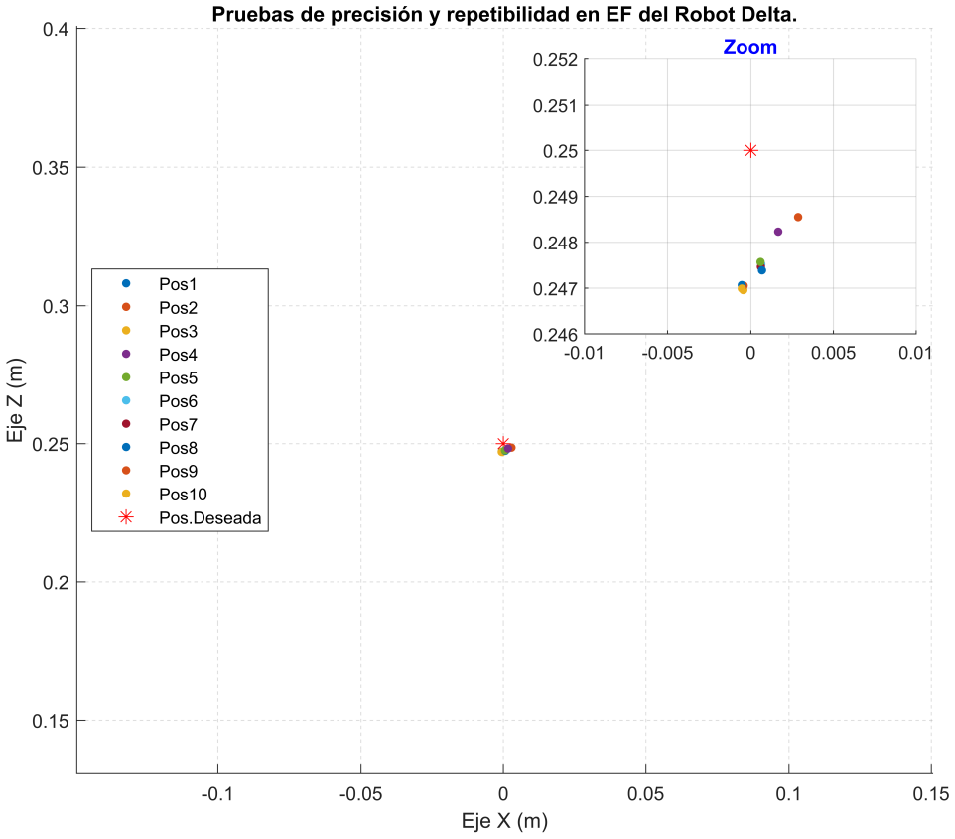


Figura 5.6: Pruebas de precisión y repetibilidad en ejes XZ.

Por lo tanto, con base en lo descrito en la Sección 2.1.4, se calculó la precisión y repetibilidad de este robot tipo Delta.

Precisión (P).

Sustituyendo los valores en la Ecuación 5.1 de las posiciones alcanzadas y las posiciones deseadas que se muestran en la Tabla 5.4, se calculó la precisión del robot tipo Delta

$$P = \sqrt{{e_x}^2 + {e_y}^2 + {e_z}^2} \tag{5.1}$$

donde:

$$e_x = \sqrt{\frac{\sum_n (x_d - x_i)^2}{n}};\ e_y = \sqrt{\frac{\sum_n (y_d - y_i)^2}{n}};\ e_z = \sqrt{\frac{\sum_n (z_d - z_i)^2}{n}}$$

n = número de pruebas
Dando como resultado una precisión del robot de $\boldsymbol{P} = \textit{0.0037 m}$

Repetibilidad (R).

Por otro lado, al sustituir los valores de las posiciones alcanzadas y las posiciones deseadas de la Tabla 5.4 en la Ecuación 5.2, se calculó la repetibilidad del robot tipo Delta

$$R = \sqrt{{\sigma_x}^2 + {\sigma_y}^2 + {\sigma_z}^2} \tag{5.2}$$

donde:

$$\sigma_x = \frac{\sum_n (x_i - x_m)^2}{n};\ \sigma_y = \frac{\sum_n (y_i - y_m)^2}{n};\ \sigma_z = \frac{\sum_n (z_i - z_m)^2}{n} \tag{5.3}$$

$$x_m = \frac{\sum_n x_i}{n}; y_m = \frac{\sum_n y_i}{n}; z_m = \frac{\sum_n z_i}{n}; \tag{5.4}$$

n = número de pruebas
Obteniendo una repetibilidad del robot de: $\boldsymbol{R} = 0.0024\ m$

En resumen, las pruebas de carga, posición, repetibilidad y precisión realizadas a este robot tipo Delta permiten visualizar respuestas similares a las de los robots comerciales, a pesar de que este fue fabricado en su mayoría por impresión 3D.

5.3. Espacio de trabajo.

Por otro lado, mediante un modelo Simulink® se guardaron algunos movimientos realizados manualmente en el estado 0 del robot para visualizar su espacio de trabajo. Posteriormente, a través de Matlab® se generó una gráfica para representar los puntos en los que puede desplazarse (Figura 5.7).

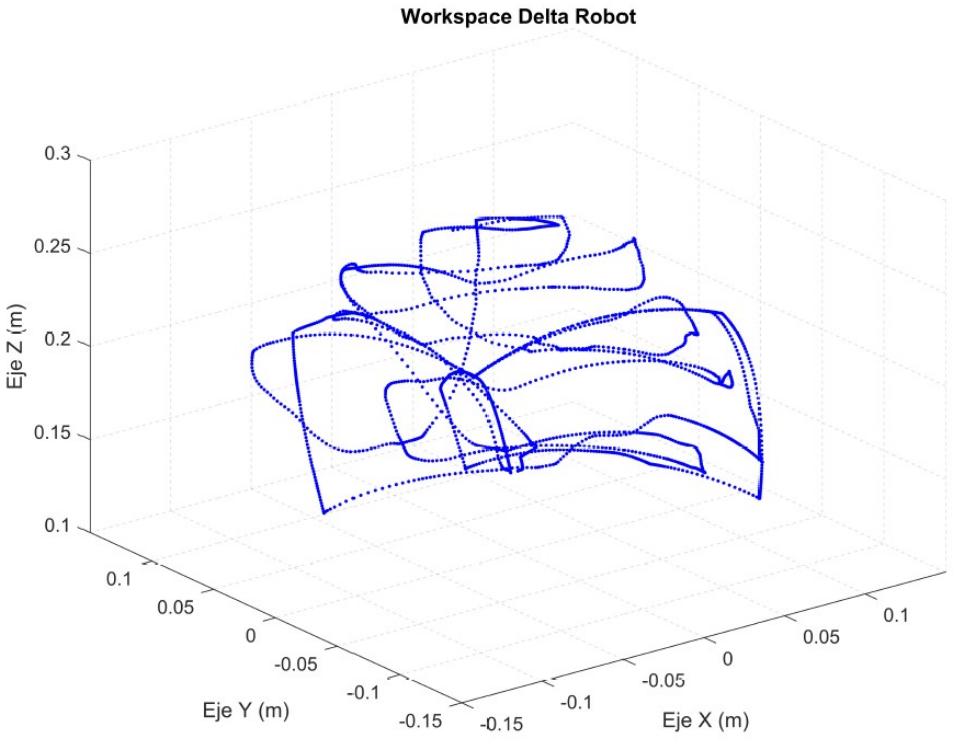
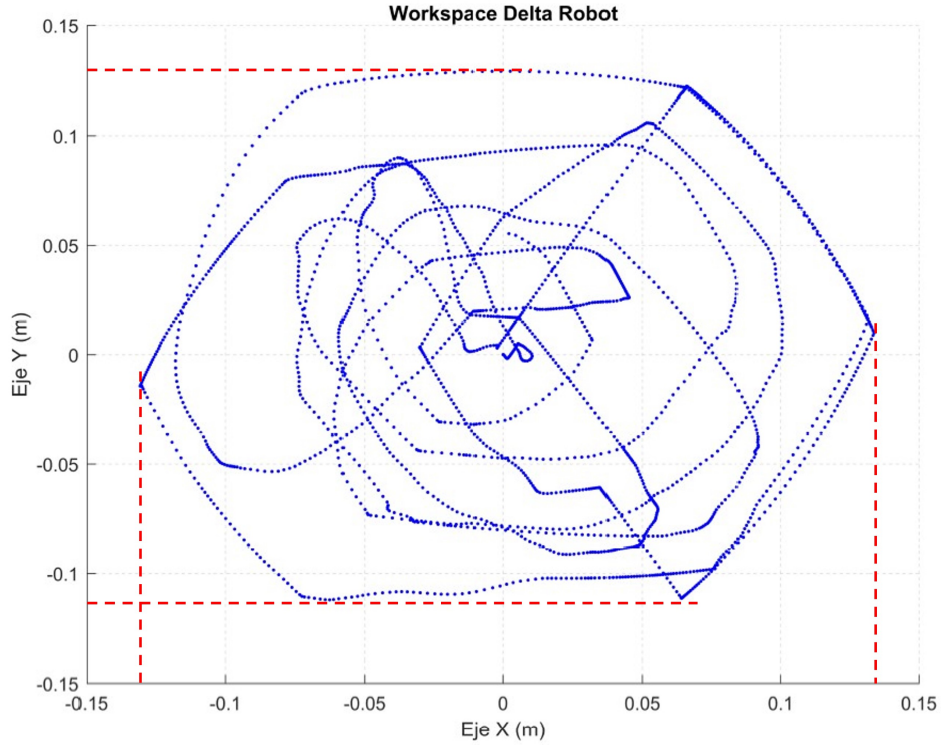
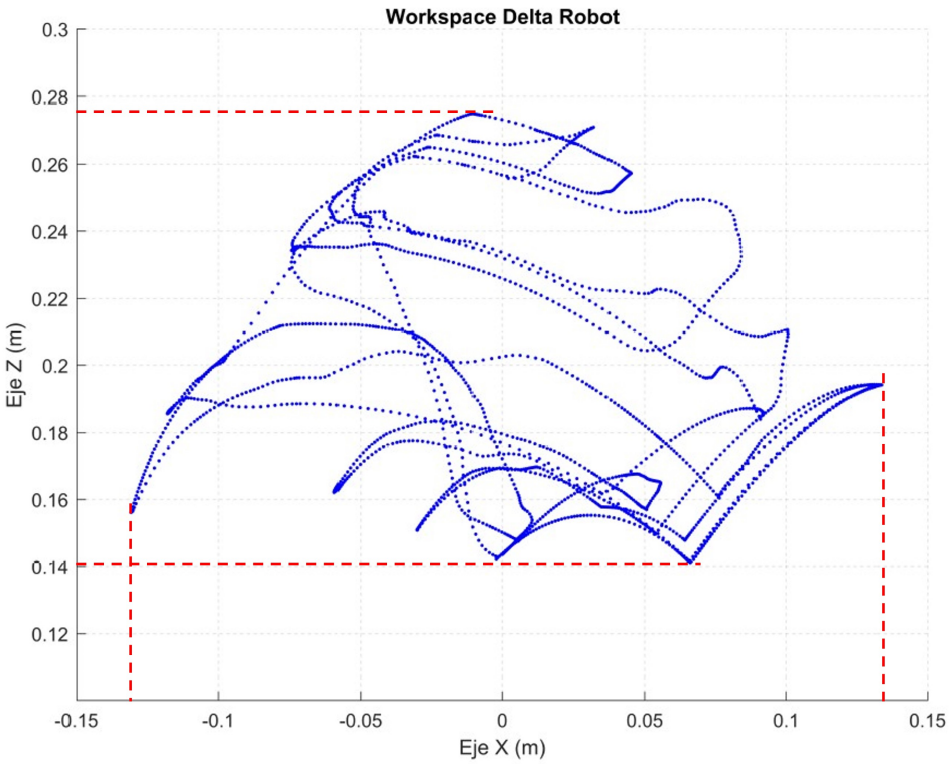


Figura 5.7: Espacio de trabajo en XYZ.

Considerando las fronteras que se muestran en los ejes del plano XY y XZ de la Figura 5.8, se define su espacio de trabajo del robot tipo Delta de **260 mm x 260 mm x 135 mm**.



(a)



(b)

Figura 5.8: (a) Espacio de trabajo en plano XY. (b) Espacio de trabajo en plano XZ.

5.4. Discusión

El desarrollo de este proyecto de rediseño del robot tipo Delta mecánicamente reconfigurable ha logrado importantes avances en términos de flexibilidad y adaptabilidad. Sin embargo, es importante destacar tanto los alcances como las limitaciones encontradas durante el proceso de investigación y desarrollo.

En cuanto a los alcances, se pudo diseñar y fabricar con éxito los componentes mecánicos necesarios para la reconfiguración del robot, lo que permitió ampliar su espacio de trabajo y mejorar su adaptabilidad a diferentes tareas. Además, se logró implementar un sistema eléctrico y de control eficiente, que permitió un control preciso y coordinado de los motores del robot. Esto se reflejó en los resultados obtenidos, donde se demostró un incremento significativo en la carga soportada por el EF.

La Tabla 5.5 muestra los gastos del material utilizado para la construcción del robot tipo Delta. Considerando que los datos necesarios para construir el robot se han liberado al público en general, el desarrollo presentado en esta tesis representa una opción accesible para investigadores, universidades y laboratorios para su fabricación e implementación en diversas aplicaciones.

Tabla 5.5: Gastos en material para construcción de robot tipo Delta.

Material	**Cantidad utilizada**	**Precio (MXP)**
Rodamientos diam. ext = 13 mm, diam. int. = 6 mm	18 pzas	1134.27
Rodamientos diam. ext. = 22 mm, diam. int. = 7 mm	6 pzas	268.6
Rodamientos diam. ext. = 6 mm, diam. int. = 3 mm	24 pzas	991.8
Tuercas y tornillos	14 pzas c/u	142.18
Motor Pololu #4861 con driver	3 pzas	3951.93
Rollo de PLA	1 pza	200
Encoders	3 pzas	1200
Tarjeta núcleo STM32-F767ZI	1 pza	700
Barras de aluminio de 1/4" diámetro	6 x 20 cm	156
Barra de latón 4mm de diámetro	6 x 3 cm	5.583
Hoja de MDF	1 pza	102.6
Hoja triplay	1 pza	50
Componentes de PCB	1 pza	30
Pernos 3mm diámetro	12 x 19 mm	29.64
Resortes	3 pzas	3.93
Seguros	30 pzas	38.85
Cable de acero 9 mm de diámetro	1.2 m	2.8
Gasto total		**9008.18 MXP**

Por otro lado, es importante reconocer las limitaciones encontradas durante el proyecto. Considerando que el presente proyecto se centró principalmente en el aspecto mecánico y de control del robot, dejando de lado otros aspectos como la interacción con el entorno. Estos aspectos pueden ser considerados en futuras investigaciones para

ampliar aún más las capacidades y aplicaciones del robot.

Además, es necesario mencionar que si bien se obtuvieron resultados satisfactorios en entornos controlados de laboratorio, es fundamental realizar pruebas y validaciones en entornos reales y aplicaciones específicas para evaluar plenamente el desempeño del robot en diferentes contextos y condiciones.

En la Tabla 5.6 se muestra una comparativa del robot desarrollado contra algunos robots comerciales, donde se puede apreciar que el dispositivo compite con las características principales de los robots comerciales, aun considerando que este dispositivo es fabricado en su mayoría por impresión 3D, ofreciendo una alternativa considerable para investigadores, laboratorios y universidades.

Tabla 5.6: Comparación de características del robot desarrollado en este proyecto contra robots comerciales.

	Novint Falcon	***Omega.3***	***Delta.3***	***Robot Delta***
Costo	499 USD	22,879.57 USD	34,858.97 USD	9008.18 MXP
Reconfigurable mecánicamente	No	No	Si	Si
Fuerza	8.8 N	12 N	20 N	16 N
GDL	3	3	3	3
Resolución en efector final	1 mm	< 0.01 mm	< 0.02 mm	1-2 mm
Espacio de trabajo	101.6 x 101.6 x 101.6 mm	Traslación 160 x 160 x 110 mm	Traslación 400 x 400 x 260 mm	260 x 260 x 135 mm
Permite modificaciones en la cinemática del robot	No	Si (costo adicional)	Si (costo adicional)	Si
Permite agregar diferentes efectores finales	No	Si (costo adicional)	Si (costo adicional)	Si
Tipo de implementación	Háptica, teleoperación, ambientes virtuales	Háptica, teleoperación, ambientes virtuales	Háptica, teleoperación, ambientes virtuales	Háptica, telerobótica, ambientes virtuales

En resumen, este proyecto de rediseño del robot tipo Delta mecánicamente reconfigurable ha logrado importantes avances en términos de flexibilidad y adaptabilidad. Si bien existen limitaciones y áreas de mejora, los resultados obtenidos sientan las bases para futuras investigaciones y mejoras en el campo de la robótica. La combinación de los alcances logrados y las limitaciones nos brinda una visión clara de las posibilidades y desafíos que aún se presentan en la búsqueda de robots más versátiles y eficientes.

Capítulo 6

Conclusiones y trabajo futuro

En este proyecto de tesis se planteó el objetivo de rediseñar la mecánica de un robot tipo Delta, permitiendo una reconfiguración mecánica que brinde la posibilidad de realizar trabajos tanto en posición vertical como en horizontal, que permita la modificación y actualización de los parámetros del robot para obtener diferentes espacios de trabajo que permitan implementarlo en múltiples aplicaciones de teleróbotica y ambientes virtuales. Por lo que es posible determinar que el robot cumple con el objetivo, el diseño permite una reconfiguración mecánica de manera simple que puede ser realizado por cualquier persona con un nivel de licenciatura.

El diseño de componentes en modelo CAD se realizó en el programa Solidworks®, donde se realizaron las simulaciones para representar los movimientos del robot tipo Delta y la interacción entre los componentes que realizan la transmisión por cable. En el Apéndice A se muestran los diagramas mecánicos 2D de cada uno de los componentes del robot, lo que garantiza y cumple con el libre acceso al diseño mecánico del robot para su fabricación o modificación de componentes que agreguen diferentes características al robot.

El análisis de elemento finito permitió considerar las propiedades físicas de los polímeros PLA y ABS. Cabe mencionar que el análisis demostró que el diseño y ambos materiales soportan sin problema la carga de 15 N definida en los requerimientos de diseño. No obstante, se realizaron pruebas físicas que demostraron que la calidad y resistencia de componentes pequeños con forma cilíndrica y fabricados por impresión 3D (RD.01-130-perno, RD.01-150-perno y RD.01-160-eje) no corresponden con la simulada en el software, debido a que el software considera una pieza completamente solida y basta con que alguna de las capas que forman una pieza por impresión 3D no se adhiera correctamente, para generar una zona frágil que no es considerada en el análisis. Además, debido a la longitud y forma cilíndrica de los componentes, no fue posible imprimir componentes totalmente uniformes, por lo que se determinó fabricarlos de aluminio 6061.

Se desarrolló e implementó el sistema eléctrico para el control de los actuadores. El esquemático de las conexiones se ilustra en el Apéndice B. Es importante mencionar que sería conveniente considerar una mejor distribución del cableado para los motores y los encoders, así como la fabricación de una caja protectora para la PCB y la tarjeta de desarrollo STM32-NUCLEO F767ZI.

La programación de la cinemática en un sistema embebido se cumplió de buena manera, se utilizó el método RCP que permitió reducir tiempos de programación y realizar el control mediante bloques en un modelo Simulink®. Posteriormente, este modelo se

conectó con el programa STM32CubeIDE de la tarjeta STM32 NUCLEO F767ZI donde se configuró el microcontrolador y se generó el código C para la comunicación.

Por otra parte, el diseño del EF no está limitado al modelo propuesto en este proyecto, es decir, cada investigador puede desarrollar su propio EF de acuerdo a su aplicación y ensamblarlo al robot tipo Delta, después podrá modificar el script con las dimensiones y peso del mismo en Matlab® que se muestra en el Apéndice D, para que este sea considerado en la cinemática del robot. Cumpliendo el objetivo de agregar diferentes efectores finales al mecanismo de la base móvil del robot tipo Delta.

Con respecto al espacio de trabajo del robot, se propuso aumentar al promedio de los robots comerciales y no comerciales, considerando que este espacio de trabajo es suficiente para implementarse en las aplicaciones descritas en el estado del arte (Sección 2.2), superando el espacio de trabajo en los ejes (x,y) y quedando cerca en el eje (z) debido a los limites mecánicos implementados para evitar que el robot entrara en zonas donde colisionaba. El espacio de trabajo de este robot tipo Delta es de 260 x 260 x 135 mm Cabe resaltar que al ser hardware libre, queda abierta la posibilidad de modificar las dimensiones de los componentes si es requerido aumentar el espacio de trabajo.

En las pruebas de carga, el robot fue capaz de soportar 28 N en su EF en la configuración para trabajos verticales, considerando que los actuadores del robot se encontraban en la misma posición para mantener alineado al centro del eje Z del EF. Además, logró desplazarse de una posición a otra cargando 11.8 N. Con respecto a la configuración para trabajos horizontales, soporta una carga en cualquier dirección y en todo su espacio de trabajo de 10.6 N. Alcanzando una carga de 16 N si la carga es positiva en dirección de la gravedad y el EF no supera la posición media de su desplazamiento en el eje Z. Considerando que la carga soportada en el EF cumple con el objetivo, ya que esta carga es suficiente para ser implementado en las aplicaciones mencionadas en el estado del arte (Sección 2.2),

Por otro lado, en las pruebas experimentales de los movimientos del robot, mostraron un error en la posición para el estado 2 de 1.296 grados y para el estado 3 de 1.8 mm. En cuanto a la precisión y repetibilidad, a pesar de estar fabricado en su mayoría por impresión 3D, en las pruebas realizadas presentó una precisión de *3.7 mm* y una repetibilidad de *2.4 mm*.

En conclusión, esta tesis ha logrado abordar de manera efectiva el problema del rediseño de un robot tipo Delta mecánicamente reconfigurable, cumpliendo con la meta de ingeniería y los objetivos planteados, mejorando su flexibilidad y adaptabilidad en diversas aplicaciones. Los resultados obtenidos demuestran un avance significativo en términos de espacio de trabajo, carga soportada, precisión y repetibilidad del robot, contribuyendo al desarrollo de robots más versátiles y eficientes.

6.1. Trabajo futuro

Como trabajo futuro sería interesante implementarlo en una aplicación de teleróbotica o ambientes virtuales, tanto en su configuración para trabajos horizontales como para trabajos verticales, así se podría evaluar su funcionamiento en escenarios reales para realizar una comparación más profunda con los robots comerciales. Además, esto permite tener una retroalimentación de acuerdo a la experiencia del usuario y poder considerar mejoras.

Además, se considera la redacción de un artículo en revista indexada donde se proponga el desarrollo de un plataforma de acceso libre para fabricación de un robot

tipo Delta para educación, donde se motive a estudiantes a desarrollar los conceptos de electrónica, programación, robótica, cinemática directa/inversa, control de posición en motores.

Referencias

[1] Á. García Álvarez, *Optimización de manipuladores paralelos para una región de trabajo objetivo mediante algoritmos genéticos*. PhD thesis, ETSI_Informatica, 2020.

[2] O. Corporation., "Omron." `https://www.ia.omron.com/products/category/robotics/industrial-robots/parallel-robots/index.html`, 2016.

[3] J.-y. Lee and S. Payandeh, *Haptic Teleoperation Systems*. Springer, 2015.

[4] Intuitive, "Da-vinci." `https://www.intuitive.com/en-us/products-and-services/da-vinci`, 2021.

[5] O. Khatib and B. Siciliano, *Springer handbook of robotics*. Springer International Publishing, 2016.

[6] J. Moo-Young, T. M. Weber, B. Kapralos, A. Quevedo, and F. Alam, "Development of unity simulator for epidural insertion training for replacing current lumbar puncture simulators," *Cureus*, vol. 13, no. 2, 2021.

[7] F. Dimension, "Haptic device." `https://www.forcedimension.com/products`, 2001.

[8] A. Barrientos *et al.*, *Fundamentos de robótica*. Biblioteca Hernán Malo González, 2007.

[9] G. L. Duffy, J. W. Moran, and W. Riley, *Quality function deployment and Lean Six Sigma applications in public health*. Quality Press, 2010.

[10] T. Howard, S. J. Culley, E. Dekoninck, *et al.*, "Creativity in the engineering design process," in *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007*, pp. 329–330, 2007.

[11] P. Brello, "Pla acido poliláctico." `https://plasticos-brello.com/material/pla-acido-polilactico/`, 2018.

[12] Ultimaker, "Ficha de datos técnicos pla." `https://docs.rs-online.com/7ade/0900766b81698006.pdf`, 2017.

[13] C. A. Avizzano, A. Filippeschi, J. M. J. Villegas, and E. Ruffaldi, "An optimal geometric model for clavels delta robot," in *2015 IEEE European Modelling Symposium (EMS)*, pp. 232–237, IEEE, 2015.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Spinger, 2010.

[15] K. Oiwa, S. Maeda, and C. Ishii, "Maneuverability evaluation of a surgical robot for single-port surgery," *International Journal of Pharma Medicine and Biological Sciences*, vol. 5, no. 1, p. 38, 2016.

[16] M. Vulliez, S. Zeghloul, and O. Khatib, "Design strategy and issues of the delthaptic, a new 6-dof parallel haptic device," *Mechanism and Machine Theory*, vol. 128, pp. 395–411, 2018.

[17] J. Magill, B. Anderson, G. Anderson, P. Hess, and S. Pratt, "Multi-axis mechanical simulator for epidural needle insertion," in *International Symposium on Medical Simulation*, pp. 267–276, Springer, 2004.

[18] G. Baud-Bovy, F. Tatti, and N. A. Borghese, "Ability of low-cost force-feedback device to influence postural stability," *IEEE transactions on haptics*, vol. 8, no. 2, pp. 130–139, 2014.

[19] S. Chehabeddine, M. H. Jamil, W. Park, D. L. Sefo, P. M. Loomer, and M. Eid, "Bi-manual haptic-based periodontal simulation with finger support and vibrotactile feedback," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–17, 2021.

[20] Á. Gutiérrez, D. Sepúlveda-Muñoz, Á. Gil-Agudo, and A. de los Reyes Guzman, "Serious game platform with haptic feedback and emg monitoring for upper limb rehabilitation and smoothness quantification on spinal cord injury patients," *Applied Sciences*, vol. 10, no. 3, p. 963, 2020.

[21] H. Shen, C. Wang, L. Xie, S. Zhou, and X. Zhou, "Research for a master-slave cerebrovascular interventional robotic system," in *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 395–398, IEEE, 2018.

[22] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999.

[23] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control*, vol. 3. wiley New York, 2006.

[24] I. Torres-Rodrıguez, J. D. Castro-Dıaz, and J. Pliego-Jiménez, "El robot paralelo novint falcon como plataforma experimental háptica," *Congreso Nacional de Control Automático, Monterrey, Nuevo León*, pp. 587–593, 2017.

[25] F. C. Park, *Encyclopedia of Systems and Control*, ch. Robot Learning, pp. 1661–1666. Springer International Publishing, 2021.

[26] J. Serracín, I. Moreno, T. Vásquez, and I. Bonilla, "Prototipo de robot paralelo delta para fortalecer el proceso educativo a nivel superior," in *Memorias de Congresos UTP*, pp. 155–160, 2017.

[27] S. Stanković, "Virtual reality and virtual environments in 10 lectures," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 18, no. 3, pp. 1–197, 2015.

[28] C. Melchiorri, *Encyclopedia of Systems and Control*, ch. Robot Learning, pp. 1911–1922. Springer International Publishing, 2021.

[29] A. B. Lasheras, "Impacto de los errores de calibración en la precisión de un robot," *Proyecto de Fin de Carrera, Universidad de Zaragoza*, 2013.

[30] SIMSCALE, "Fea." `https://www.simscale.com/docs/simwiki/fea-finite-element-analysis/what-is-fea-finite-element-analysis/`, 2011.

[31] SIMSCALE, "Vonmises-stress." `https://www.simscale.com/docs/simwiki/fea-finite-element-analysis/what-is-von-mises-stress/`, 2011.

[32] R. L. Norton, *Diseño de máquinas.* sd: Pearson-Prentice Hall,, 1999.

[33] D. L. Goetsch and S. Davis, "Despliegue de la función de calidad.(quality function deployment)," *D. Goestsch, D. Despliegue de la Función de Calidad.(Quality Function Deployment)*, 1994.

[34] haptics house, "Novint Falcon - haptic device." `https://https://hapticshouse.com/`, 2021.

[35] S. Martin and N. Hillier, "Characterisation of the novint falcon haptic device for application as a robot manipulator," in *Australasian Conference on Robotics and Automation (ACRA)*, pp. 291–292, Citeseer, 2009.

[36] E. Scalona, D. Hayes, Z. Del Prete, E. Palermo, and S. Rossi, "Perturbed point-to-point reaching tasks in a 3d environment using a portable haptic device," *Electronics*, vol. 8, no. 1, p. 32, 2019.

[37] B. Menges, A. Guenard, and P. Henaff, "Improving control of a teleoperated robot using an adaptive oscillator to perform a rhythmic task for foundry applications," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pp. 346–350, IEEE, 2021.

[38] W. Park, V. Babushkin, S. Tahir, and M. Eid, "Haptic guidance to support handwriting for children with learning difficulties," *IEEE Transactions on Haptics*, 2021.

[39] M. Mohammadi, T. L. Baldi, S. Scheggi, and D. Prattichizzo, "Fingertip force estimation via inertial and magnetic sensors in deformable object manipulation," in *2016 IEEE Haptics Symposium (HAPTICS)*, pp. 284–289, IEEE, 2016.

[40] F. Chen, H. Shen, C. Wang, L. Xie, S. Zhou, and H. Wang, "The model experimental evaluation of the cardiovascular interventional surgery robot system," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp. 114–120, IEEE, 2018.

[41] G. Salvietti, L. Meli, G. Gioioso, M. Malvezzi, and D. Prattichizzo, "Multicontact bilateral telemanipulation with kinematic asymmetries," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 445–456, 2016.

[42] X. Zhang, R. Zhang, L. Chen, and X. Zhang, "Natural gesture control of a delta robot using leap motion," in *Journal of Physics: Conference Series*, p. 032042, IOP Publishing, 2019.

[43] H. Saafi, C. Preault, M. A. Laribi, and S. Zeghloul, "A novel kinematic of a 4 do fs haptic device based on the delta robot architecture," in *International Conference on Robotics in Alpe-Adria Danube Region*, pp. 699–706, Springer, 2017.

[44] L. Guanyang, G. Xuda, L. Lingzhi, and W. Yan, "Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration," *Chinese Journal of Aeronautics*, vol. 32, no. 3, pp. 723–736, 2019.

[45] J. Somló, G. D. Varga, M. Zenkl, and B. Mikó, "The "phantom" delta robot a new device for parallel robot investigations," *Acta Polytechnica Hungarica*, vol. 15, no. 4, pp. 143–160, 2018.

[46] J. M. Jacinto-Villegas, O. Portillo-Rodríguez, R. Martinez-Mendez, C. Daza-Merino, A. H. Vilchis-González, and J. C. Avila-Vilchis, "Sistema para control de posición basado en rapid control prototyping (rcp) usando simulink y swb32," *Komputer Sapiens*, vol. 3, p. 11–15, Sep-Dec 2019.

[47] M. A. Arana-González, J. M. Jacinto-Villegas, B. Saldivar, and O. Portillo-Rodríguez, "Estimación y validación experimental de un modelo de primer orden para el control de velocidad de un motor de cd," *EURE-UAQ*, vol. 1, p. 22–34, Aug 2022.

[48] C. C. Martínez, J. C. Ávila-Vilchis, J. M. Jacinto-Villegas, B. Saldivar, and A. H. Vilchis-González, "Sliding mode control for the regulation problem of an aerodynamic angular system: Experimental platform and validation," *International Journal of Control, Automation and Systems*, vol. 19, pp. 2395–2405, 2021.

[49] D. E. Oku, E. P. Obot, and C. Author, "Comparative study of pd, pi and pid controllers for control of a single joint system in robots," *Int. J. Eng. Sci*, vol. 7, no. 9, p. V2, 2018.

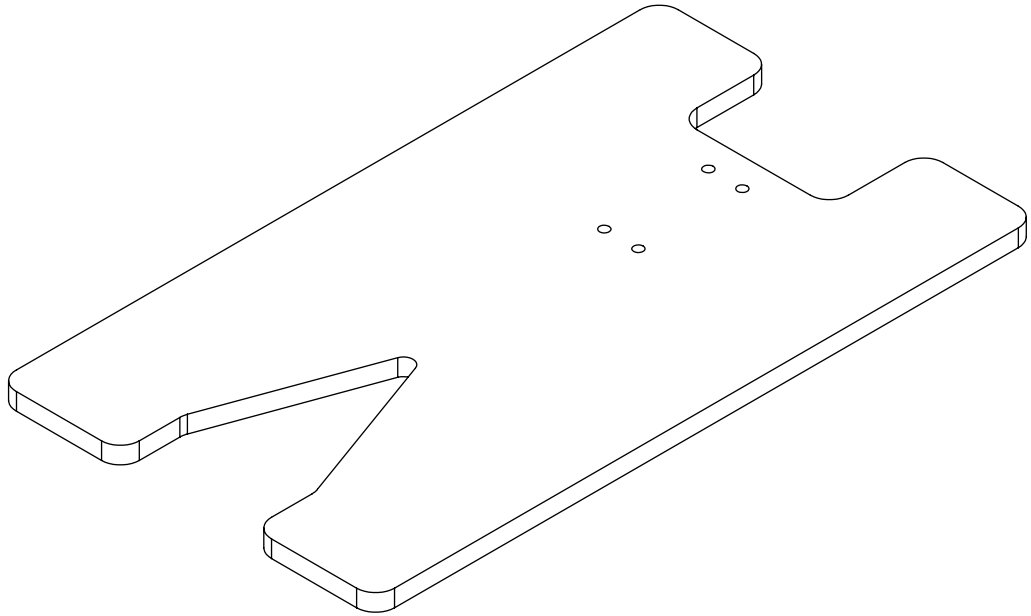
[50] A. Olsson, "Modeling and control of a delta-3 robot," *MSc Theses*, 2009.

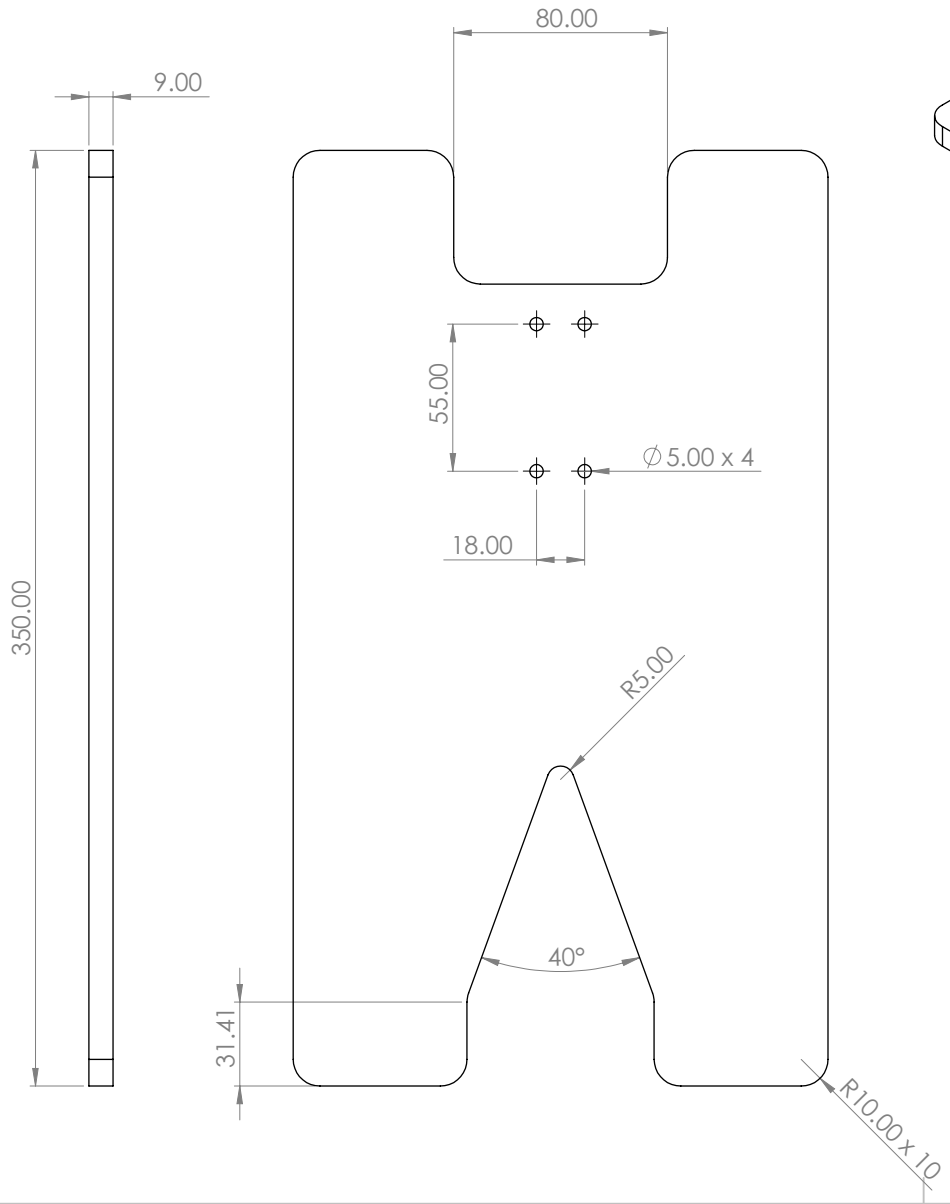
[51] R. L. Williams, "The delta parallel robot: Kinematics solutions robert l. williams ii, ph. d., williar4@ ohio. edu mechanical engineering," *Ohio University*, October 2016.

Apéndices

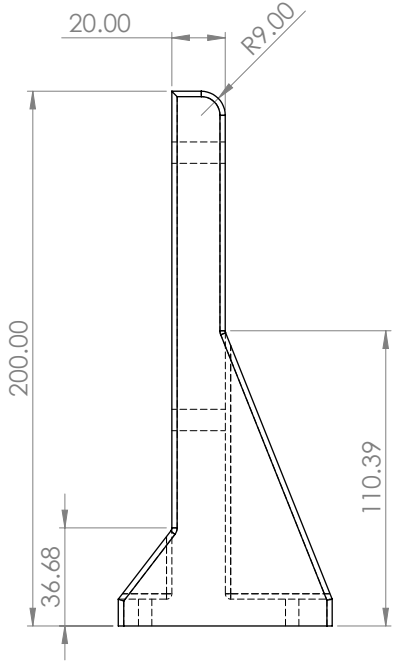
Apéndice A

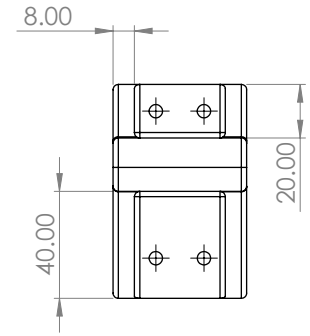
Diagramas mecánicos 2D

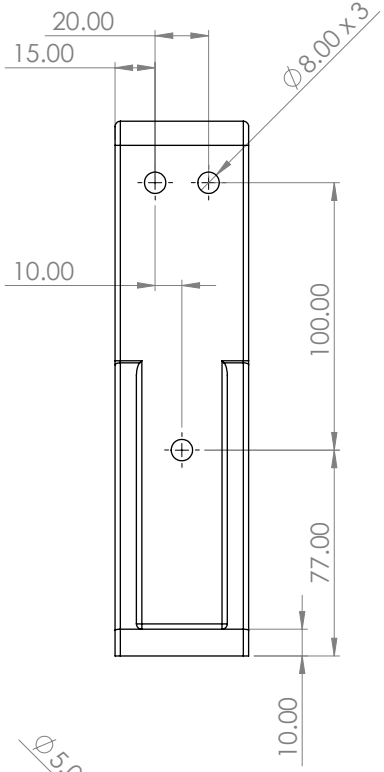


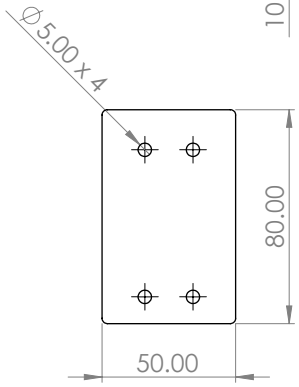


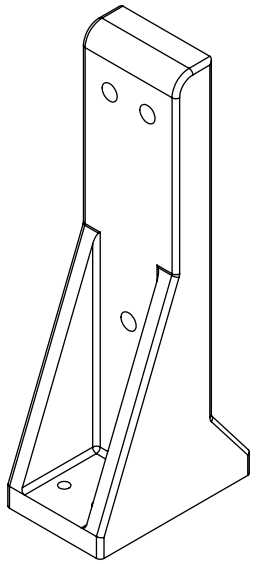
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
							UAEMex	
	NAME	SIGNATURE	DATE				TITLE: Base General para trabajos horizontales	
DRAWN	Ing. Raúl Serna R.		30/08/23					
CHK'D								
APPV'D								
MFG								
Q.A				MATERIAL: Triplay			DWG NO. RD.01-110	A3
				WEIGHT:			SCALE:1:2	SHEET 1 OF 1



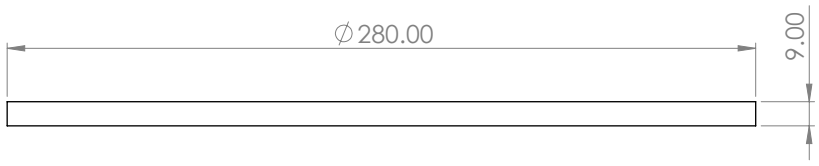


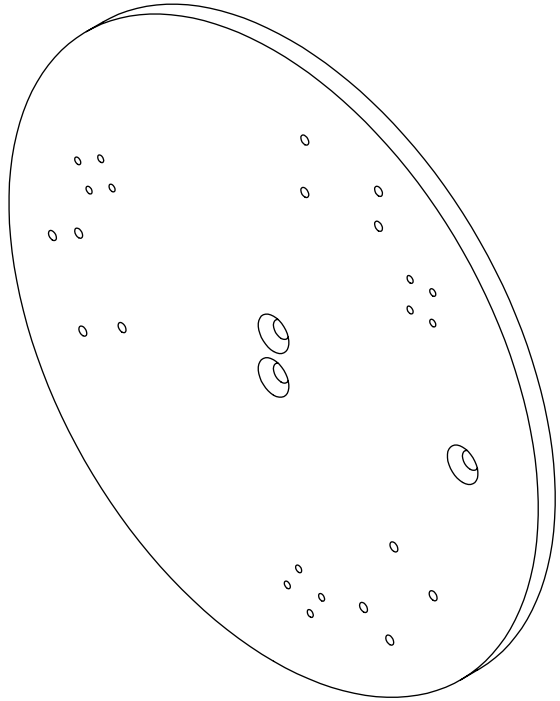


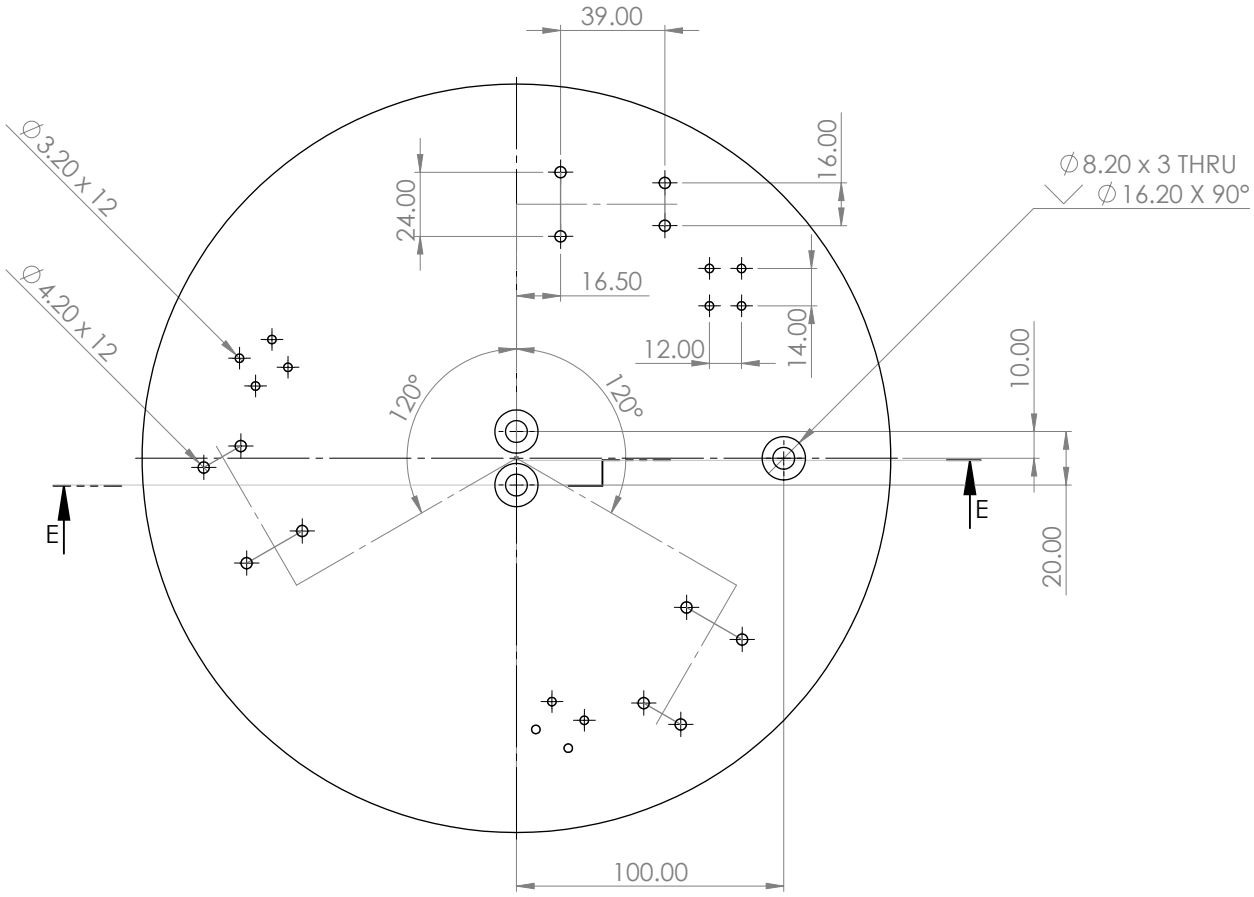




UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Soporte para ensamble de base de trabajos horizontales y verticales.	
DRAWN	Ing. Raúl Serna R.		30/08/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: PLA		DWG NO. RD.01-115	A3
				WEIGHT:		SCALE:1:2	SHEET 1 OF 1

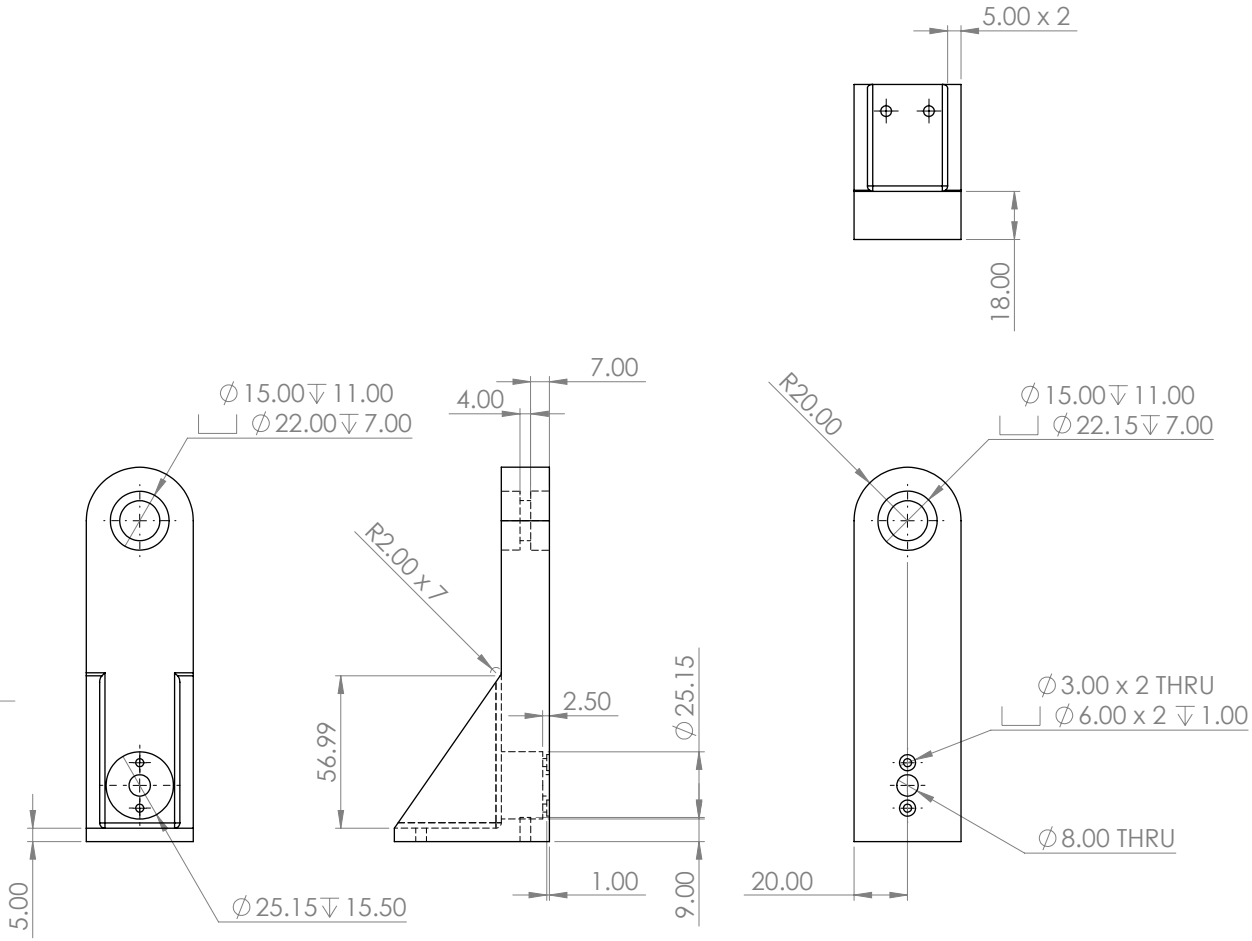


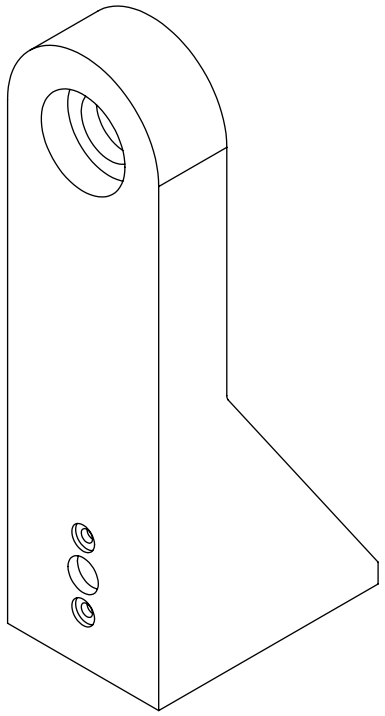


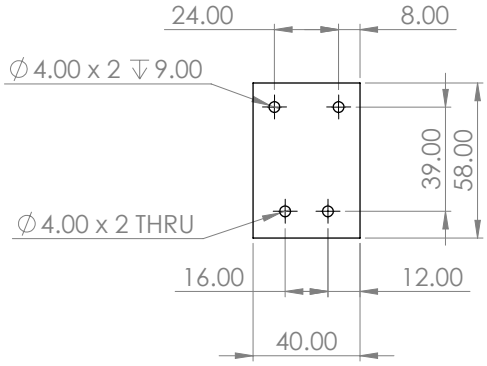


SECTION E-E

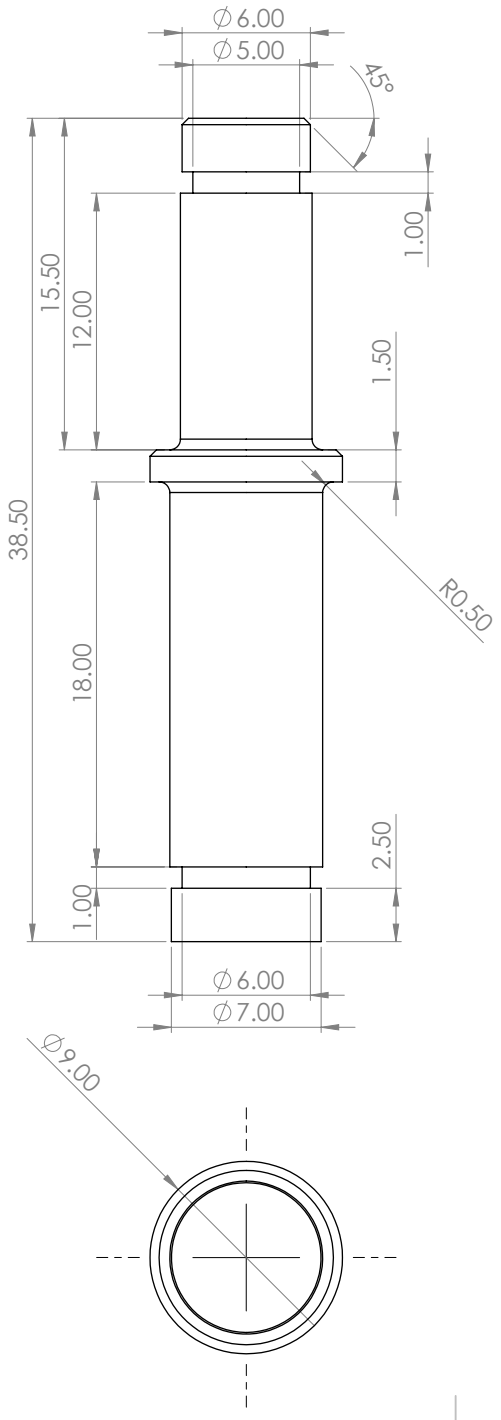
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE				UAEMex
DRAWN	Ing. Raúl Serna R.		01/09/23				TITLE: Base circular para trabajos verticales
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: MDF			DWG NO. RD.01-120 / A3
				WEIGHT:			SCALE:1:2 / SHEET 1 OF 1

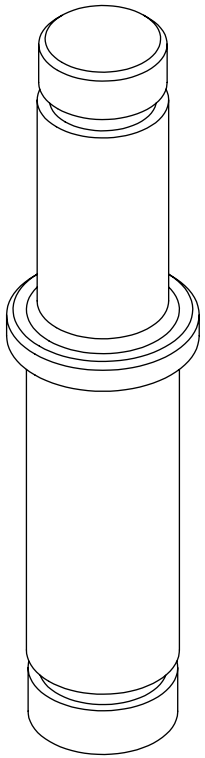




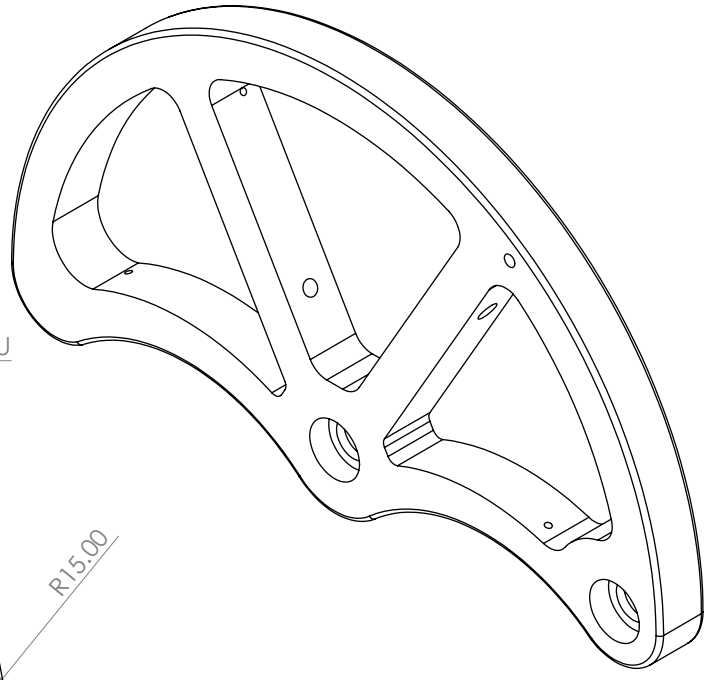


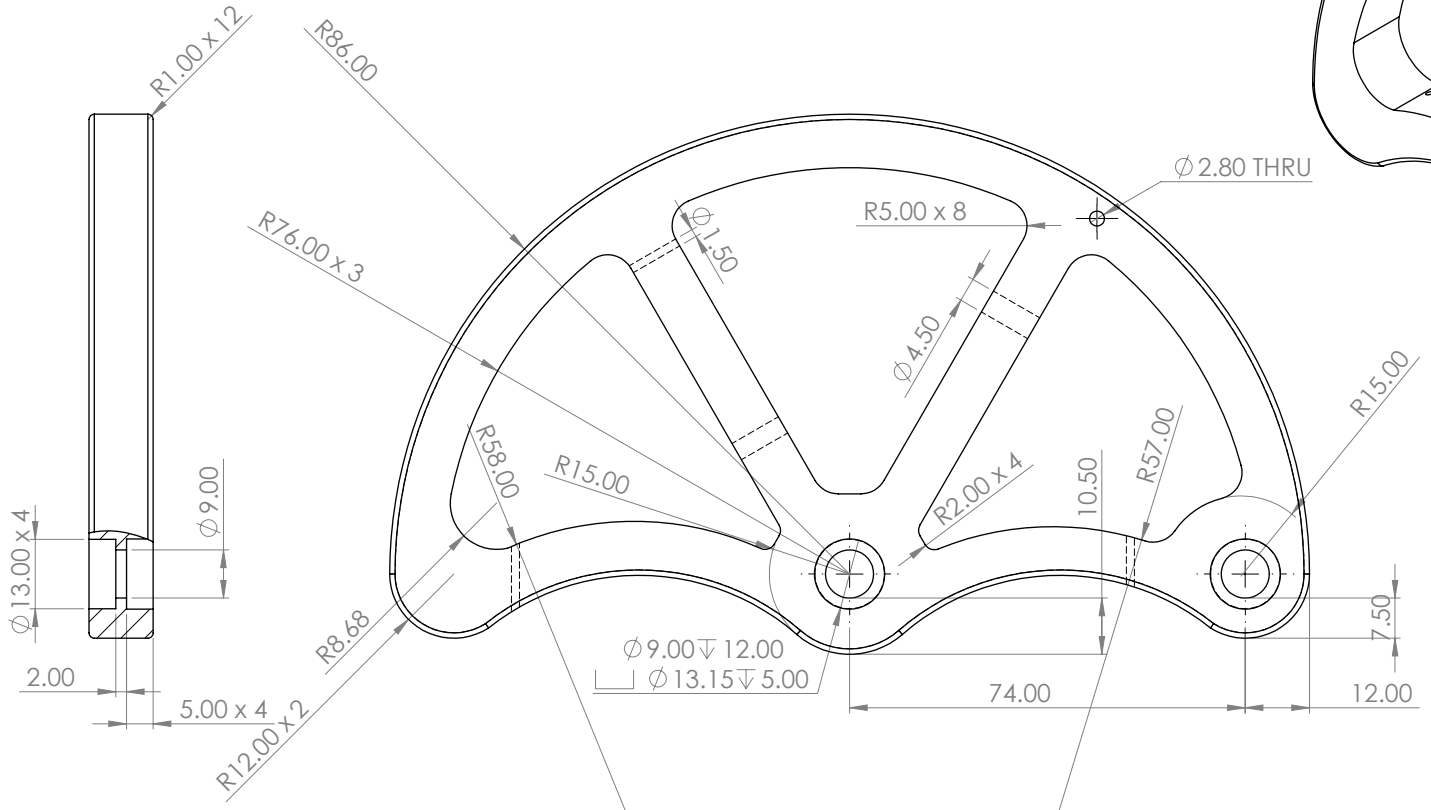
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
					UAEMex	
	NAME	SIGNATURE	DATE		TITLE: Base Soporte para capstan y motor	
DRAWN	Ing. Raúl Serna R.		01/09/23			
CHK'D						
APP'V'D						
MFG						
Q.A				MATERIAL: PLA	DWG NO. RD.01-125	A3
				WEIGHT:	SCALE:1:2	SHEET 1 OF 1

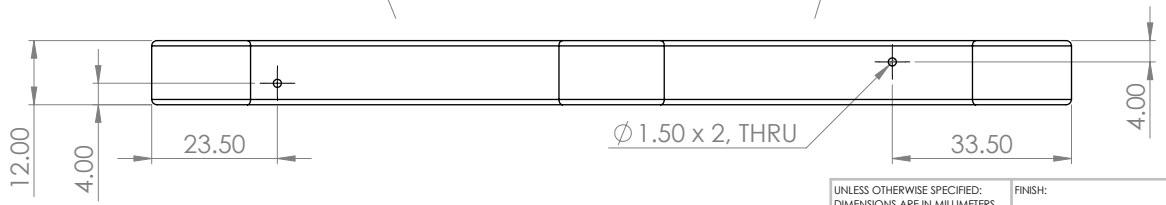




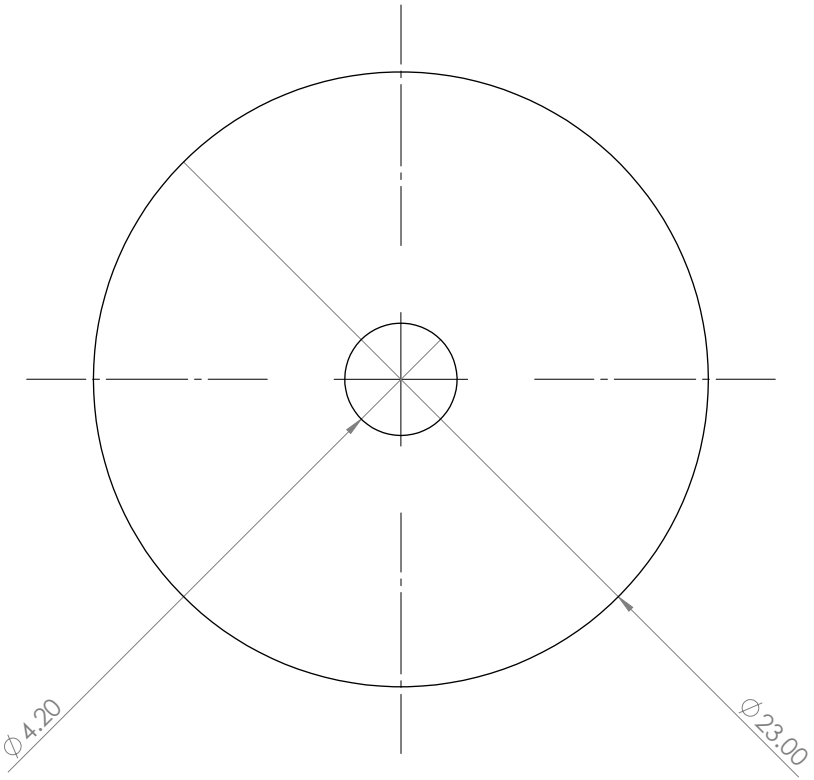
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Perno para ensamble de capstan y base de motor	
DRAWN	Ing. Raúl Serna R.		05/09/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: Aluminio		DWG NO. RD.01-130	A3
				WEIGHT:		SCALE:4:1	SHEET 1 OF 1

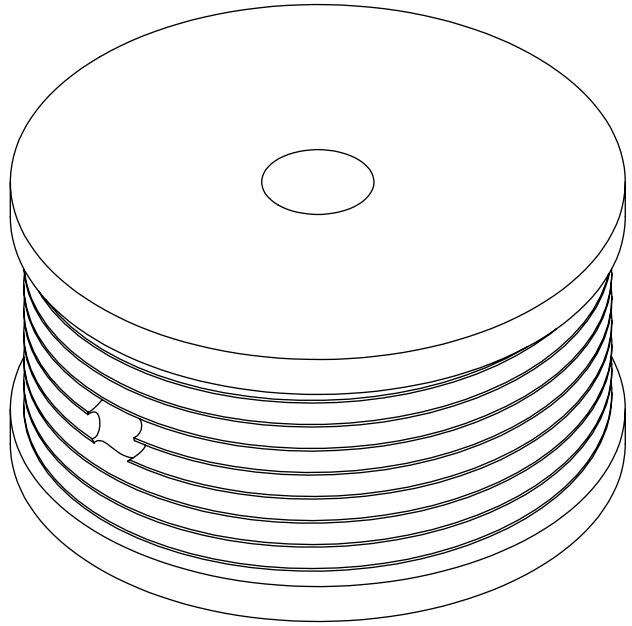


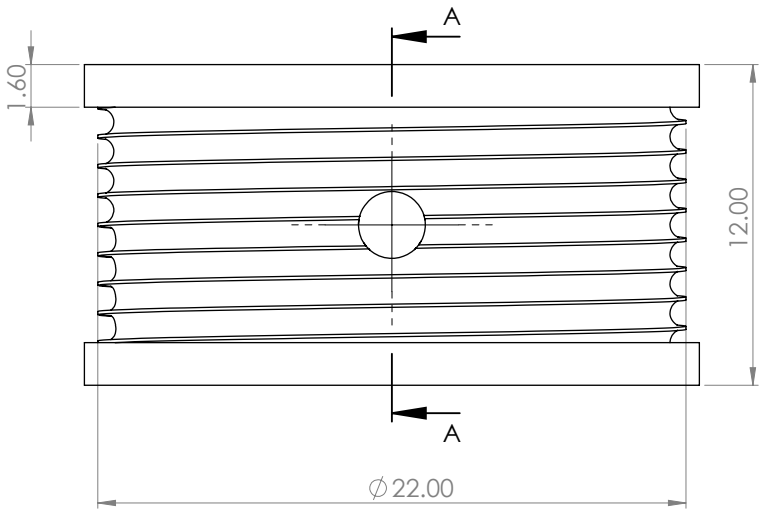


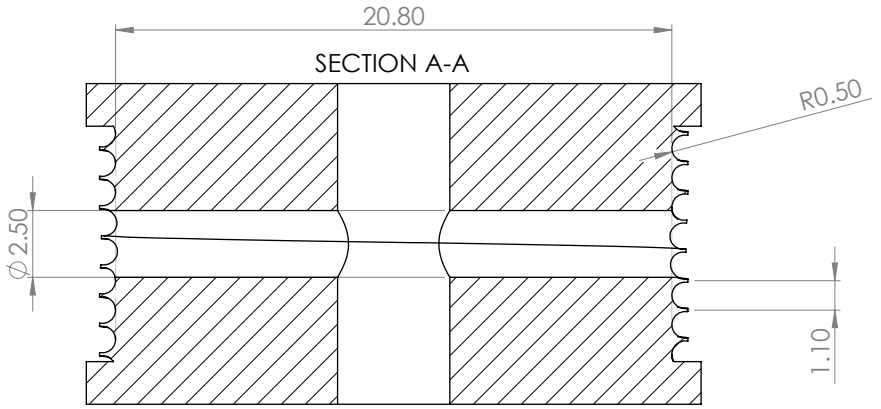


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN	Ing. Raúl Serna R.		05/09/23			Capstan	
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
				PLA		RD.01-135	
				WEIGHT:		SCALE:1:1	SHEET 1 OF 1

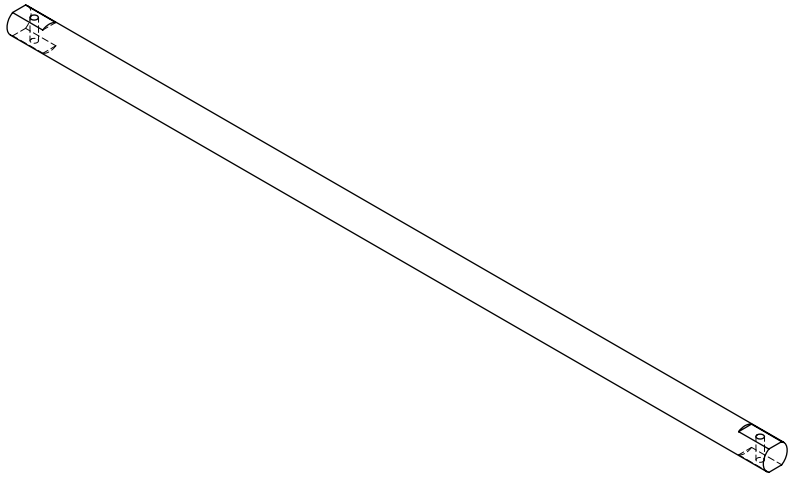


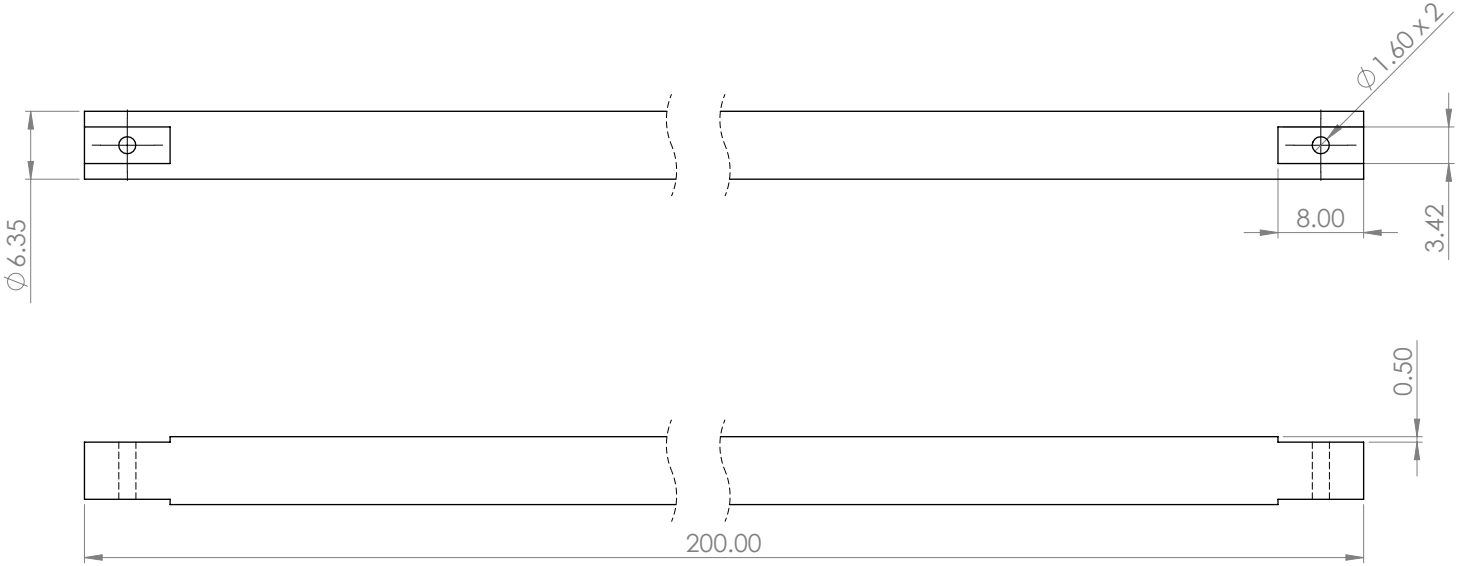




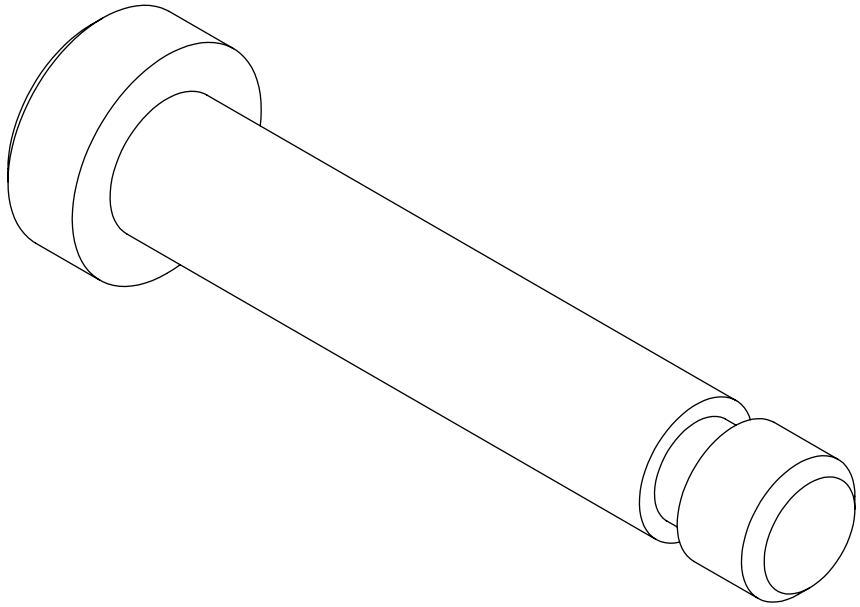


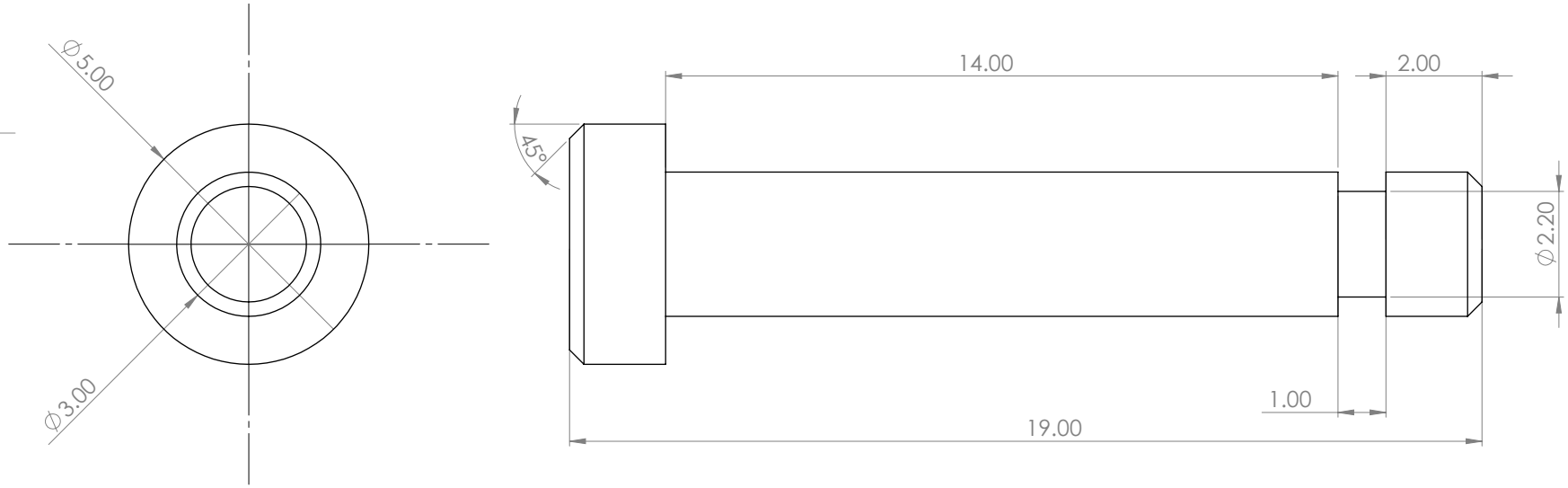
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN	Ing. Raúl Serna R.		05/09/23			Polea	
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
				PLA		RD.01-140	
				WEIGHT:		SCALE:5:1	SHEET 1 OF 1



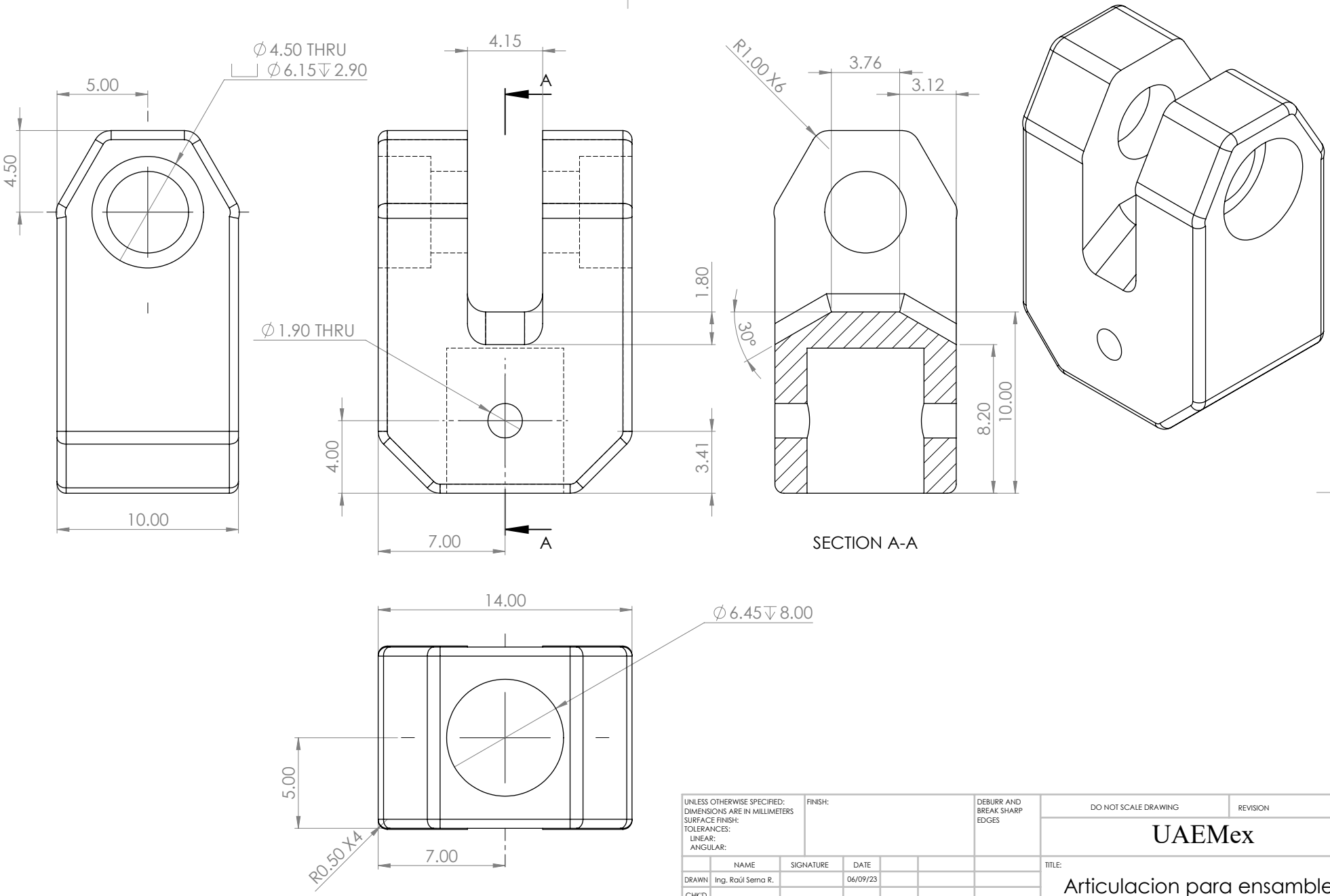


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Barras paralelas circulares	
DRAWN	Ing. Raúl Serna R.		05/09/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: Aluminio		DWG NO. RD.01-145	A3
				WEIGHT:		SCALE:2:1	SHEET 1 OF 1

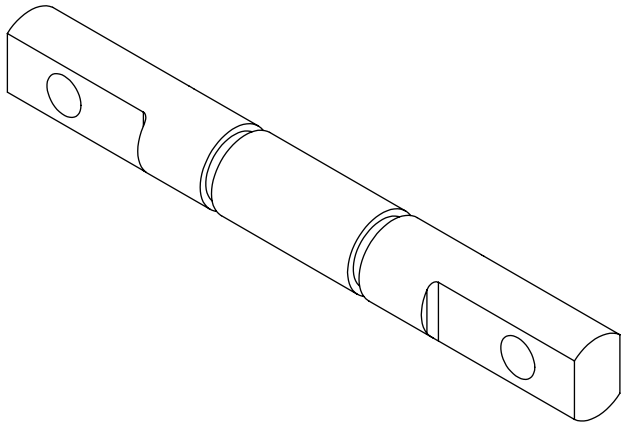


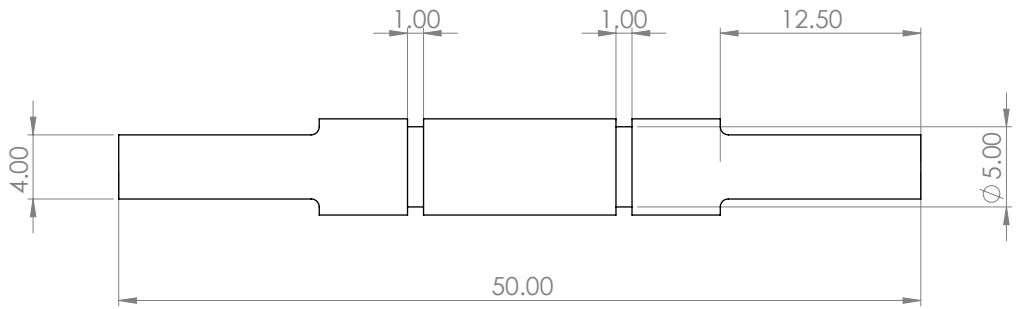


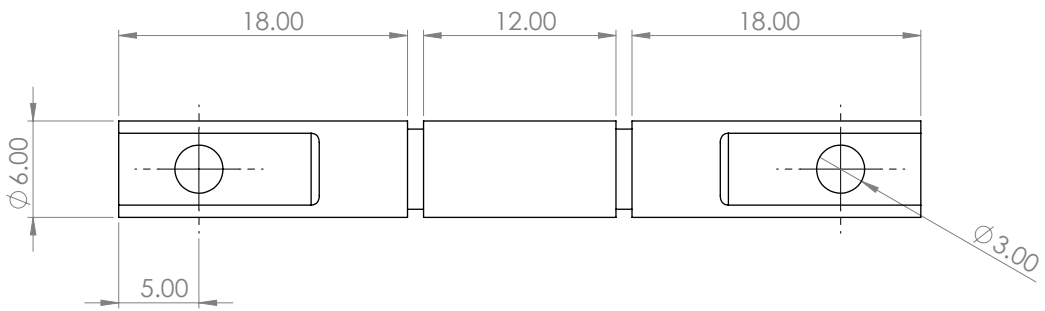
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
							UAEMex	
	NAME	SIGNATURE	DATE				TITLE: Perno para ensamble de barras paralelas con articulación	
DRAWN	Ing. Raúl Serna R.		05/09/23					
CHK'D								
APPV'D								
MFG								
Q.A				MATERIAL: Aluminio			DWG NO. RD.01-150	A3
				WEIGHT:			SCALE:10:1	SHEET 1 OF 1

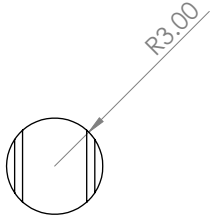


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Articulacion para ensamble con barras paralelas	
DRAWN	Ing. Raúl Serna R.		06/09/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: PLA		DWG NO. RD.01-155	A3
				WEIGHT:		SCALE:5:1	SHEET 1 OF 1

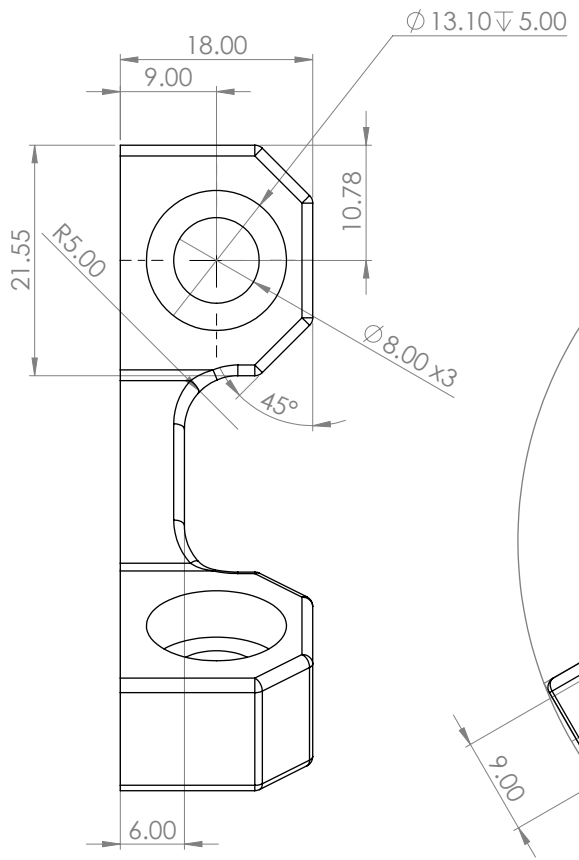


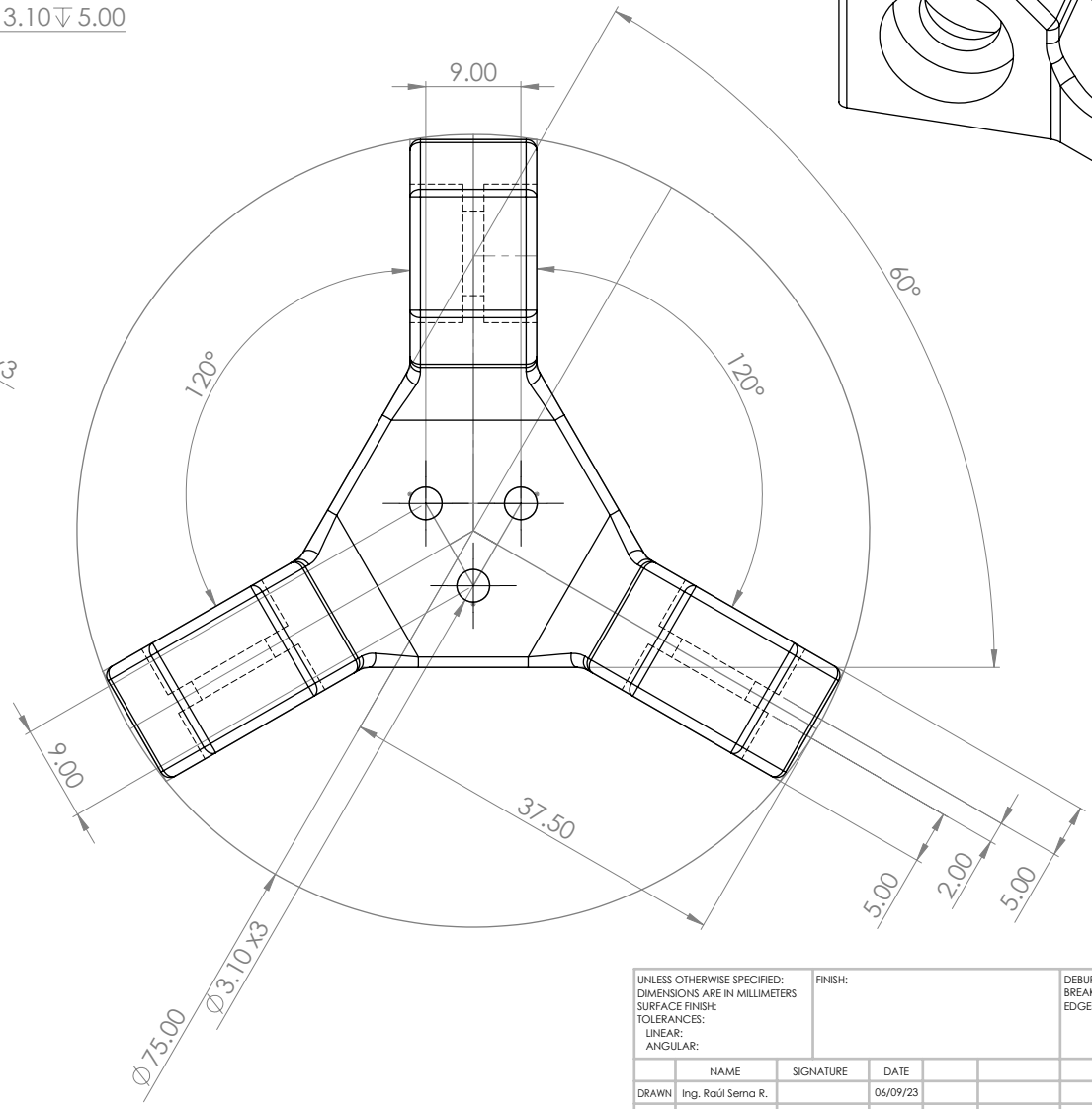


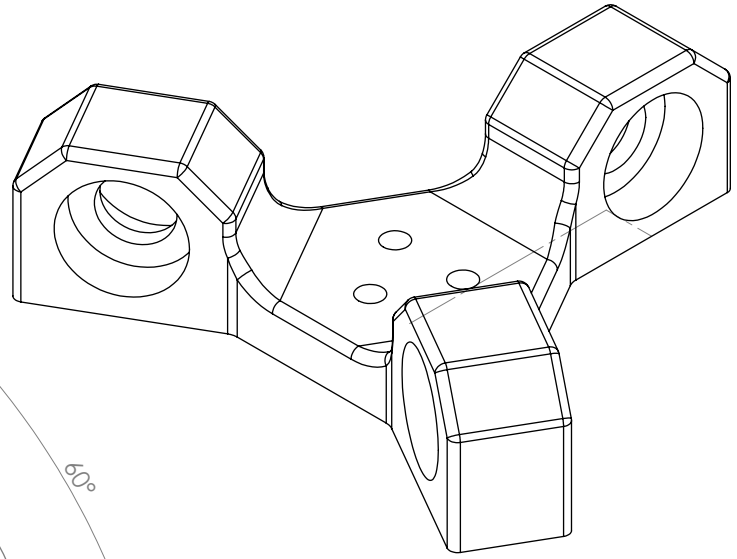




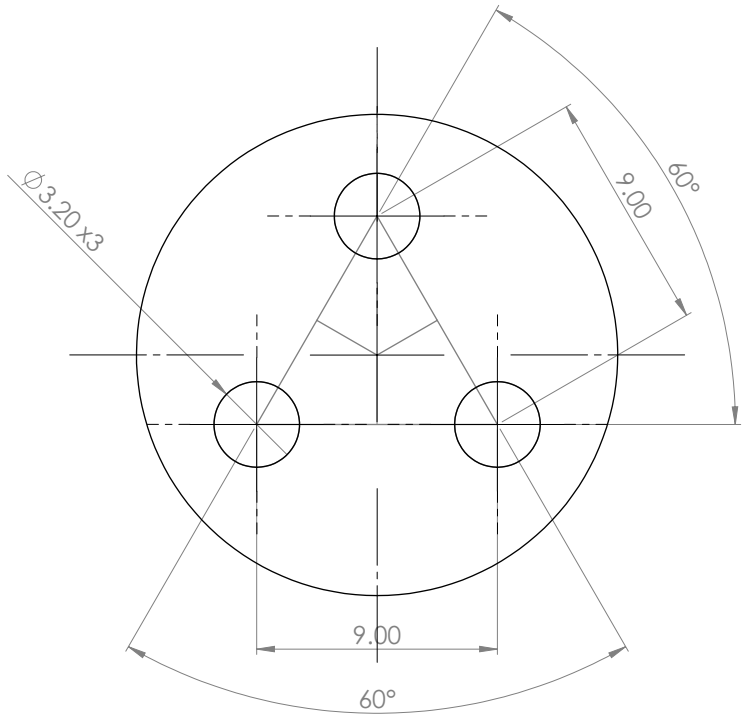
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
							UAEMex	
	NAME	SIGNATURE	DATE				TITLE: Eje para ensamble de paralelas con capstan y base movil	
DRAWN	Ing- Raúl Serna R.		05/08/23					
CHK'D								
APPV'D								
MFG								
Q.A				MATERIAL: Aluminio			DWG NO. RD.01-160	A3
				WEIGHT:			SCALE:2:1	SHEET 1 OF 1

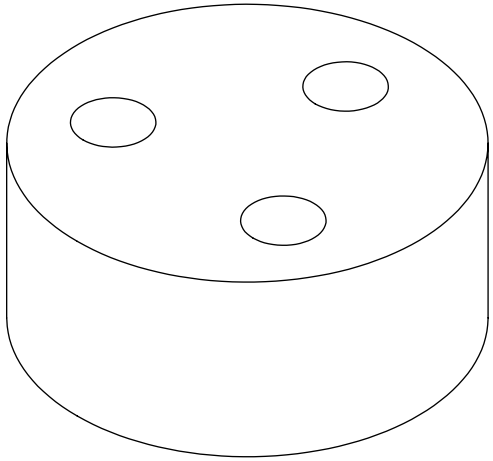




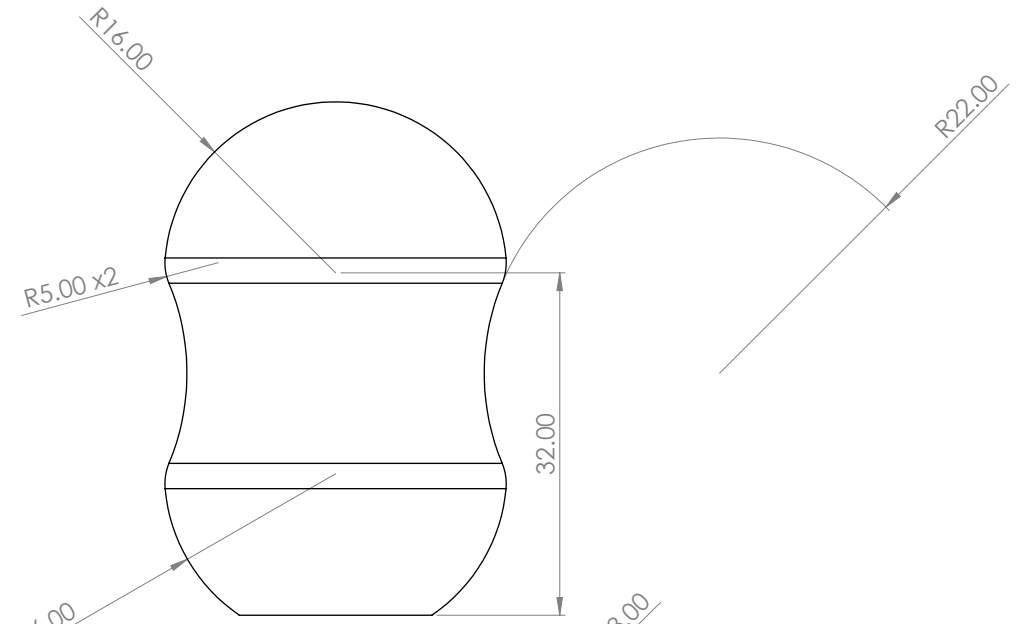


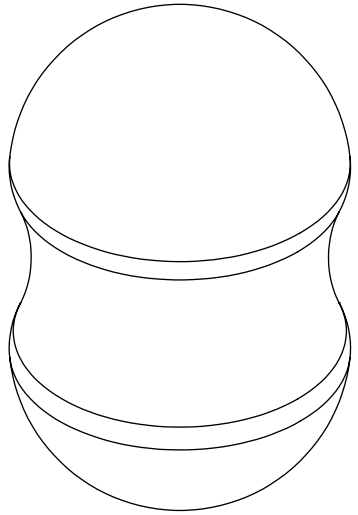
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN	Ing. Raúl Serna R.		06/09/23			Base movil	
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
						RD.01-165	
				WEIGHT:		SCALE:2:1	SHEET 1 OF 1

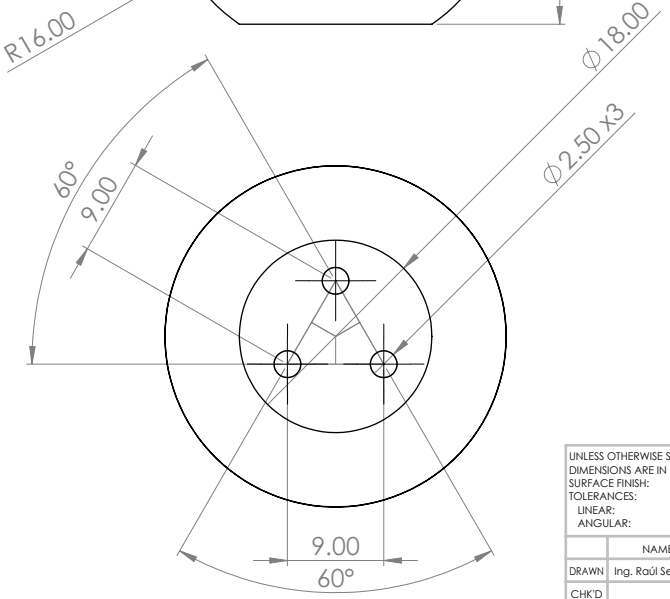




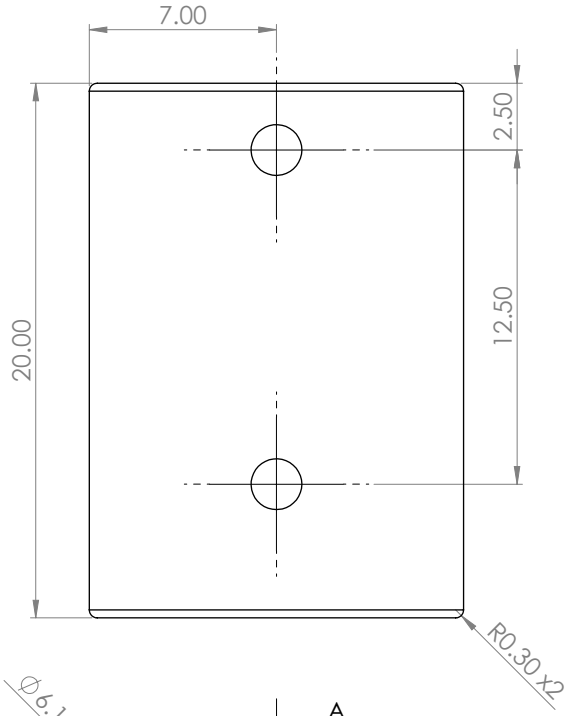
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Soporte para efector final	
DRAWN	Ing. Raúl Serna R.		11/09/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: PLA		DWG NO. RD.01-170	A3
				WEIGHT:		SCALE:5:1	SHEET 1 OF 1

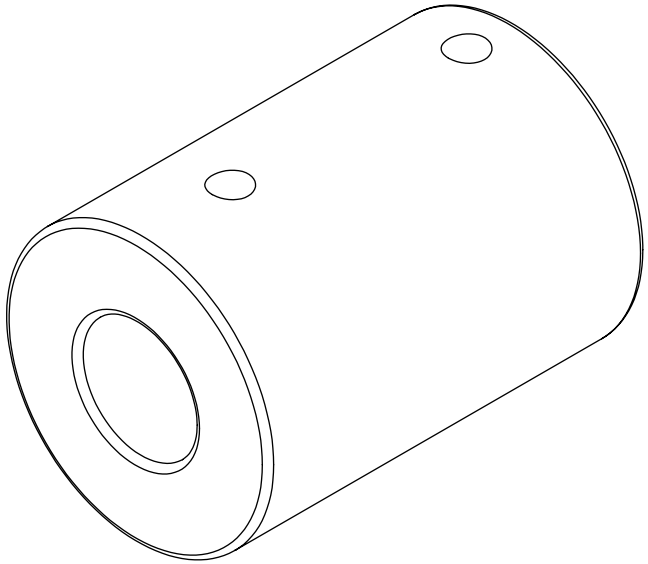


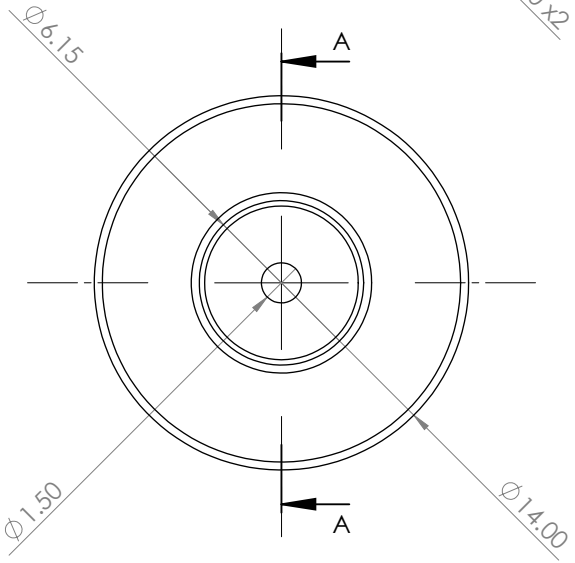


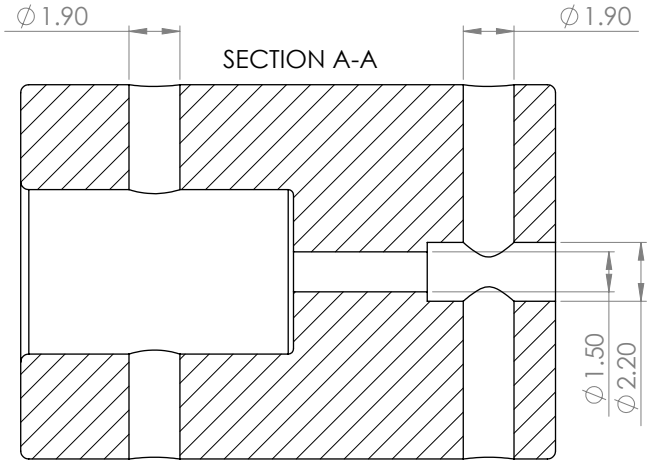


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
							UAEMex	
	NAME	SIGNATURE	DATE				TITLE:	
DRAWN	Ing. Raúl Serna R.		11/09/23				Efector final	
CHK'D								
APPV'D								
MFG								
Q.A				MATERIAL:			DWG NO.	A3
				PLA			RD.01-175	
				WEIGHT:			SCALE:2:1	SHEET 1 OF 1

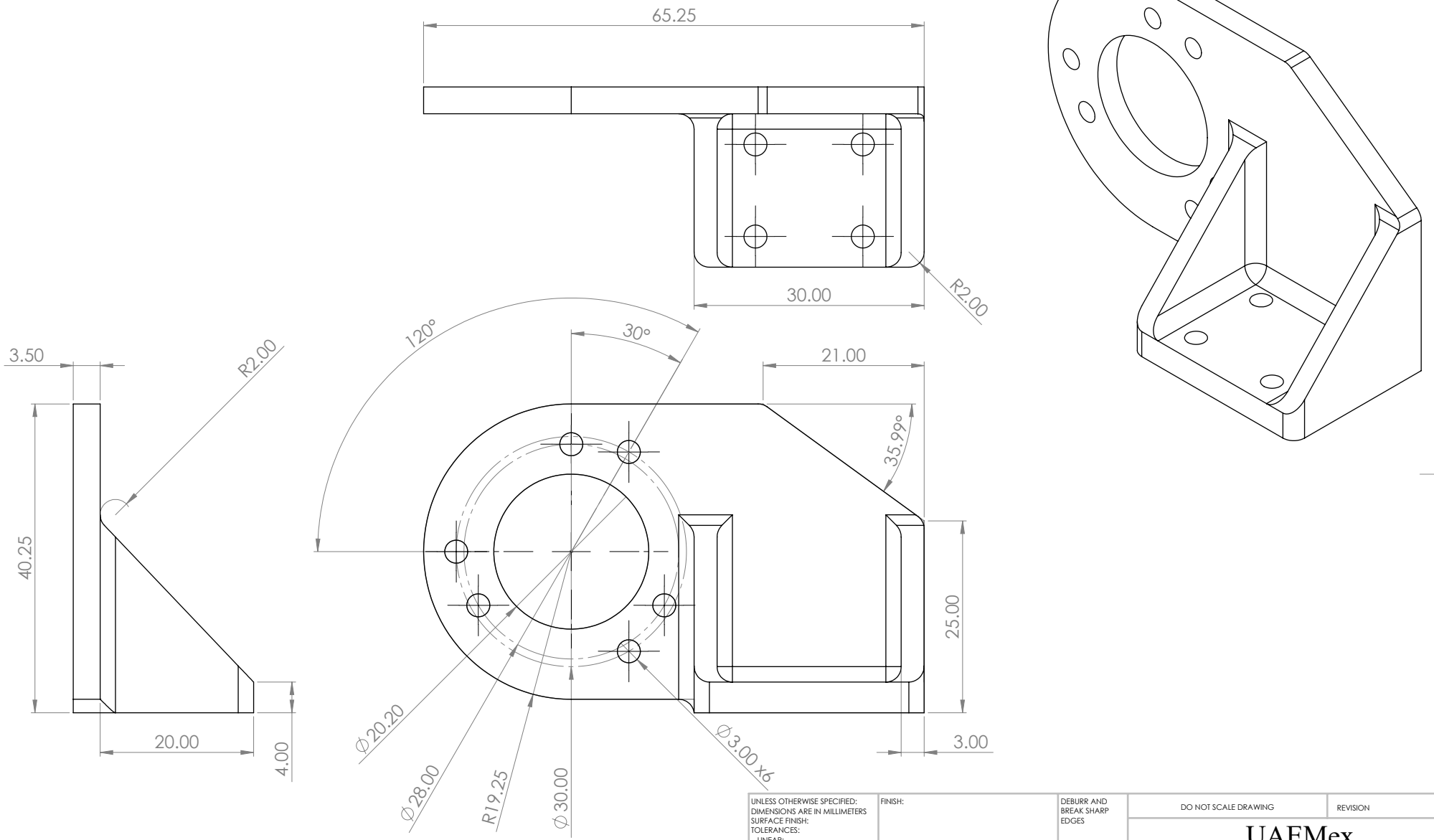








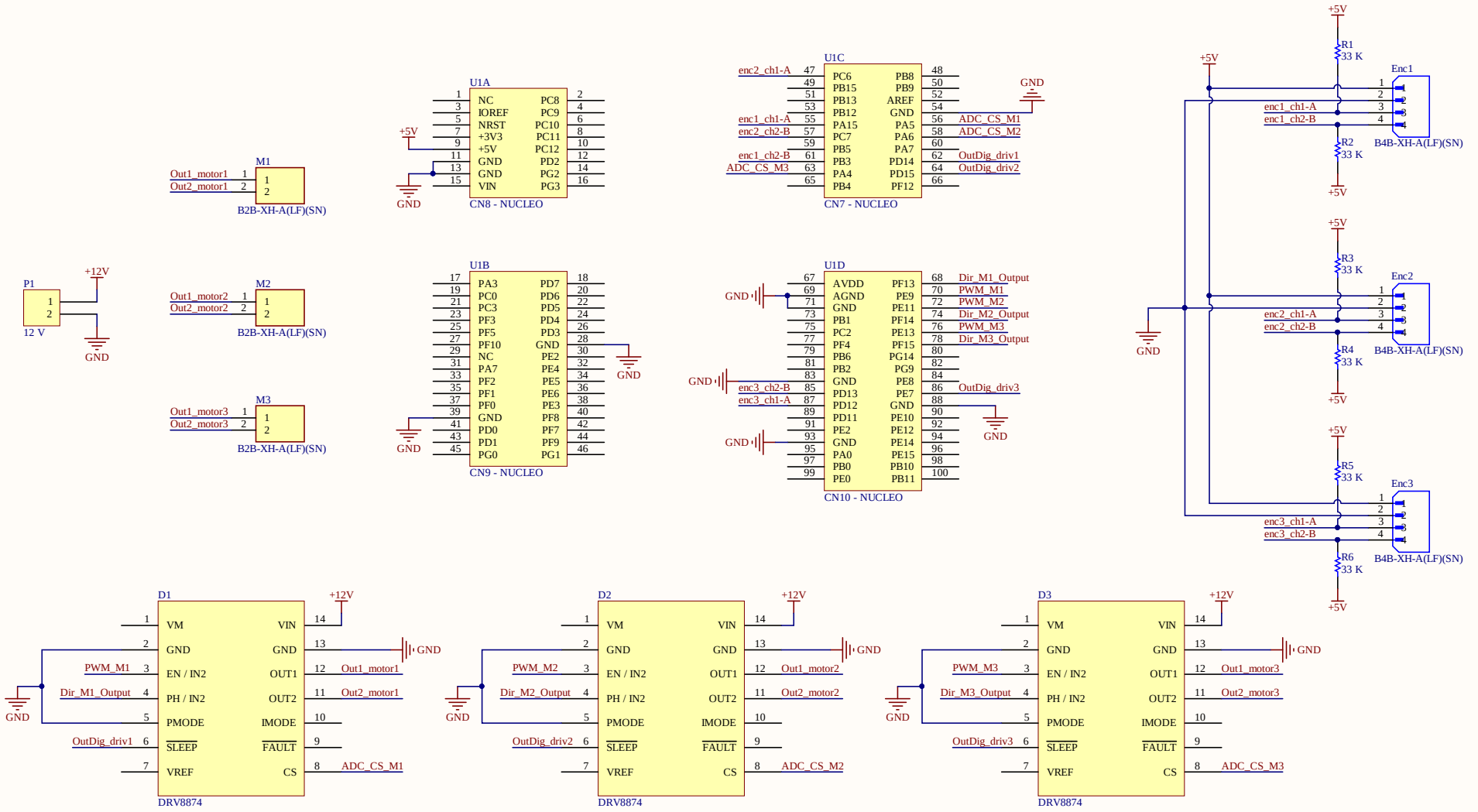
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN	Ing. Raúl Serna R.		11/09/23			Cople para encoder	
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
				PLA		RD.01-180	
				WEIGHT:		SCALE:5:1	SHEET 1 OF 1

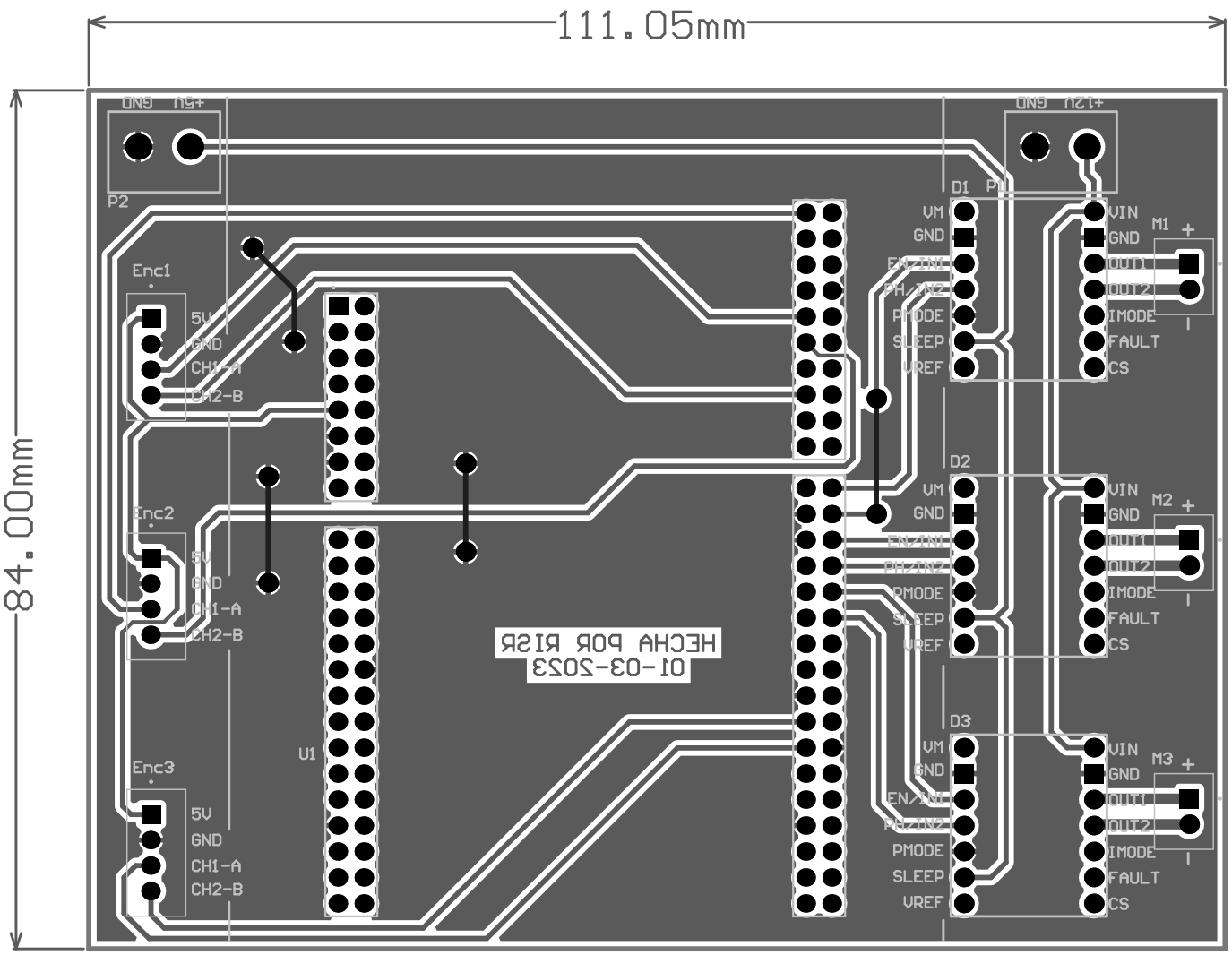


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
						UAEMex	
	NAME	SIGNATURE	DATE			TITLE: Base para encoder	
DRAWN	Ing. Raúl Serna R.		12/09/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: PLA		DWG NO. RD.01-185	A3
				WEIGHT:		SCALE:2:1	SHEET 1 OF 1

Apéndice B

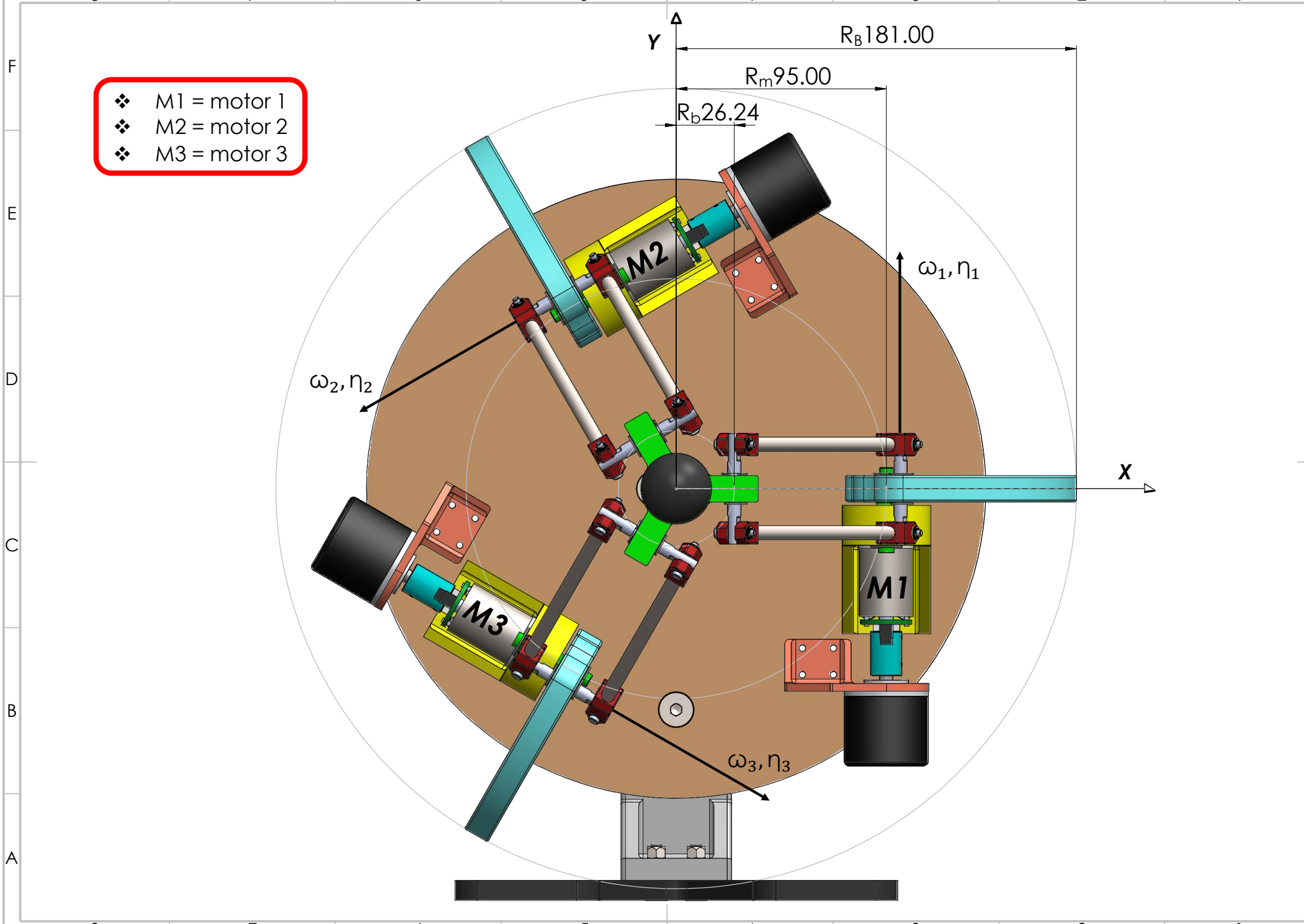
Esquemático PCB

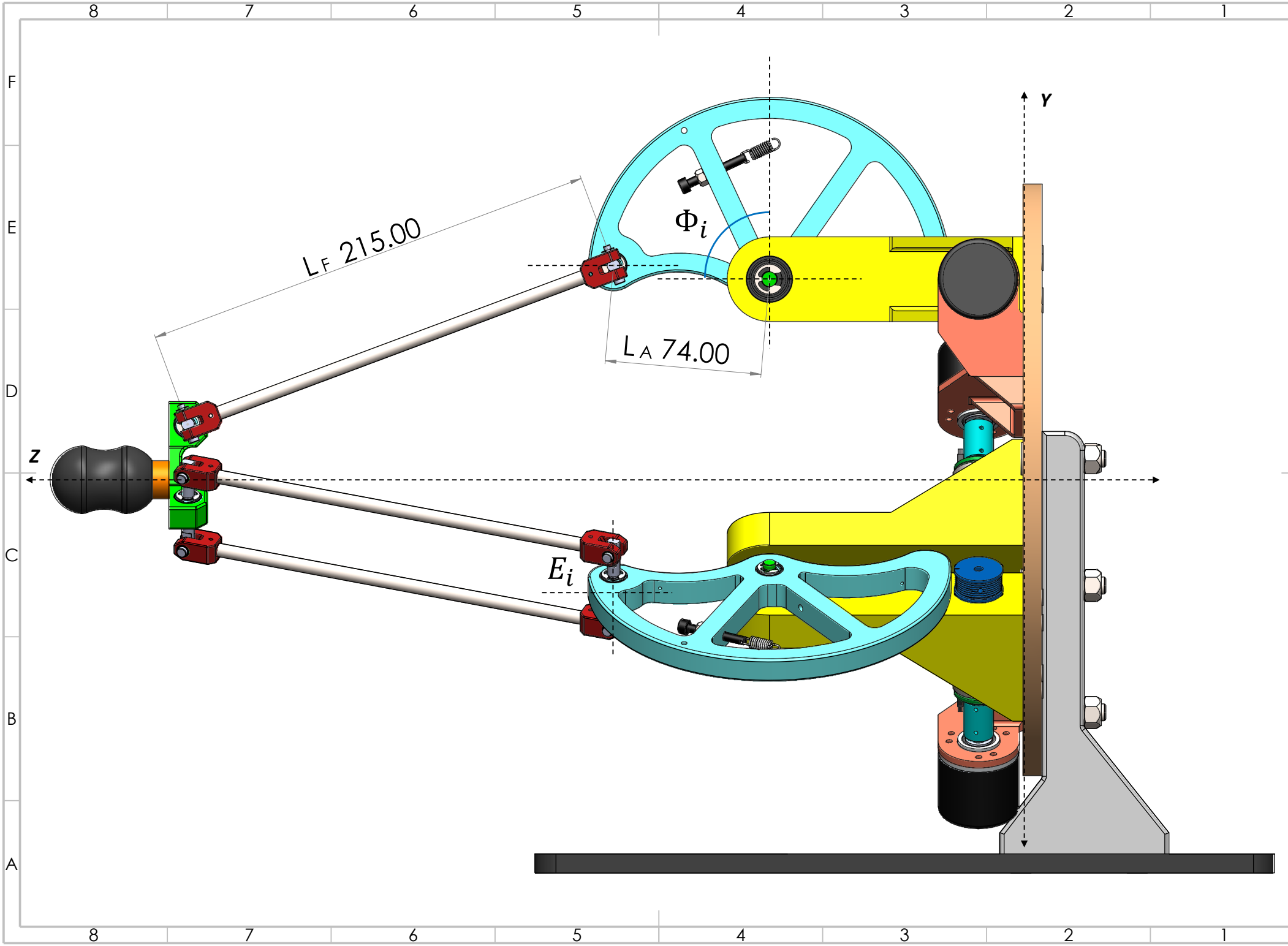

M1
Out1_motor1
Out2_motor1
B2B-XH-A(LF)(SN)
U1A
NC
IOREF
NRST
+3V3
+5V
GND
GND
VIN
PC8
PC9
PC10
PC11
PC12
PD2
PG2
PG3
CN8 - NUCLEO
+5V
GND
U1C
enc2_ch1-A
enc1_ch1-A
enc2_ch2-B
enc1_ch2-B
ADC_CS_M3
PC6
PB15
PB13
PB12
PA15
PC7
PB5
PB3
PA4
PB4
PB8
PB9
AREF
GND
PA5
PA6
PA7
PD14
PD15
PF12
ADC_CS_M1
ADC_CS_M2
OutDig_driv1
OutDig_driv2
CN7 - NUCLEO
GND
+5V
R1
33 K
Enc1
enc1_ch1-A
enc1_ch2-B
R2
33 K
B4B-XH-A(LF)(SN)
+5V
P1
12 V
+12V
GND
M2
Out1_motor2
Out2_motor2
B2B-XH-A(LF)(SN)
U1B
PA3
PC0
PC3
PF3
PF5
PF10
NC
PA7
PF2
PF1
PF0
GND
PD0
PD1
PG0
PD7
PD6
PD5
PD4
PD3
GND
PE2
PE4
PE5
PE6
PE3
PF8
PF7
PF9
PG1
CN9 - NUCLEO
GND
GND
U1D
AVDD
AGND
GND
PB1
PC2
PF4
PB6
PB2
GND
PD13
PD12
PD11
PE2
GND
PA0
PB0
PE0
PF13
PE9
PE11
PF14
PE13
PF15
PG14
PG9
PE8
PE7
GND
PE10
PE12
PE14
PE15
PB10
PB11
Dir_M1_Output
PWM_M1
Dir_M2_Output
PWM_M2
Dir_M3_Output
PWM_M3
OutDig_driv3
CN10 - NUCLEO
GND
GND
GND
GND
GND
+5V
R3
33 K
Enc2
enc2_ch1-A
enc2_ch2-B
R4
33 K
B4B-XH-A(LF)(SN)
+5V
+5V
R5
33 K
Enc3
enc3_ch1-A
enc3_ch2-B
R6
33 K
B4B-XH-A(LF)(SN)
+5V
M3
Out1_motor3
Out2_motor3
B2B-XH-A(LF)(SN)
D1
VM
GND
EN / IN2
PH / IN2
PMODE
SLEEP
VREF
VIN
GND
OUT1
OUT2
IMODE
FAULT
CS
DRV8874
+12V
GND
PWM_M1
Dir_M1_Output
OutDig_driv1
Out1_motor1
Out2_motor1
ADC_CS_M1
D2
DRV8874
+12V
GND
PWM_M2
Dir_M2_Output
OutDig_driv2
Out1_motor2
Out2_motor2
ADC_CS_M2
D3
DRV8874
+12V
GND
PWM_M3
Dir_M3_Output
OutDig_driv3
Out1_motor3
Out2_motor3
ADC_CS_M3
Title: Esquematico Robot Delta
Draw by: Ing. Raúl Ismael Serna Ruiz
Size: A4
Revision: 1
Co. UAEMex
Date: 03/08/2023
Project: PCB_RobotDelta.PrjPcb
File: RobotDelta.SchDoc
Sheet 1 of 1

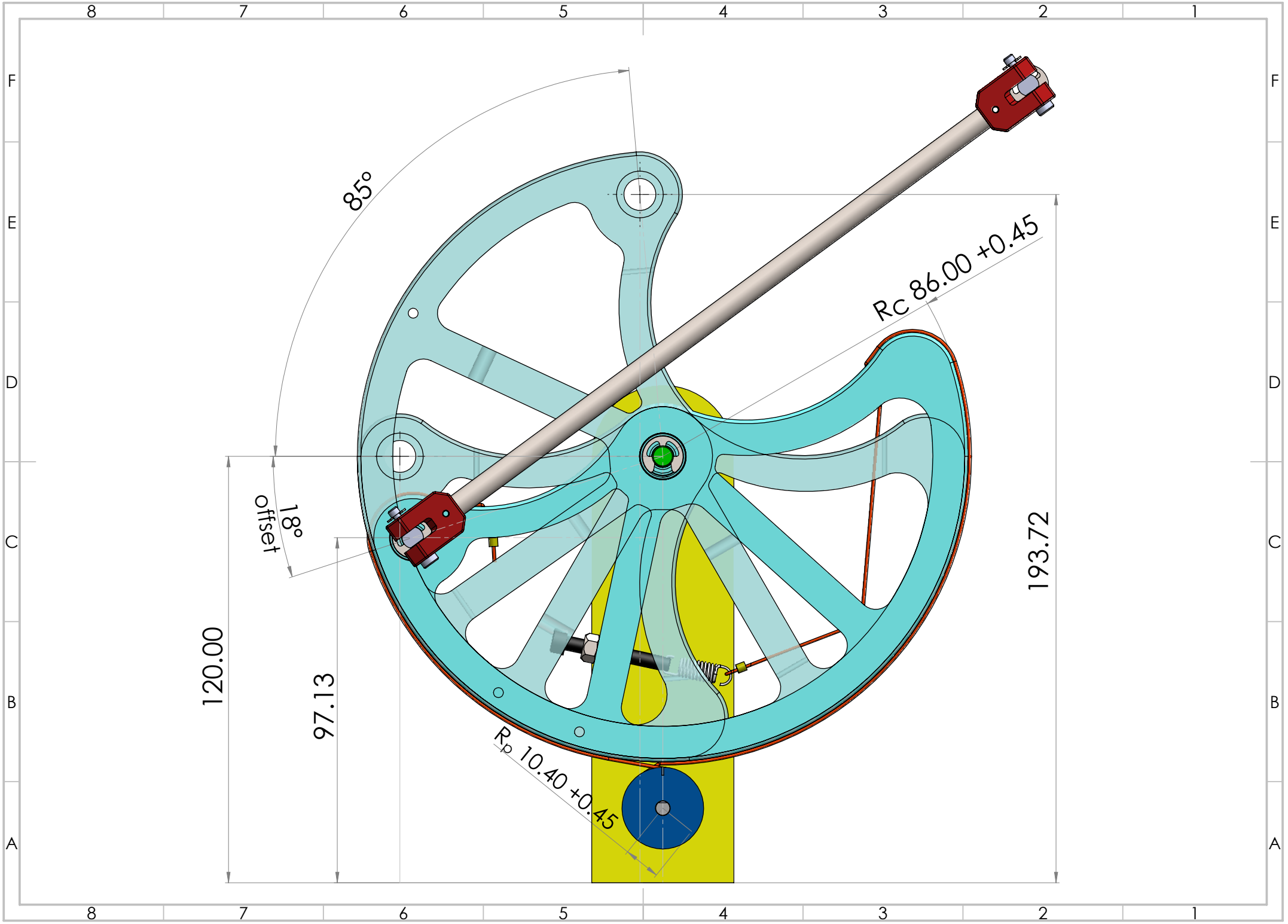

111.05mm
84.00mm
GND
+9V
P2
Enc1
5V
GND
CH1-A
CH2-B
Enc2
Enc3
U1
HECHA POR RISR
01-03-2023
GND
+12V
P1
D1
D2
D3
VM
EN/IN1
PH/IN2
PMODE
SLEEP
VREF
VIN
OUT1
OUT2
IMODE
FAULT
CS
M1
M2
M3

Apéndice C

Dimensiones del Robot tipo Delta


M1 = motor 1
M2 = motor 2
M3 = motor 3
Y
X
R_B181.00
R_m95.00
R_b26.24
ω_1, η_1
ω_2, η_2
ω_3, η_3
M1
M2
M3


L_F 215.00
Φ_i
L_A 74.00
E_i
Y
Z


85°
18°
offset
Rc 86.00 +0.45
193.72
120.00
97.13
Rp 10.40 +0.45

Apéndice D

Script de Matlab® para cargar dimensiones del Robot tipo Delta

```
%  REMEDI PARAMETER MODEL SETTINGS V2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Generic Multilayer Robot control interface.
% This configuration file is divided into layer from the
% Lowest level of control (the driver) to higher
% control layer
%
% For each layer information have been divided into two groups
% Information exchangeable between blocks and with the remote
% system, and information used in basic configuration, marked as
% parameters in the file definition
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Loading Initial Configuration for the REMEDI DEVICE')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Kinematics Device Information
%
% This section is device specific and relates the design
% Parameters to the transmission
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Definition of the reference systems
% O: Center of Reference (Incenter of J1,J2,J3 joint axes)
% Z.axis Vertical
% X.axis horizontal, Directed versus Joint 1
% Joint 1 (SAID FI1) Rotation on motor 1 capstan, zero when horizontal
% Joint 2 (SAID FI2) Rotation on motor 2 capstan, zero when horizontal
% Joint 3 (SAID FI3) Rotation on motor 3 capstan, zero when horizontal
disp('Setting the Device information')
DELTAr = struct;
DELTAr.KDI = struct;
DELTAr.JointOffset    = 18*pi/180*[-1 -1 -1]; % [rad] FI offset at reset pos - offset
DELTAr.RadiusCapstan  =  single(86.0+0.45)   ; % [mm] CAD Measure + 1/2 rod thick - radio capstan + 1/2 diametro de cable de acero
DELTAr.Radius_pm      =  single((20.8+0.9)/2); % [mm] CAD Meas. + 1/2 Rod thick - radio polea + 1/2 diametro de cable de acero
DELTAr.LegDown        = 0.074   ; % [m] Lower leg length - radio de capstan
DELTAr.LegUP          = 0.217  ; % [m] Upper leg length - longitud de barras paralelas (215 mm en CAD), se coloco el fisico aproximado
DELTAr.RadiusCircleDW =  0.095  ; % [m] Lower Radius formed by joints centers - radio o distancia de base movil a eje motor
%DELTAr.RadiusCircleUP =  95    ; % [mm] Radius Disk Joint2 - no se utiliza

DELTAr.KDI.Km2j = single(-DELTAr.RadiusCapstan/DELTAr.Radius_pm*eye(3));
DELTAr.KDI.Kj2m = single(inv(DELTAr.KDI.Km2j)); % Joint2Motor Transmission


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Kinematics Filter Information
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

DELTAr.DOFLoad       = 0.000; %0.226; %.....peso de dispositivo extra (disp. para ecografia) en el EE para compensar gravedad y firccion
DELTAr.PlateW        = 0.030; %[Kg] Moving plate & payload mass - base movil
DELTAr.UpperCouplerW = 0.006; %[Kg] Upper wrist joint - Eje (articulacion) superior de ensamble con barras paralelas
DELTAr.LowerCouplerW = 0.006; %[Kg] Lower Elbow joint - Eje (articulacion) inferior de ensamble con barras paralelas
```

```
63  DELTAr.ArmMass       = 0.044; %[Kg] Two identical bars (22gr each) - barras paralelas
64  DELTAr.CapstanW      = 0.084; %[Kg] capstan
65  DELTAr.Capstan_gPOS  = [-0.00065, 0.03037]; %[m] ref frame in CAD snapshot - centro de
    masa de capstan
66  DELTAr.Capstan_elbowPOS = [0.000, 0.074]; % [m] same as lower leg
67
68  % Internal computation for gravity compensation each forearm mass is split
69  % in two halves, one is assumed located at the elbow joint, the second
70  % on the wrist. Each joint directly compensates for the elbow equivalent
71  % masses plus the projection to the joint of a reamining gragitational
72  %force assumed applied in the coupler.
73
74  DELTAr.UPmass = DELTAr.PlateW + DELTAr.DOFLoad +...
75      3*(DELTAr.UpperCouplerW + DELTAr.ArmMass/2); % this is mass the stick apparent mass
76
77  DELTAr.JointMass = DELTAr.LowerCouplerW + DELTAr.CapstanW + DELTAr.ArmMass/2;
78  DELTAr.JointCOG  = ...
79      (DELTAr.Capstan_gPOS*DELTAr.CapstanW+...
80            DELTAr.Capstan_elbowPOS*(DELTAr.LowerCouplerW + DELTAr.ArmMass/2))/...
81            DELTAr.JointMass;
82
83  DELTAr.M0    = DELTAr.JointMass * norm(DELTAr.JointCOG);
84  DELTAr.FIcog= atan2(DELTAr.JointCOG(1),DELTAr.JointCOG(2));
85
86
87
88  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89  %
90  % Kinematics Filter Information
91  %
92  %
93  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94
95  DELTAr.KFI = struct;
96  DELTAr.KFI.dqFiltFreq = 100;          %[Hz]
97  DELTAr.KFI.EofFiltFreq = 100;         %[Hz]
98  DELTAr.KFI.JointPosRefFiltFreq = 20; %[Hz]
99  DELTAr.KFI.EofPosRefFiltFreq = 20;    %[Hz]
100
101 %DELTAr
102
103
104 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105 %
106 % Kinematics Setting information
107 %
108 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
109
110 disp('Setting the Kinematic information')
111
112 KS = struct;
113 KS.V = struct;
114 KS.Description = 'Kinematic setting structure mapped to Datastore';
115 KS.V.qm_pos = single([0 0 0]);
116 KS.V.qm_vel = single([0 0 0]);
117 KS.V.q_pos = single([0 0 0]);
118 KS.V.q_vel = single([0 0 0]);
119 KS.V.EE_pos = single([0 0 0]);
120 KS.V.EE_vel = single([0 0 0]);
121 KS.V.EE_abs_pos = single([0 0 0]);
122 KS.V.qm_offset   = single( DELTAr.JointOffset*DELTAr.KDI.Km2j ); % [rad]
123
124 KP = struct;
125 KP.Description = 'Kinematic Params structure used to config';
126 KP.ROT_Loc2Base = single(eye(3));
127 KP.ROT_Base2Abs = single(eye(3));
128 KP.POS_Loc2Base = single([0 0 0]);
129 KP.POS_Base2Abs = single([0 0 0]);
130 KP.KDI = DELTAr.KDI; % Definition of Relevant Device Information
```

```
KP.KFI = DELTAr.KFI; % Definition of Relevant position filters
KP.DEV = DELTAr;

%KS
%KP

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Parametros motor.
% Constante motor(K)
%
Motors = struct;
Motors.k = [0.1541 0.1704 0.1765];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Create bus objects into the MATLAB base workspace
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SimulinkBUScellInfo =  { ...
  { ...
    'CMD_RefType', ...
    '', ...
    '', ...
    'Auto', ...
    '-1', {...
{'SysEnCmd', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'SysRunCmd', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'ModeCmd', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'GravCompEnCmd', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'FrictCompEnCmd', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'EofForceRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'JointPosRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'JointVelRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'EofVelRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'EofPosRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'JointTorqueRef', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'MotorDutyCycleCalib', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
    } ...
  } ...
  { ...
    'KinematicsType', ...
    '', ...
    '', ...
    'Auto', ...
    '-1', {...
{'qm_pos', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('rad'), sprintf(
'This is the encoder joint position')}; ...
{'qm_vel', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('rad'), sprintf(
'This is the encoder joint velocity')}; ...
{'q_pos', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('m'), sprintf(
'This is the position of robot joints')}; ...
{'q_vel', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('m'), sprintf('The
positionvelocity of robot joints')}; ...
{'EE_pos', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('m'), sprintf(
'This is the position of the EE in the robot defined coordinates')}; ...
{'EE_vel', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('m/s'), sprintf(
'This is the velocity of the EE in robot coordinates')}; ...
{'EE_abs_pos', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('m'), sprintf(
'Absolute position of the EE')}; ...
{'qm_offset', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], sprintf('rad'), sprintf
('Value to show when reading is ZERO qm= qm_raw+qm_offset')}; ...
    } ...
  } ...
  { ...
    'MonitorType', ...
    '', ...
    '', ...
```

```
    'Auto', ...
    '-1', {...
{'Tm', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'PWMref', 3, 'single', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'DriverEn', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'CurSat', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'VelSat', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'CmdRxTimestamp', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'SysErr', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
{'SysStatus', 1, 'int16', -1, 'real', 'Sample', 'Fixed', [], [], '', ''}; ...
    } ...
  } ...
}';

Simulink.Bus.cellToObject(SimulinkBUScellInfo)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Remove Temporary created variables
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear SimulinkBUScellInfo;
clear DELTAr;
qm_offset = single(0);

% MODEL PARAMETER SETTINGS
```

Apéndice E

Script para validación de la cinemática directa.

```
 function [Xee, Jinv, Jd] = KinDelta(fi,DEBUG)
% this function computes the delta direct and differential
% cinematics of the delta 3D using a direct straightforward
% solution of the delta internal triangle defined as S1, S2, S3

%
%
persistent Ldown;
persistent Lup;
persistent Rdown;
persistent Rup;
persistent Rint;
persistent CTHETA;
persistent STHETA;
persistent Rot1;
persistent Rot2;
persistent Rot3;
persistent SC1;
persistent SC2;
persistent SC3;
% persistent RotZ120;
% persistent RotZ120_;

if (isempty(Ldown))
    % Initialize delta geometric constants
    Ldown = single(0.074);  % The arm connected to the base
    Lup   = single(0.217); % The forearm connected to the top coupler (se ingreso el
    valor fisico aproximado)
    Rdown = single(0.095);  % Base Radius
    Rup   = single(0.02624); % Coupler Radius

    Rint  = Rdown -Rup ;   % Delta Radius

    CTHETA = single(-0.5); % cos (2*pi/3);
    STHETA = single(8.660254037844387e-01); % sin(2*pi/3)


    % First we compute the rotational Versors:
    %S = @(w) [0 -w(3) w(2); w(3) 0 -w(1); -w(2) w(1) 0];
    Rot1  = -[0 1 0]';           % -S([0 0 1])*[1 0 0]';
    Rot2  = -[-STHETA CTHETA 0]'; %-S([0 0 1])*[CTHETA STHETA 0]';
    Rot3  = -[ STHETA CTHETA 0]'; %-S([0 0 1])*[CTHETA -STHETA 0]';

     SC1 = Rint*[CTHETA; STHETA; 0];
     SC2 = Rint*[CTHETA; -STHETA; 0];
     SC3 = Rint*[1; 0; 0];
end


% First we compute S1..3 location

% S1 is supposed aligned on the X axis
% S2 is rotated by THETA   = 2*pi/3
% S3 is rotated by -THEHTA = -2*pi/3

%
cf1 = cos(fi(1));
sf1 = sin(fi(1));
cf2 = cos(fi(2));
sf2 = sin(fi(2));
cf3 = cos(fi(3));
sf3 = sin(fi(3));


S01 = single( [ ...
     Ldown*cf1; ... %X component
               0           ; ... %Y component
    Ldown*sf1; ... %Z component
]);
```

```
69
70
71  S1 = single( S01 + [Rint; 0; 0]);
72
73
74  S02 = single ( Ldown * [ ...
75      cf2*CTHETA; ... %X component
76      cf2*STHETA; ... %X component
77      sf2; ... %Z component
78  ]);
79
80  S2 = single ( S02 + Rint*[CTHETA; STHETA; 0 ] );
81
82  S03 = single( Ldown*[ ...
83      cf3*CTHETA; ... %X component
84      -cf3*STHETA; ... %X component
85      sf3; ... %Z component
86  ]);
87
88  S3 = single ( S03 + Rint*[CTHETA; -STHETA; 0 ] );
89
90  % Now compute the internal triangle vector with respect to S1
91  S12 = S2-S1;
92  S13 = S3-S1;
93
94  % INCIRC is the internal circumference center it is achieved by
95  % solving the relationships:
96  % S12' (INCIRC-S12/2) = 0
97  % S13' (INCIRC-S13/2) = 0
98  % Having a minimum norm!
99
100 d11 = S12'*S12;
101 d33 = S13'*S13;
102 d13 = S12'*S13;
103
104 % We are solving the problem:
105 %  [   S12'   ] INCIRC = S12'*S12/2
106 %  [   S13'   ]        = S13'*S13/2
107 %
108 % Usinf t e pseudinverse on S1X matrix said A
109 %
110 % INCIRC =  A' inv(A*A')
111 %
112 % Where the inverse symmetrix matrix terms are:
113 %
114 %   d11 = S12'*S12
115 %   d13 = S12'*S13
116 %   d33 = S13'*S13
117 %
118 % We split the peusoinverse computation in two parts:
119 %  inv([d11 d13;d13 d33])*[d11;d33]
120 %    ans =
121 %    (d11*d33)/(- d13^2 + d11*d33) - (d13*d33)/(- d13^2 + d11*d33)
122 %    (d11*d33)/(- d13^2 + d11*d33) - (d11*d13)/(- d13^2 + d11*d33)
123
124 DET = d11*d33-d13*d13;
125
126 P1 =  d33*(d11 - d13)/(DET);
127 P2 =  d11*(d33 - d13)/(DET);
128
129 % This is the position of the internal circumpherence
130 % By construction it is already expressed w.r.t S1
131 % So its norm is the radius of the circumpherence
132 INCIRC = (S12*P1 + S13*P2)*0.5;
133
134 % Now we compute the orthogonal ray and position
135 Norm = cross(S12,S13);
136
137 % DIST S1 to INCIRC = S1I
```

```
k = (Lup^2 - (INCIRC' * INCIRC) )/ (Norm'*Norm);
Xee = S1 + INCIRC + sqrt(k)*Norm;
%Xee = S1 + INCIRC + sqrt(complex(x))*Norm;


% Differential cinematic analysis, we base our computation
% based on the length of the upper arms which are constant:
%
% NAMELY
%    norm(Xee-S1) = Lup
%    or in a squared form:
%    (Xee-S1)' (Xee-S1) = Lup^2
% If we differentiate we have:
%
%    2*(Xee-S1)' (Xee_dot -S1_dot) = 0
%
% Which led us to the following fundamental relaitonship
%
%  (Xee-S1)' Vee = Xee' * S1_dot
%
% Where we eliminated the null product S1' S1_dot
% And replacind S1_dot with S1orth * q1_dot we have
%
% (Xee-S1)' Vee = Xee' * S1orth q_dot
%
% We have
%    q_dot  = 1/(Xee'S1orth)* (Xee-S1)' Vee
%
% Which is the first row of the inverse Jacobian

Jinv= single(zeros(3,3));

% (Pee-S1)/(Pee dot S1Ortho)
Jinv(1,:) = (Xee-S1)'/((Xee-S1)' * cross(Rot1, S01));
Jinv(2,:) = (Xee-S2)'/((Xee-S2)' * cross(Rot2, S02));
Jinv(3,:) = (Xee-S3)'/((Xee-S3)' * cross(Rot3, S03));


% Direct Differential cinematic
%
%
Jd= single(zeros(3,3));
dir1= cross((Xee-0.5*(S2+S3)),(S3-S2));
Jd(:,1) = dir1 /(Jinv(1,:) * dir1);

dir2= cross((Xee-0.5*(S3+S1)),(S1-S3));
Jd(:,2) = dir2 /(Jinv(2,:) * dir2);

dir3= cross((Xee-0.5*(S1+S2)),(S2-S1));
Jd(:,3) = dir3 /(Jinv(3,:) * dir3);

if DEBUG    % Kinematic diagnosis
    % Drad a 3D plot about the vectors in the match

    drawArrow = @(a,b,varargin) quiver3(a(1),a(2),a(3),b(1)-a(1),b(2)-a(2),...
        b(3)-a(3),0,varargin{:} );

    xMax = 0.15;
    yMax = xMax;
    zMax = 0.28;
    OO = [0 0 0];
    XX = [1 0 0].*xMax;
    YY = [0 1 0].*yMax;
    ZZ = [0 0 1].*zMax;
    % Draw ref frame
    figure(4),
    drawArrow(OO,XX,'linewidth',2,'LineStyle','--','color','k'); hold on;
    drawArrow(OO,YY,'linewidth',2,'LineStyle','--','color','k');
    drawArrow(OO,ZZ,'linewidth',2,'LineStyle','--','color','k');
```

```
%     drawArrow(OO,SC1,'linewidth',3,'color','r');
%     drawArrow(SC1,S1,'linewidth',3,'color','g');

    drawArrow(S1,S2,'linewidth',2,'LineStyle','--','color','b');
    drawArrow(S1,S3,'linewidth',2,'LineStyle','--','color','b');
    points= [S1'; S2'; S3'];
    fill3(points(:,1),points(:,2),points(:,3),'r');
    alpha(0.3);
    clear points

    drawArrow(S1,Xee,'linewidth',1,'LineStyle','--','color','b');
    drawArrow(S2,Xee,'linewidth',1,'LineStyle','--','color','b');
    drawArrow(S3,Xee,'linewidth',1,'LineStyle','--','color','b');

    drawArrow(S1,S1+INCIRC,'linewidth',2,'LineStyle','--','color','c');
    drawArrow(S1+INCIRC,Xee,'linewidth',2,'color','m');
    drawArrow(S1+INCIRC,Norm.*5,'linewidth',2,'LineStyle','--','color','k');
    % drawArrow(SC1,S1,'linewidth',3,'LineStyle','--','color','g');
%     drawArrow(OO,SC2,'linewidth',3,'color','r');
%     drawArrow(SC2,S2,'linewidth',3,'color','g');
%     drawArrow(OO,SC3,'linewidth',3,'color','r');
%     drawArrow(SC3,S3,'linewidth',3,'color','g');
    xlabel('Eje x (m)'),xlim([-xMax,xMax]);
    ylabel('Eje y (m)');ylim([-yMax,yMax]);
    zlabel('Eje z (m)');

    % - Riprova per gli altri S -
    % Now compute the internal triangle vector with respect to S2
    S21 = S1-S2;
    S23 = S3-S2;
    d11 = S21'*S21;
    d33 = S23'*S23;
    d13 = S21'*S23;
    DET = d11*d33-d13*d13;
    P1 =  d33*(d11 - d13)/(DET);
    P2 =  d11*(d33 - d13)/(DET);
    INCIRC2 = (S21*P1 + S23*P2)*0.5;
    % Now we compute the orthogonal ray and position
    Norm2 = cross(S23,S21);
    % DIST S1 to INCIRC = S1I
    k = (Lup^2 - (INCIRC2' * INCIRC2) )/ (Norm2'*Norm2);
    Xee2 = (S2 + INCIRC2 + sqrt(k)*Norm2).*[1 1 0.80]';

    % Now compute the internal triangle vector with respect to S3
    S31 = S1-S3;
    S32 = S2-S3;
    d11 = S31'*S31;
    d33 = S32'*S32;
    d13 = S31'*S32;
    DET = d11*d33-d13*d13;
    P1 =  d33*(d11 - d13)/(DET);
    P2 =  d11*(d33 - d13)/(DET);
    INCIRC3 = (S31*P1 + S32*P2)*0.5;
    % Now we compute the orthogonal ray and position
    Norm3 = cross(S31,S32);
    % DIST S1 to INCIRC = S1I
    k = (Lup^2 - (INCIRC3' * INCIRC3) )/ (Norm3'*Norm3);
    Xee3 = (S3 + INCIRC3 + sqrt(k)*Norm3).*[1 1 0.80]';

    %%  Ploting text and origens for visualization
%     plot3(S1(1),S1(2),S1(3),'o','MarkerSize',10,'LineWidth',3);
%     plot3(S2(1),S2(2),S2(3),'o','MarkerSize',10,'LineWidth',3);
%     plot3(S3(1),S3(2),S3(3),'o','MarkerSize',10,'LineWidth',3);
%     plot3(Xee(1),Xee(2),Xee(3),'*','MarkerSize',10,'LineWidth',3);

    S1_ = double([S1(1),S1(2),S1(3)]);
    S2_ = double([S2(1),S2(2),S2(3)]);
    S3_ = double([S3(1),S3(2),S3(3)]);
    Xee_ = double([Xee(1),Xee(2),Xee(3)]);
```

```
276
277       % se sustituye S1, S2 y S3 por C1, C2 y C3 para representacion en
278       % español
279       text((S1_(1)),S1_(2),S1_(3),'C1','Fontsize',12);
280       text((S2_(1))-0.005,S2_(2)+0.01,S2_(3),'C2','Fontsize',12);
281       text((S3_(1)),S3_(2),S3_(3),'C3','Fontsize',12);
282       text(Xee_(1),Xee_(2),Xee_(3),'EF','Fontsize',12);
283
284       %%
285       % drawArrow(S2,S1,'linewidth',2,'LineStyle','--','color','y');
286       % drawArrow(S2,S3,'linewidth',2,'LineStyle','--','color','y');
287       drawArrow(S2,S2+INCIRC2,'linewidth',2,'LineStyle','--','color','c');
288      % drawArrow(OO,Xee2,'linewidth',1,'LineStyle','--','color','m');
289
290       % drawArrow(S3,S1,'linewidth',2,'LineStyle','--','color','y');
291       % drawArrow(S3,S2,'linewidth',2,'LineStyle','--','color','y');
292       drawArrow(S3,S3+INCIRC3,'linewidth',2,'LineStyle','--','color','c');
293       %drawArrow(OO,Xee3,'linewidth',1,'LineStyle','--','color','m');
294   end
```

```m
%% plot_Kinematis
close all
clc

%% Plot para generar 3 posiciones diferentes
% se ingresa el angulo de cada capstan para representar el robot Delta y su
% efector final
KinDelta_RS([0*pi/180,0*pi/180,0*pi/180],1);

subplot(1,3,1);
KinDelta_RS([-18*pi/180,-18*pi/180,-18*pi/180],1);
title('Posición retraida (Condición 1)');

subplot(1,3,2);
KinDelta_RS([60*pi/180,10*pi/180,30*pi/180],1);
title('Posición inclinada (Condición 2)');

subplot(1,3,3);
KinDelta_RS([85*pi/180,85*pi/180,85*pi/180],1);
title('Posición extendida (Condición 3)');

```

Apéndice F

Software STM32CubeIDE

```c
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2023 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "adc.h"
#include "dma.h"
#include "tim.h"
#include "usart.h"
#include "gpio.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */

#include <stdio.h>
#include <string.h>
#include <stdlib.h> // libreria para interrupciones en flotantes
#include <math.h>
#include "..\..\simulink\Derivada_vel_ert_rtw\Derivada_vel.c" // direccion de la carpeta simulink para obtener la velocidad de los motores
#include "..\..\simulink\Control_posicion_gral_ert_rtw\Control_posicion_gral.c" // direccion de la carpeta simulink para control de los motores
#include "..\..\simulink\Cinematica_DeltaR_ert_rtw\Cinematica_DeltaR.c" // direccion de la carpeta simulink para cinematica del robot
#include "..\..\simulink\val_DeltaEE_ert_rtw\val_DeltaEE.c" // direccion de la carpeta simulink para validacion de posicion del EE y capstan en estados 0, 1 y 2

/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

#define pi 3.1416
#define size 100  //para el ciclo for
#define PPR_enc 1600 // PPR a 400 * 4 (cuadratura)
#define N_ENC_16 3 //represents the number of encoders of 16 bits

#define length 3 // numero de canales de ADC
//#define NBSAMPLES 13 // constante define el orden del filtro
//#define N_ADC 3 // numero de canales de adc para el filtro

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
```

```
66
67  /* USER CODE BEGIN PV */
68
69  uint8_t TEXT[200];
70  uint16_t ADC1_buffer[3]; // variable tipo vector para cada canal de ADC
71  uint16_t adc1_ch4, adc1_ch5, adc1_ch6; // Variables para pasar el valor del Buffer del
    ADC
72  uint8_t Btn_user, CONTADOR = 0, estado = 0;
73
74  int32_t cnt_enc1,cnt_enc2,cnt_enc3; //variable entero con signo para lectura de encoder
75  float cnt_M1,cnt_M2,cnt_M3; //variables para lectura de motor
76  uint8_t dir1,dir2,dir3; // variables para direccion de giro (sentido horario y
    antihorario)
77  float pos_gradM1, pos_gradM2, pos_gradM3; // variables para la posicion del motor en
    grados
78  float pos_radM1,pos_radM2,pos_radM3; // variables para la posicion del motor en radianes
79  float pos_deseadaM1 = 0, pos_deseadaM2 = 0, pos_deseadaM3 = 0; // variables para
    posicion del motor
80  float PosJoint_1,PosJoint_2,PosJoint_3; // variables para posicion de capstan
81  float PosEE_x,PosEE_y,PosEE_z; // variables para posicion del efector final
82  float duty1,duty2,duty3; Variables para mapeo de la señal de control
83  uint16_t pwm1 = 0, pwm2 = 0, pwm3 = 0; // variables para pwm
84  float V1, V2, V3; // variable de voltaje en cada motor
85  uint16_t cnt_Vel = 0;
86
87  float vel_rpm_m1,vel_rpm_m2,vel_rpm_m3,vel_rads_m1,vel_rads_m2,vel_rads_m3,K_m1,K_m2,K_m3
    ; // varibles para velocidad y constante de motor (K)
88  float velEnc1,velEnc2,velEnc3; // variables para velocidad de encoder
89
90  // variables para interrupcion en terminal serial
91  uint8_t Rx_data[size];
92  uint8_t Rx_indx;
93  uint8_t Rx_Buffer[size];
94  uint8_t Transfer_cplt;
95
96  //variables para ingresar la posicion de cada motor o EE separado por "," segun sea el
    caso
97  const int16_t cadena_length = 4; //write numbers of string in the chain separeted by
    comma
98  float *ptr;  //lo mismo a float ptr[];
99
100 uint8_t flag, flag_vel, activa, EN_Driver; // banderas para activacion de condiciones
101
102 /* USER CODE END PV */
103
104 /* Private function prototypes -----------------------------------------------*/
105 void SystemClock_Config(void);
106 /* USER CODE BEGIN PFP */
107
108 //Use this function when TIM2 and/or TIM5 (have 32 bits) are activated as Encoder
109 int32_t encoder_read_32_bits(TIM_HandleTypeDef *htim, uint8_t reset, uint8_t *dir) //
    ENCODER 1
110 {
111     /*--- TIM2 or TIM5 are timer of 32 bits ---*/
112     /*--- NOTE: In order to use TIM5 as Encoder SB13 bridge must be removed, but
        Ethernet will not work ---*/
113 #if (htim == htim2 || htim == htim5) //"#if preprocessor directive"
114     static int32_t cnt_enc=0;
115     cnt_enc = __HAL_TIM_GET_COUNTER(htim);
116     *dir = __HAL_TIM_IS_TIM_COUNTING_DOWN(htim);
117
118     if (reset)
119     {
120         __HAL_TIM_SET_COUNTER(htim,0);
121     }
122
123     return cnt_enc;
124
125 #endif
```

```
126 }
127
128 //Use this function when Timers have 16 bits and are activated as Encoder
129 int32_t encoder_read_16_to_32_bits_num_enc(TIM_HandleTypeDef *htim, uint8_t num_enc,
    uint8_t reset, uint8_t *dir) // Encoder 2 y Encoder 3
130 {
131     /*--- num_enc should start with cero ---*/
132     if (num_enc>N_ENC_16) return 0;
133     static int32_t cnt_enc[N_ENC_16]={0};
134     static int32_t curr[N_ENC_16]={0};
135     static int32_t old[N_ENC_16]={0};
136     //int32_t cnt_enc=0;
137     //int32_t curr=0,old=0;
138     old[num_enc] = curr[num_enc];
139     curr[num_enc] = (uint16_t)__HAL_TIM_GET_COUNTER(htim);
140     cnt_enc[num_enc]+= (int16_t)(curr[num_enc]-old[num_enc]); // Extend from 16 to 32
    bits
141     *dir = __HAL_TIM_IS_TIM_COUNTING_DOWN(htim);
142     if(reset)
143     {
144         cnt_enc[num_enc] = 0;
145     }
146     return cnt_enc[num_enc];
147 }
148
149 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
150 {
151     uint8_t i;
152 // Prevent unused argument(s) compilation warning
153    UNUSED(huart);
154
155     if (huart!= &huart3)
156         return;
157     HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);//eco con Led verde cada que se presiona una
    tecla
158
159     if (huart -> Instance == USART3)//current UART
160     {
161         if (Rx_indx==0) {
162             for (i=0;i<size;i++)
163                 Rx_Buffer[i]=0;
164         }   //clear Rx_Buffer before receiving new data
165
166         if (Rx_data[0]!=13) //if received data different from ascii 13 (enter button)
167             {
168             Rx_Buffer[Rx_indx++] = Rx_data[0];    //add data to Rx_Buffer
169             }
170         else            //if received data = 13
171             {
172             Rx_indx=0;
173             Transfer_cplt=1;//transfer complete, data is ready to read
174             }
175
176         HAL_UART_Receive_IT(&huart3, Rx_data, 1);   //activate UART receive interrupt
        every time
177         //HAL_UART_Transmit(&huart3, (uint8_t *)Rx_data, strlen(Rx_data),100);  //echo
        to display the commands on the terminal
178                                                                 //(for the length use
                                                                    strlen or sizeof only
                                                                    for basic UART)
179     }
180 }
181
182 // funcion para ingresar posicion a motores indivudualmente
183 float *split(const int16_t cadena_len) {
184     //char str[80] = "123,456,789"; //for test
185     const char s[] = ","; //symbol to separate
186     char *newcadena;
187     float *cadena=0;
```

```
188       int i=0;
189
190       /* get the first token */
191       newcadena = strtok(Rx_Buffer, s);
192       //HAL_Delay(1);
193       //cadena[0] = atof(newcadena); //atoi to int;  atof to float
194       cadena[0] = strtof(newcadena,NULL);
195       i++;
196
197           while(newcadena !=0 && i<=cadena_len) {
198               newcadena = strtok(NULL, s);
199               //HAL_Delay(1);
200               //cadena[i] = atof(newcadena); //atoi to int;  atof to float
201               cadena[i] = strtof(newcadena,NULL);
202               i++;
203
204           }
205
206          i=0;
207          return cadena;
208   }
209
210   // funcion para activacion y desactivacion de drivers
211   void parseline(void)
212   {
213       uint8_t TEXT_local[100];
214
215       //pos_deseadaM1 = atof(Rx_Buffer);//cambio variable string a tipo flotante
216       //estado = atoi(Rx_Buffer); //cambio variable string a tipo entero
217
218       if (EN_Driver)
219       {
220           HAL_GPIO_WritePin(Output_driver1_GPIO_Port, Output_driver1_Pin, 1); // activa el
              driver 1
221           HAL_GPIO_WritePin(Output_driver2_GPIO_Port, Output_driver2_Pin, 1); // activa el
              driver 2
222           HAL_GPIO_WritePin(Output_driver3_GPIO_Port, Output_driver3_Pin, 1); // activa el
              driver 3
223       }
224       else
225       {
226           HAL_GPIO_WritePin(Output_driver1_GPIO_Port, Output_driver1_Pin, 0); // desactiva
              el driver 1
227           HAL_GPIO_WritePin(Output_driver2_GPIO_Port, Output_driver2_Pin, 0); // desactiva
              el driver 2
228           HAL_GPIO_WritePin(Output_driver3_GPIO_Port, Output_driver3_Pin, 0); // desactiva
              el driver 3
229       }
230       //return EN_Driver;
231   }
232
233   //funcion para mapeo
234   float map(float x, float in_min, float in_max, float out_min, float out_max) {
235     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
236   }
237
238   // funcion para correr programa a diferentes frecuancias
239   void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
240   {
241       static int16_t cnt1 = 0;
242       static int16_t cnt2 = 0;
243       static int16_t cnt3 = 0;
244
245       if (htim -> Instance == TIM6)
246       {
247           //inician los contadores
248           cnt1++;
249           cnt2++;
250           cnt3++;
```

```
251         //::::::::::::::::::::::::::::::frecuencia de muestreo a 1 KHz en esta
            seccion::::::::::::::::::::::::::::::::::::
252
253         //HAL_GPIO_TogglePin(LED_ROJO_GPIO_Port, LED_ROJO_Pin);
254
255         //adc1_ch3 = rr_fir_filter(0, ADC1_buffer[0]);
256         adc1_ch4 = ADC1_buffer[0]; // driver M1
257         adc1_ch5 = ADC1_buffer[1]; // driver M2
258         adc1_ch6 = ADC1_buffer[2]; // driver M3
259         //adc1_ch4 = rr_fir_filter(1, ADC1_buffer[1]);
260         V1 = adc1_ch5*3.296/4095;
261         V2 = adc1_ch6*3.296/4095;
262         V3 = adc1_ch4*3.296/4095;
263
264         estado = (uint8_t)ptr[0]; // variable para seleccionar estados del robot (modo
            calibracion o control)
265
266         //:::::::::::::::::: encoder original ::::::::::::::::::
267         //pos_grad = (cnt_enc1/(N*PPR))*360; el datasheet proporciona el PPR en
            cuadratura, PPR = 211.2
268
269         //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Inicio seccion para motor
            M1<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
270
271         cnt_enc1 = encoder_read_32_bits(&htim2, Btn_user, &dir1);
272         //cnt_M1 = float(cnt_enc1 / 4.4);
273
274         //pos_gradM1 = (360*cnt_enc1)/211.2; // para encoder original del motor
275         pos_gradM1 = ((-360*cnt_enc1)/PPR_enc)/4.4; // PPR = 400*4, nota: se divide la
            posicion del encoder entre el gearbox para conocer la posicion del motor
276         //pos_radM1 = ((cnt_enc1/PPR_enc)*-2*pi)/4.4;
277         pos_radM1 = (pos_gradM1*pi)/180;
278         //pos_rad = (pos_grad*pi)/180;
279         //pos_deseadaM1 = ptr[1]; // posicion deseada para motor 1
280
281         // >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Fin seccion para motor
            M1<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
282
283         // >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Inicio seccion para motor
            M2<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
284
285         cnt_enc2 = encoder_read_16_to_32_bits_num_enc(&htim3, 0, Btn_user, &dir2);
286         //cnt_M2 = float(cnt_enc2 / 4.4);
287
288         //pos_gradM2 = (360*cnt_enc2)/211.2; // para encoder original del motor
289         pos_gradM2 = ((-360*cnt_enc2)/PPR_enc)/4.4; // PPR = 400*4 nota: se divide la
            posicion del encoder entre el gearbox para conocer la posicion del motor
290         //pos_radM2 = ((cnt_enc2/PPR_enc)*-2*pi)/4.4;
291         pos_radM2 = (pos_gradM2*pi)/180;
292         //pos_deseadaM2 = pos_deseadaM1;
293         //pos_deseadaM2 = ptr[2]; // posicion deseada para motor 2
294
295         //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Fin seccion para motor
            M2<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
296
297         //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Inicio seccion para motor
            M3<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
298
299         cnt_enc3 = encoder_read_16_to_32_bits_num_enc(&htim4, 1, Btn_user, &dir3);
300         //cnt_M3 = float(cnt_enc3 / 4.4);
301
302         //pos_gradM3 = (360*cnt_enc3)/211.2; // para encoder original del motor
303         pos_gradM3 = ((-360*cnt_enc3)/PPR_enc)/4.4; // PPR = 400*4 nota: se divide la
            posicion del encoder entre el gearbox para conocer la posicion del motor
304         //pos_radM3 = ((cnt_enc3/PPR_enc)*-2*pi)/4.4;
305         pos_radM3 = (pos_gradM3*pi)/180;
306         //pos_deseadaM3 = pos_deseadaM1;
307         //pos_deseadaM3 = ptr[3]; // posicion deseada para motor 3
308
```

```
309         // >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Fin seccion para motor
            M3<<
```

```
370             Derivada_vel_U.entrada_pos3 = pos_gradM3; // posicion de entrada - M3
371
372             val_DeltaEE_U.PosM1 = pos_gradM1;
373             val_DeltaEE_U.PosM2 = pos_gradM2;
374             val_DeltaEE_U.PosM3 = pos_gradM3;
375
376             // ***************** FUNCION STEP DE SIMULINK ************************
377
378             Derivada_vel_step(); // funcion simulink
379             val_DeltaEE_step();
380
381             // ************** SALIDAS SIMULINK *******************
382             // :::::::::::::::::::: MOTOR 1 ::::::::::::::::::::
383
384             vel_rads_m1 = Derivada_vel_Y.salida_vel_1; // velocidad en rad/s
385             vel_rpm_m1 = vel_rads_m1 * 30/pi;
386             velEnc1 = vel_rpm_m1 * 4.4;
387             K_m1 = Derivada_vel_Y.K_motor1; // constante del motor 1
388
389             // :::::::::::::::::::: MOTOR 2 ::::::::::::::::::::
390
391             vel_rads_m2 = Derivada_vel_Y.salida_vel_2; // velocidad en rad/s
392             vel_rpm_m2 = vel_rads_m2 * 30/pi;
393             velEnc2 = vel_rpm_m2 * 4.4;
394             K_m2 = Derivada_vel_Y.K_motor2; // constante del motor 2
395
396             // :::::::::::::::::::: MOTOR 3 ::::::::::::::::::::
397
398             vel_rads_m3 = Derivada_vel_Y.salida_vel_3; // velocidad en rad/s
399             vel_rpm_m3 = vel_rads_m3 * 30/pi;
400             velEnc3 = vel_rpm_m3 * 4.4;
401             K_m3 = Derivada_vel_Y.K_motor3; // constante del motor 3
402
403             PosJoint_1 = val_DeltaEE_Y.Joint_pos[0]; //posicion de capstan 1
                (articulacion 1) en grados
404             PosJoint_2 = val_DeltaEE_Y.Joint_pos[1]; //posicion de capstan 2
                (articulacion 2) en grados
405             PosJoint_3 = val_DeltaEE_Y.Joint_pos[2]; //posicion de capstan 3
                (articulacion 3) en grados
406
407             PosEE_x = val_DeltaEE_Y.EE_pos[0]; //posicion de EE x
408             PosEE_y = val_DeltaEE_Y.EE_pos[1]; //posicion de EE y
409             PosEE_z = val_DeltaEE_Y.EE_pos[2]; //posicion de EE z
410
411             // Cuando se envia 0 el giro es horario (negativo), 1 para giro antihorario
                (positivo)
412             HAL_GPIO_WritePin(Output_M1_GPIO_Port, Output_M1_Pin,0);
413             HAL_GPIO_WritePin(Output_M2_GPIO_Port, Output_M2_Pin,0);
414             HAL_GPIO_WritePin(Output_M3_GPIO_Port, Output_M3_Pin,0);
415
416             pwm1 = 700;
417             pwm2 = 790;
418             pwm3 = 700;
419
420             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1,pwm1);
421             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2,pwm2);
422             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3,pwm3);
423
424             flag_vel = (vel_rads_m1 == 0 && vel_rads_m2 == 0 && vel_rads_m3 == 0);
425
426             if (flag_vel)
427             {
428                 //HAL_GPIO_WritePin(LED_ROJO_GPIO_Port, LED_ROJO_Pin, 1);
429                 activa = 1;
430             }
431             else
432             {
433                 //HAL_GPIO_WritePin(LED_ROJO_GPIO_Port, LED_ROJO_Pin, 0);
434                 activa = 0;
```

```
435             }
436 
437             if (activa)
438             {
439                 cnt_Vel ++;
440                 if (cnt_Vel >= 500)
441                 {
442                     __HAL_TIM_SET_COUNTER(&htim2,0); // resetea encoders a cero
443                     __HAL_TIM_SET_COUNTER(&htim3,0);
444                     __HAL_TIM_SET_COUNTER(&htim4,0);
445 
446                     cnt_Vel = 500;
447                     EN_Driver = 0;
448                     parseline();
449                 }
450             }
451         }
452 
453 // ::::::::::::::::::::::: ESTADO 2 - CONTROL DE POSICION DEL ROBOT MEDIANTE POSICION DE MOTORES:::::::::::::::::::::
454 
455         if (estado == 2) // ingresa al estado en modo Control de robot
456         {
457             HAL_GPIO_WritePin(LED_ROJO_GPIO_Port, LED_ROJO_Pin, 0);
458             cnt_Vel = 0;
459             parseline();
460             EN_Driver = 1;
461 
462             /* La posicion deseada se multiplica por -1 debido a que se invirtio el giro en los motores,
463              * en sentido antihorario el giro es negativo y en sentido horario el giro es positivo.
464              * Asi podemos ingresar valores positivos mediante la interrupcion en la terminal serial
465              */
466             pos_deseadaM1 = (-1)*ptr[1];//-240; // se manda el M1 a la posicion 0
467             pos_deseadaM2 = (-1)*ptr[2];//-240; // se manda el M2 a la posicion 0
468             pos_deseadaM3 = (-1)*ptr[3];//-240; // se manda el M3 a la posicion 0
469 /*
470             pos_deseadaM1 = ptr[1]; // se manda el M1 a la posicion 0
471             pos_deseadaM2 = ptr[2]; // se manda el M2 a la posicion 0
472             pos_deseadaM3 = ptr[3]; // se manda el M3 a la posicion 0
473 */
474             // ****************** ENTRADAS SIMULINK ********************
475             // ::::::::::::::::::::::: MOTOR 1 :::::::::::::::::::::::
476 
477             Control_posicion_gral_U.pos_deseada_M1 = pos_deseadaM1;
478             Control_posicion_gral_U.pos_actual_M1 = pos_gradM1;
479 
480             // ::::::::::::::::::::::: MOTOR 2 :::::::::::::::::::::::
481 
482             Control_posicion_gral_U.pos_deseada_M2 = pos_deseadaM2;
483             Control_posicion_gral_U.pos_actual_M2 = pos_gradM2;
484 
485             // ::::::::::::::::::::::: MOTOR 3 :::::::::::::::::::::::
486 
487             Control_posicion_gral_U.pos_deseada_M3 = pos_deseadaM3;
488             Control_posicion_gral_U.pos_actual_M3 = pos_gradM3;
489 
490             val_DeltaEE_U.PosM1 = pos_gradM1; // motor 1
491             val_DeltaEE_U.PosM2 = pos_gradM2; // motor 2
492             val_DeltaEE_U.PosM3 = pos_gradM3; // motor 3
493 
494             // funcion step de simulink
495             Control_posicion_gral_step();
496             val_DeltaEE_step();
497 
498             // ************** SALIDAS SIMULINK ********************
499             // :::::::::::::::::::: MOTOR 1 ::::::::::::::::::::
```

```
500
501             duty1 = fabs( Control_posicion_gral_Y.SignalCont_torque_M1);
502             pwm1 = map(duty1, 0, 35, 75, 3599);
503
504             if(pwm1 > 3599)
505             {
506                 pwm1 = 3599;
507             }
508
509             // ************** motor 2 ********************
510
511             duty2 = fabs(Control_posicion_gral_Y.SignalCont_torque_M2);
512             pwm2 = map(duty2, 0, 40, 75, 3599);
513
514             if(pwm2 > 3599)
515             {
516                 pwm2 = 3599;
517             }
518
519             // ************** salidas motor 3 ********************
520
521             duty3 = fabs(Control_posicion_gral_Y.SignalCont_torque_M3);
522             pwm3 = map(duty3, 0, 30, 75, 3599);
523
524             if(pwm3 > 3599)
525             {
526                 pwm3 = 3599;
527             }
528
529             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1,pwm1);
530             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2,pwm2);
531             __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3,pwm3);
532
533             HAL_GPIO_WritePin(Output_M1_GPIO_Port, Output_M1_Pin,Control_posicion_gral_Y.
                Giro_M1); // giro para motor M1
534             HAL_GPIO_WritePin(Output_M2_GPIO_Port, Output_M2_Pin,Control_posicion_gral_Y.
                Giro_M2); // giro para motor M2
535             HAL_GPIO_WritePin(Output_M3_GPIO_Port, Output_M3_Pin,Control_posicion_gral_Y.
                Giro_M3); // giro para motor M3
536
537             PosJoint_1 = val_DeltaEE_Y.Joint_pos[0]; //posicion de capstan 1
                (articulacion 1) en grados
538             PosJoint_2 = val_DeltaEE_Y.Joint_pos[1]; //posicion de capstan 2
                (articulacion 2) en grados
539             PosJoint_3 = val_DeltaEE_Y.Joint_pos[2]; //posicion de capstan 3
                (articulacion 3) en grados
540
541             PosEE_x = val_DeltaEE_Y.EE_pos[0]; //posicion de EE x
542             PosEE_y = val_DeltaEE_Y.EE_pos[1]; //posicion de EE y
543             PosEE_z = val_DeltaEE_Y.EE_pos[2]; //posicion de EE z
544         }
545
546
547 // :::::::::::::::::::::::::::::::::: ESTADO 3 - CINEMATICA INVERSA DEL
    ROBOT::::::::::::::::::::::::::::::::::::::
548
549         if (estado == 3)
550         {
551             HAL_GPIO_WritePin(LED_ROJO_GPIO_Port, LED_ROJO_Pin, 1);
552             //Cinematica_DeltaR_U.estado = estado;
553             cnt_Vel = 0;
554             parseline();
555             EN_Driver = 1;
556
557             // ::::::::::::::::::::FUNCIONES SIMULINK:::::::::::::::::::::::::::::
558
559             // ********ENTRADAS*********
560
561             // ************** motor 1 ********************
```

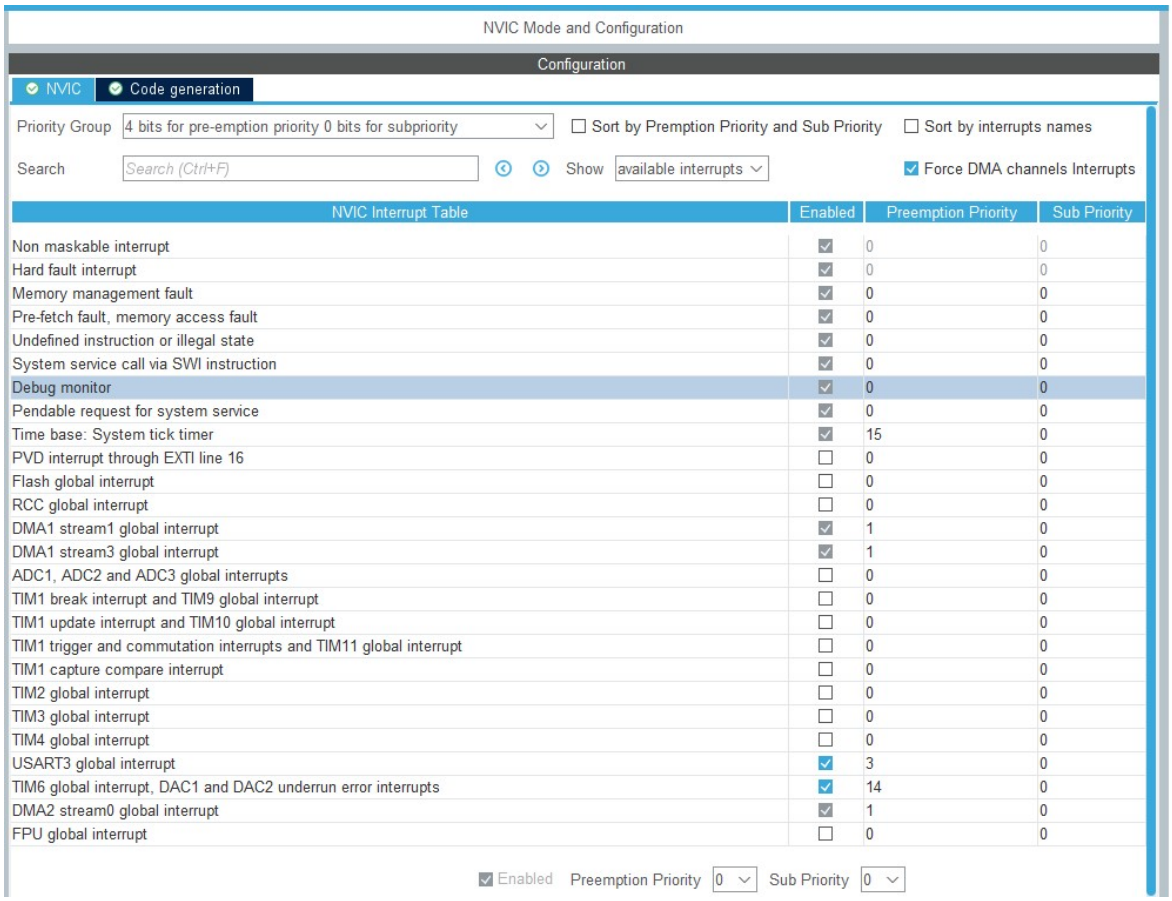
```
562
563             Cinematica_DeltaR_U.PosM1 = pos_gradM1;//pos_radM1; // posicion de entrada
                de motor 1 en grados
564             Cinematica_DeltaR_U.Pos_ref_x = ptr[1]; // posicion en x del EE
565
566             // ************** motor 2 ********************
567
568             Cinematica_DeltaR_U.PosM2 = pos_gradM2;//pos_radM2; // posicion de entrada
                de motor 2 en grados
569             Cinematica_DeltaR_U.Pos_ref_y = ptr[2]; // posicion en y del EE
570
571             // ************** motor 3 ********************
572
573             Cinematica_DeltaR_U.PosM3 = pos_gradM3;//pos_radM3; // posicion de entrada
                de motor 3 en grados
574             Cinematica_DeltaR_U.Pos_ref_z = ptr[3]; // posicion en z del EE
575
576             // ******************funcion step de simulink************************
577             Cinematica_DeltaR_step();
578
579             // ********SALIDAS*********
580
581             PosJoint_1 = Cinematica_DeltaR_Y.Joint_pos[0]; //posicion de capstan 1
                (articulacion 1) en grados - a partir de la posicion del motor
582             PosJoint_2 = Cinematica_DeltaR_Y.Joint_pos[1]; //posicion de capstan 2
                (articulacion 2) en grados - a partir de la posicion del motor
583             PosJoint_3 = Cinematica_DeltaR_Y.Joint_pos[2]; //posicion de capstan 3
                (articulacion 3) en grados - a partir de la posicion del motor
584
585             PosEE_x = Cinematica_DeltaR_Y.EE_pos[0]; //posicion de EE x - a partir de la
                posicion del motor
586             PosEE_y = Cinematica_DeltaR_Y.EE_pos[1]; //posicion de EE y - a partir de la
                posicion del motor
587             PosEE_z = Cinematica_DeltaR_Y.EE_pos[2]; //posicion de EE z - a partir de la
                posicion del motor
588
589             // ************** salidas motor 1 ********************
590
591             duty1 = fabs(Cinematica_DeltaR_Y.Current[0]);
592             pwm1 = map(duty1, 0, 21, 120, 3599);
593             //pwm1 = map(duty1, 0, 18, 75, 3599);
594
595             if(pwm1 > 3599)
596             {
597                 pwm1 = 3599;
598             }
599
600             // ************** salidas motor 2 ********************
601
602             duty2 = fabs(Cinematica_DeltaR_Y.Current[1]);
603             pwm2 = map(duty2, 0, 25, 120, 3599);
604             //pwm2 = map(duty2, 0, 17, 75, 3599);
605
606             if(pwm2 > 3599)
607             {
608                 pwm2 = 3599;
609             }
610
611             // ************** salidas motor 3 ********************
612
613             duty3 = fabs(Cinematica_DeltaR_Y.Current[2]);
614             pwm3 = map(duty3, 0, 21, 120, 3599);
615             //pwm3 = map(duty3, 0, 20, 75, 3599);
616
617             if(pwm3 > 3599)
618             {
619                 pwm3 = 3599;
620             }
621
```

```
622                 __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1,pwm1);
623                 __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2,pwm2);
624                 __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3,pwm3);
625
626                 HAL_GPIO_WritePin(Output_M1_GPIO_Port, Output_M1_Pin,Cinematica_DeltaR_Y.Giro
                    [0]); // giro para motor M1
627                 HAL_GPIO_WritePin(Output_M2_GPIO_Port, Output_M2_Pin,Cinematica_DeltaR_Y.Giro
                    [1]); // giro para motor M2
628                 HAL_GPIO_WritePin(Output_M3_GPIO_Port, Output_M3_Pin,Cinematica_DeltaR_Y.Giro
                    [2]); // giro para motor M3
629             }
630
631             if (estado > 3)
632             {
633                 estado = 0;
634             }
635
636             if (cnt1 == 200) // 200 ms = 5Hz
637             {
638                 HAL_GPIO_TogglePin(LED_AZUL_GPIO_Port, LED_AZUL_Pin);
639                 cnt1 = 0;
640             }
641
642             if (cnt2 == 10) // 10 ms = 100 Hz
643             {
644
645                 cnt2 = 0;
646             }
647
648             if (cnt3 == 8) // 8 ms = 125Hz
649             {
650
                    //sprintf(TEXT,"E%1u,EE.des(x,y,z):%5.2f,%5.2f,%5.2f,EE(x,y,z):%9.6f,%9.6f,%9
                    .6f,C1,C2,C3:%6.2f,%6.2f,%6.2f,SigCurrent:%8.2f,%8.2f,%8.2f\r\n",estado,Cinem
                    atica_DeltaR_U.Pos_ref_x,Cinematica_DeltaR_U.Pos_ref_y,Cinematica_DeltaR_U.Po
                    s_ref_z,PosEE_x,PosEE_y,PosEE_z,PosJoint_1,PosJoint_2,PosJoint_3,Cinematica_D
                    eltaR_Y.Current[0],Cinematica_DeltaR_Y.Current[1],Cinematica_DeltaR_Y.Current
                    [2]);
651                 sprintf(TEXT+7,"%1u, %9.6f, %9.6f, %9.6f, %6.2f, %6.2f, %6.2f, %7.2f, %7.2f,
                    %7.2f\r\n",estado,PosEE_x,PosEE_y,PosEE_z,PosJoint_1,PosJoint_2,PosJoint_3,
                    pos_gradM1,pos_gradM2,pos_gradM3);
652                 HAL_UART_Transmit_DMA(&huart3, (uint8_t*)TEXT, strlen(TEXT));
653
654                 cnt3 = 0;
655             }
656         }
657     }
658
659     /* USER CODE END PFP */
660
661     /* Private user code ---------------------------------------------------------*/
662     /* USER CODE BEGIN 0 */
663
664     /* USER CODE END 0 */
665
666     /**
667       * @brief  The application entry point.
668       * @retval int
669       */
670     int main(void)
671     {
672       /* USER CODE BEGIN 1 */
673
674       /* USER CODE END 1 */
675
676       /* MCU Configuration--------------------------------------------------------*/
677
678       /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
679       HAL_Init();
```

```
680
681     /* USER CODE BEGIN Init */
682
683     /* USER CODE END Init */
684
685     /* Configure the system clock */
686     SystemClock_Config();
687
688     /* USER CODE BEGIN SysInit */
689
690     /* USER CODE END SysInit */
691
692     /* Initialize all configured peripherals */
693     MX_GPIO_Init();
694     MX_DMA_Init();
695     MX_USART3_UART_Init();
696     MX_ADC1_Init();
697     MX_TIM1_Init();
698     MX_TIM2_Init();
699     MX_TIM6_Init();
700     MX_TIM4_Init();
701     MX_TIM3_Init();
702     /* USER CODE BEGIN 2 */
703
704     HAL_UART_Receive_IT(&huart3, Rx_data, 1); // inicializa la interrupcion en el terminal
        serial
705     HAL_ADC_Start_DMA(&hadc1, (uint32_t *)ADC1_buffer, length);
706     sprintf(TEXT, "UAEM RS");
707
708     //It starts to 0 the ptr vector (buffer), only one time
709     for(int i; i<=cadena_length; i++)
710     {
711       ptr[i] = 0.0;
712     }
713
714     // inicializa los timers en modo encoder con ambos canales, A y B
715       HAL_TIM_Encoder_Start(&htim2, TIM_CHANNEL_ALL); // el TIM2 de 32 bits
716       HAL_TIM_Encoder_Start(&htim3, TIM_CHANNEL_ALL); // el TIM3 es de 16 bits
717       HAL_TIM_Encoder_Start(&htim4, TIM_CHANNEL_ALL); // el TIM4 es de 16 bits
718
719       HAL_TIM_Base_Start_IT(&htim6);
720
721       HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1); // inicializa timmer en modo pwm canal 1
          para M1
722       HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2); // inicializa timmer en modo pwm canal 2
          para M2
723       HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3); // inicializa timmer en modo pwm canal 3
          para M3
724
725       __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1,pwm1);
726       __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2,pwm2);
727       __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3,pwm3);
728
729       //inicializa para bloques simulink
730       Control_posicion_gral_initialize();
731       Derivada_vel_initialize();
732       Cinematica_DeltaR_initialize();
733       val_DeltaEE_initialize();
734
735     /* USER CODE END 2 */
736
737     /* Infinite loop */
738     /* USER CODE BEGIN WHILE */
739     while (1)
740     {
741         if(Transfer_cplt) // si se cumple la bandera al teclear ENTER(13 en ascii)
742         {
743             //parseline(); //se ejecuta la funcion parseline
744
```

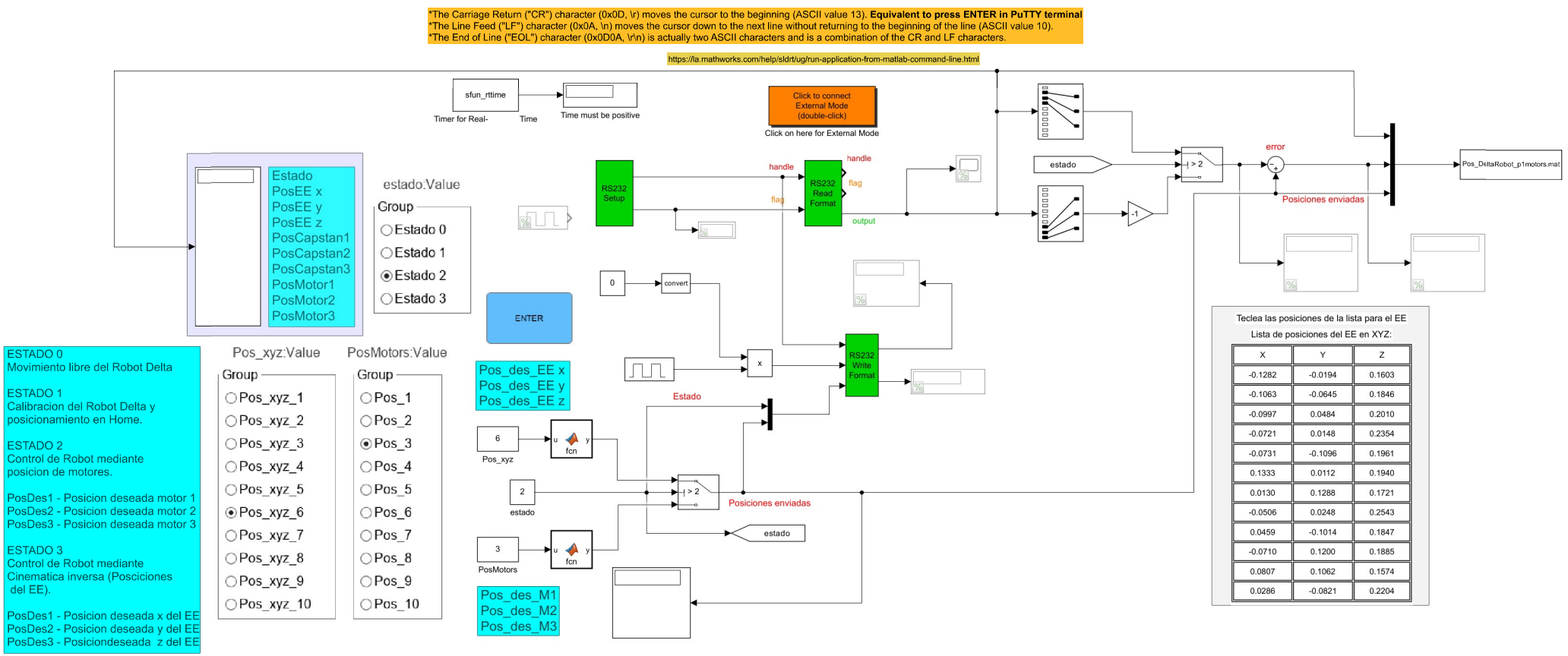
```
745             ptr = split(cadena_length);//ptr contains the string separated by comma
746 
747             //HAL_Delay(100);
748             Transfer_cplt=0; //reinicio la bandera a cero
749         }
750         //Btn_user = HAL_GPIO_ReadPin(BTN_AZUL_GPIO_Port, BTN_AZUL_Pin);
751         //HAL_Delay(100);
752 
753       /* USER CODE END WHILE */
754 
755       /* USER CODE BEGIN 3 */
756     }
757     /* USER CODE END 3 */
758   }
759 
760   /**
761     * @brief System Clock Configuration
762     * @retval None
763     */
764   void SystemClock_Config(void)
765   {
766     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
767     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
768 
769     /** Configure LSE Drive Capability
770     */
771     HAL_PWR_EnableBkUpAccess();
772 
773     /** Configure the main internal regulator output voltage
774     */
775     __HAL_RCC_PWR_CLK_ENABLE();
776     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
777 
778     /** Initializes the RCC Oscillators according to the specified parameters
779     * in the RCC_OscInitTypeDef structure.
780     */
781     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
782     RCC_OscInitStruct.HSEState = RCC_HSE_BYPASS;
783     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
784     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
785     RCC_OscInitStruct.PLL.PLLM = 4;
786     RCC_OscInitStruct.PLL.PLLN = 216;
787     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
788     RCC_OscInitStruct.PLL.PLLQ = 2;
789     RCC_OscInitStruct.PLL.PLLR = 2;
790     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
791     {
792       Error_Handler();
793     }
794 
795     /** Activate the Over-Drive mode
796     */
797     if (HAL_PWREx_EnableOverDrive() != HAL_OK)
798     {
799       Error_Handler();
800     }
801 
802     /** Initializes the CPU, AHB and APB buses clocks
803     */
804     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
805                                   |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
806     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
807     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
808     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
809     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
810 
811     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_7) != HAL_OK)
812     {
813       Error_Handler();
```

```
814     }
815   }
816
817   /* USER CODE BEGIN 4 */
818
819   /* USER CODE END 4 */
820
821   /**
822     * @brief  This function is executed in case of error occurrence.
823     * @retval None
824     */
825   void Error_Handler(void)
826   {
827     /* USER CODE BEGIN Error_Handler_Debug */
828     /* User can add his own implementation to report the HAL error return state */
829     __disable_irq();
830     while (1)
831     {
832     }
833     /* USER CODE END Error_Handler_Debug */
834   }
835
836   #ifdef  USE_FULL_ASSERT
837   /**
838     * @brief  Reports the name of the source file and the source line number
839     *         where the assert_param error has occurred.
840     * @param  file: pointer to the source file name
841     * @param  line: assert_param error line source number
842     * @retval None
843     */
844   void assert_failed(uint8_t *file, uint32_t line)
845   {
846     /* USER CODE BEGIN 6 */
847     /* User can add his own implementation to report the file name and line number,
848        ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
849     /* USER CODE END 6 */
850   }
851   #endif /* USE_FULL_ASSERT */
852
```


NVIC Mode and Configuration
Configuration
NVIC
Code generation
Priority Group
4 bits for pre-emption priority 0 bits for subpriority
Sort by Premption Priority and Sub Priority
Sort by interrupts names
Search
Search (Ctrl+F)
Show
available interrupts
Force DMA channels Interrupts
NVIC Interrupt Table
Enabled
Preemption Priority
Sub Priority
Non maskable interrupt 0 0
Hard fault interrupt 0 0
Memory management fault 0 0
Pre-fetch fault, memory access fault 0 0
Undefined instruction or illegal state 0 0
System service call via SWI instruction 0 0
Debug monitor 0 0
Pendable request for system service 0 0
Time base: System tick timer 15 0
PVD interrupt through EXTI line 16 0 0
Flash global interrupt 0 0
RCC global interrupt 0 0
DMA1 stream1 global interrupt 1 0
DMA1 stream3 global interrupt 1 0
ADC1, ADC2 and ADC3 global interrupts 0 0
TIM1 break interrupt and TIM9 global interrupt 0 0
TIM1 update interrupt and TIM10 global interrupt 0 0
TIM1 trigger and commutation interrupts and TIM11 global interrupt 0 0
TIM1 capture compare interrupt 0 0
TIM2 global interrupt 0 0
TIM3 global interrupt 0 0
TIM4 global interrupt 0 0
USART3 global interrupt 3 0
TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts 14 0
DMA2 stream0 global interrupt 1 0
FPU global interrupt 0 0
Enabled
Preemption Priority
0
Sub Priority
0

Apéndice G

Modelos Simulink®.


*The Carriage Return ("CR") character (0x0D, \r) moves the cursor to the beginning (ASCII value 13). **Equivalent to press ENTER in PuTTY terminal**
*The Line Feed ("LF") character (0x0A, \n) moves the cursor down to the next line without returning to the beginning of the line (ASCII value 10).
*The End of Line ("EOL") character (0x0D0A, \r\n) is actually two ASCII characters and is a combination of the CR and LF characters.
https://la.mathworks.com/help/sldrt/ug/run-application-from-matlab-command-line.html
sfun_rttime
Timer for Real-
Time
Time must be positive
Click to connect External Mode (double-click)
Click on here for External Mode
Estado
PosEE x
PosEE y
PosEE z
PosCapstan1
PosCapstan2
PosCapstan3
PosMotor1
PosMotor2
PosMotor3
estado:Value
Group
Estado 0
Estado 1
Estado 2
Estado 3
RS232 Setup
RS232 Read Format
handle
flag
output
estado
error
Posiciones enviadas
Pos_DeltaRobot_p1motors.mat
ENTER
0
convert
x
RS232 Write Format
Pos_des_EE x
Pos_des_EE y
Pos_des_EE z
Estado
6
Pos_xyz
fcn
2
estado
3
PosMotors
Pos_des_M1
Pos_des_M2
Pos_des_M3
ESTADO 0
Movimiento libre del Robot Delta
ESTADO 1
Calibracion del Robot Delta y posicionamiento en Home.
ESTADO 2
Control de Robot mediante posicion de motores.
PosDes1 - Posicion deseada motor 1
PosDes2 - Posicion deseada motor 2
PosDes3 - Posicion deseada motor 3
ESTADO 3
Control de Robot mediante Cinematica inversa (Posciciones del EE).
PosDes1 - Posicion deseada x del EE
PosDes2 - Posicion deseada y del EE
PosDes3 - Posiciondeseada z del EE
Pos_xyz:Value
Group
Pos_xyz_1
Pos_xyz_2
Pos_xyz_3
Pos_xyz_4
Pos_xyz_5
Pos_xyz_6
Pos_xyz_7
Pos_xyz_8
Pos_xyz_9
Pos_xyz_10
PosMotors:Value
Group
Pos_1
Pos_2
Pos_3
Pos_4
Pos_5
Pos_6
Pos_7
Pos_8
Pos_9
Pos_10
Teclea las posiciones de la lista para el EE
Lista de posiciones del EE en XYZ:
X Y Z
-0.1282 -0.0194 0.1603
-0.1063 -0.0645 0.1846
-0.0997 0.0484 0.2010
-0.0721 0.0148 0.2354
-0.0731 -0.1096 0.1961
0.1333 0.0112 0.1940
0.0130 0.1288 0.1721
-0.0506 0.0248 0.2543
0.0459 -0.1014 0.1847
-0.0710 0.1200 0.1885
0.0807 0.1062 0.1574
0.0286 -0.0821 0.2204

Fecha de inicio	**Fecha de terminación**
Agosto de 2021	Agosto de 2023

Firma del alumno: _____

Ing. Raúl Ismael Serna Ruiz

Director (es) del tema de tesis

Dr. Juan Manuel Jacinto Villegas _____

Codirector de tesis

Dr. Juan Carlos Ávila Vilchis _____

Tutor Adjunto

Dr. Otniel Portillo Rodríguez _____