



Universidad Nacional de San Luis
Facultad de Ingeniería y Ciencias Agropecuarias

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE DRON.

Tustanoski, Juan Manuel y Ratto Iglesias, Gabriel Andres

Trabajo final de Ingeniería Mecatrónica

Director: Dr. Ing. Eduardo Maximiliano, Asensio.

Codirector: Mg. Ing. Guillermo Luciano, Magaldi.

Villa Mercedes, San Luis

2025

DERECHO DE AUTOR

© 2025, Tustanoski Juan Manuel y Ratto Iglesias Gabriel Andres.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

AGRADECIMIENTOS

A nuestros padres y madres, por el enorme esfuerzo que realizaron a lo largo de todos estos años, haciéndonos posible estudiar esta carrera, aun estando lejos de casa y teniendo que mudarnos a otra ciudad. Gracias por brindarnos la tranquilidad y el apoyo necesario para poder dedicarnos plenamente al estudio, por acompañarnos en los momentos difíciles y alentarnos a seguir adelante. Agradecemos los valores, la educación y la formación que nos transmitieron desde siempre, los cuales fueron fundamentales para convertirnos en las personas y profesionales que aspiramos a ser.

A Belén Ante, de parte de Juan. Gracias por haberme brindado su gran acompañamiento en los momentos difíciles que atravesé durante el desarrollo del proyecto.

Al Ing. Carlos Conte, por brindarnos un dron cuadricóptero para desarmar, hacer pruebas y extraer componentes para la primer versión prototipo del dron que se desarrolló para este trabajo final.

A nuestros compañeros de la facultad por compartir el día a día y los vínculos generados.

A los docentes de la facultad, por su dedicación y trabajo en nuestra formación.

A Dr. Ing. Eduardo Maximiliano, Asensio. y Mg. Ing. Guillermo Luciano, Magaldi. Director y Codirector de este Trabajo Final por su guía y colaboración en este proyecto.

Al Ing. Carlos Catuogno por su asesoramiento sobre los componentes del cuadricóptero.

Finalmente, a la Universidad Nacional de San Luis, por brindarnos un entorno de calidad para desarrollarnos como profesionales.

RESUMEN

En este informe se presenta el desarrollo de un prototipo funcional de cuadricóptero en configuración "X", el cual tiene como objetivo poder realizar diversas modificaciones, con fines didácticos. Esto permite la posibilidad de probar diversos controladores, agregar o cambiar los sensores, utilizar configuraciones diversas para observar estados, así como también realizar cambios físicos al chasis y los actuadores.

Este prototipo fue diseñado en un software CAD 3D y sus piezas fueron fabricadas por medio de la impresión 3D. Además, se realizó un modelo matemático del mismo y se implementó en MATLAB para hallar un controlador LQR deseado. Así mismo, se incorporó en Simulink para realizar simulaciones con pruebas del modelo y el controlador.

Finalmente se implementó el controlador en el kit de desarrollo ESP32, el cual toma los datos del sensor MPU6050 y, por medio del algoritmo de Madgwick, se lo procesó con el fin de mejorar la estimación de los estados. Se utilizaron motores brushless con controladores ESC para recibir las acciones de control del controlador LQR.

Palabras claves — Brushless, Cuadricóptero, ESP32, Impresión 3D, LQR, Mpu6050.

ÍNDICE DE CONTENIDO

CONTENIDO

1.1	Objetivos	13
1.1.1	Objetivo general	13
1.1.2	Objetivos específicos.....	13
1.2	Alcances y limitaciones.....	13
1.3	Metodología.....	14
1.4	Marco teórico.....	19
1.4.1	Controlador LQR	19
1.4.2	Motores brushless DC	19
1.4.3	Modulo IMU	20
1.4.4	Controlador electrónico de velocidad (ESC)	20
1.4.5	Impresión 3D FDM	20
1.4.6	Conceptos básicos sobre el movimiento de un dron.....	21
1.4.7	Grados de libertad de un dron	22
1.4.8	Tensor de inercia del cuadricóptero.....	23
1.4.9	Matrices de rotación	23
1.4.10	Velocidades angulares del cuadricóptero	24
1.4.11	Clasificaciones de drones según su masa máxima de despegue	24
CAPITULO 2: Modelado CAD 3D.....		26
CAPITULO 3: Modelado matemático.....		33
3.1	Dinámicas de los actuadores:.....	38
3.2	Modelado por variable de estado no lineal.....	39
3.3	Linealización:.....	41
CAPITULO 4: Diseño del controlador y simulación.....		44

4.2 Controlabilidad y Observabilidad:	45
4.3 Controlador LQR:	47
CAPITULO 5: Implementación	62
5.1 Hardware	62
5.2 Medición del empuje generado por los motores.....	66
5.3 Diagrama de conexiones eléctricas	68
5.4 Mando remoto	70
5.5 Programación del microcontrolador	71
5.6 Pruebas de vuelo.....	77
5.7 Análisis de costos y cronograma de ejecución.....	80
CAPITULO 6: Conclusión y trabajo a futuro	82
Glosario.....	83
Referencias Bibliográficas	84

ÍNDICE DE FIGURAS

FIGURAS

Figura N° 1: <i>Configuraciones más comunes de los drones de ala rotatoria</i>	10
Figura N° 2: <i>Ejemplo de un dron que utiliza Ardupilot, Iris Quadcopter, cortesía de [1]</i>	11
Figura N° 3: <i>Captura del proyecto de pratikPhadte, cortesía de [3]</i>	11
Figura N° 4: <i>Captura del proyecto del usuario Max Imagination, cortesía de [4]</i>	12
Figura N° 5: <i>Imagen del dron DJI AGRAS T100, cortesía de [5]</i>	12
Figura N° 6: <i>Esquema de motor BLDC, cortesía de [6]</i>	19
Figura N° 7: <i>Esquema de impresión FDM, cortesía de [7]</i>	20
Figura N° 8: <i>Conceptos de movimiento del cuadricóptero</i>	22

Figura N° 9: <i>Modelo simplificado de un motor.</i>	27
Figura N° 10: <i>Uno de los brazos del dron con el ESC.</i>	28
Figura N° 11: <i>Chasis con brazos acoplados.</i>	28
Figura N° 12: <i>Batería montada sobre el chasis con las bridas.</i>	29
Figura N° 13: <i>Parte inferior del chasis.</i>	29
Figura N° 14: <i>Parte inferior del chasis segunda vista.</i>	30
Figura N° 15: <i>Tren de aterrizaje.</i>	30
Figura N° 16: <i>Vista completa del dron.</i>	31
Figura N° 17: <i>Vista superior del dron.</i>	32
Figura N° 18: <i>Vista lateral del dron.</i>	32
Figura N° 19: <i>Ejes cartesianos y ángulos de Euler.</i>	34
Figura N° 20: <i>Vista superior del dron.</i>	38
Figura N° 21: <i>Diagrama de bloques del funcionamiento del dron.</i>	45
Figura N° 22: <i>Resultados de Controlabilidad y Observabilidad</i>	47
Figura N° 23: <i>Diagrama de bloques en Simulink.</i>	49
Figura N° 24: <i>Subsistema que contiene a la planta.</i>	50
Figura N° 25: <i>Referencias del mando remoto.</i>	52
Figura N° 26: <i>Grafico de los estados Theta, P, Y y V.</i>	53
Figura N° 27: <i>Grafico de los estados Phi, Q, X y U.</i>	54
Figura N° 28: <i>Grafico de los estados Rho, R, Z y W.</i>	55
Figura N° 29: <i>Señales de control.</i>	57
Figura N° 30: <i>Resultados del input "ROLL"</i>	58
Figura N° 31: <i>Resultados del input "PITCH"</i>	59
Figura N° 32: <i>Resultados del input "Rho_Ref"</i>	60
Figura N° 33: <i>Resultados del input "Ft_Ref"</i>	61
Figura N° 34: <i>Tabla técnica del motor B2212. Cortesía de [15].</i>	63

Figura N° 35: <i>Placa de desarrollo Esp32</i>	63
Figura N° 36: <i>Sensor MPU6050</i>	64
Figura N° 37: <i>ESC del Dron</i>	64
Figura N° 38: <i>Dron ensamblado sin la batería</i>	65
Figura N° 39: <i>Dron completo</i>	65
Figura N° 40: <i>Dron con protectores de hélices</i>	66
Figura N° 41: <i>CAD 3D del banco de motores</i>	66
Figura N° 42: <i>Banco de prueba de motores</i>	67
Figura N° 43: <i>Interpolación cuadrática</i>	68
Figura N° 44: <i>Diagrama de conexiones eléctricas</i>	69
Figura N° 45: <i>Mando remoto</i>	70
Figura N° 46: <i>Diagrama de conexiones eléctricas del mando remoto</i>	71
Figura N° 47: <i>Diagrama de flujo del funcionamiento del dron</i>	76
Figura N° 48: <i>Dron listo para despegar</i>	77
Figura N° 49: <i>Dron en vuelo</i>	78
Figura N° 50: <i>Dron descendiendo</i>	79
Figura N° 51: <i>Foto con el Dron</i>	79
Figura N° 52: <i>Diagrama de Gantt del proyecto</i>	81

ÍNDICE DE TABLAS

TABLAS

Tabla N° 1: <i>Valores de fuerza de empuje vs PWM</i>	67
Tabla N° 2: <i>Análisis de costo del prototipo</i>	80

CAPITULO 1: PROPUESTA

Los drones son vehículos aéreos no tripulados, comúnmente llamados UAV por su abreviación en inglés. Estos son usados en una diversa gama de aplicaciones incluyendo, pero no limitándose a:

- Fotografía aérea
- Topografía
- Inspección de infraestructura
- Monitoreo de cultivos
- Plantación de semillas / fumigación
- Vigilancia
- Envíos

Los drones se pueden clasificar en dos categorías principales, de ala fija, cuya estructura es similar a la de un avión convencional y de ala rotatoria, que utilizan uno o varios rotores para despegar o moverse verticalmente. El dron realizado en este proyecto se encuentra en esta segunda categoría.

Dentro de los drones de ala rotatoria se pueden subdividir por su número de rotores y su disposición. A continuación, se presenta una imagen exponiendo las disposiciones más comunes:

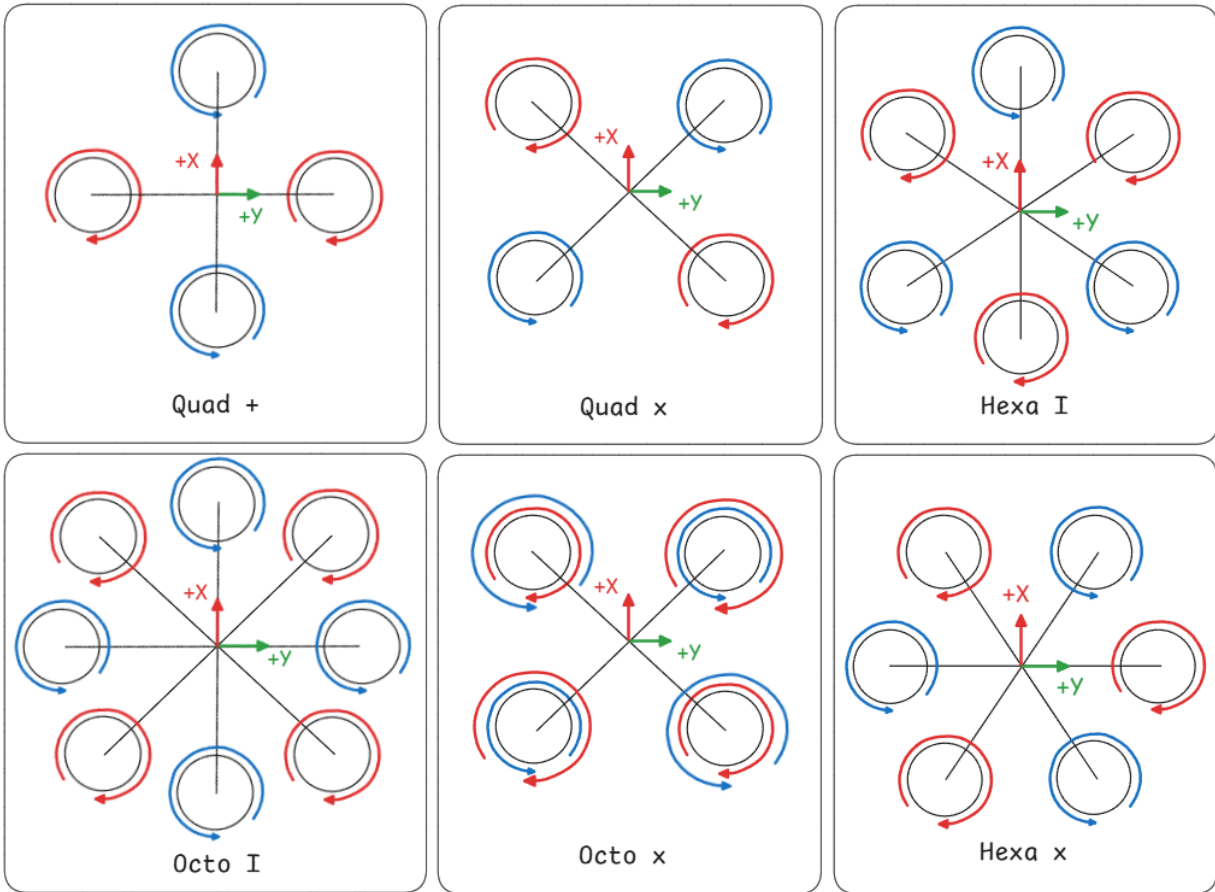


Figura N° 1: Configuraciones más comunes de los drones de ala rotatoria.

En concreto, un cuadricóptero, es un dron impulsado por cuatro motores, generalmente colocados en configuración “cruz” o “más”, con dos motores girando en sentido horario y dos en sentido antihorario para contrarrestar los torques generados por estos.

El cuadricóptero es un sistema subactuado que puede moverse en seis grados de libertad (tres de rotación y tres de traslación). Si bien este puede moverse en el eje Z vertical sin alterar las otras orientaciones y traslaciones, no puede trasladarse en los ejes horizontales sin cambiar su actitud (La actitud es un conjunto de ángulos que describen la orientación del dron, normalmente Yaw Pitch y Roll), es por ello por lo que se aplican sistemas de control que mantienen el cuadricóptero estable en torno a los valores deseados y máximos que puede lograr sin perder la estabilidad.

En la actualidad existen una gran cantidad de drones comerciales, como, por ejemplo, los de la marca DJI. Así como también varias plataformas de desarrollo de software libre como Ardupilot, un sistema de código abierto que soporta distintos tipos de drones y provee una plataforma sólida para control y supervisión de vuelo utilizando múltiples controles PID superpuestos para el control de altitud, navegación, control posicional, etc. que permiten la creación de modelos personalizados con relativa facilidad.



Figura N° 2: *Ejemplo de un dron que utiliza Ardupilot, Iris Quadcopter, cortesía de [1].*

Otro proyecto de desarrollo es el del usuario de Github pratikPhadte [2], donde desarrolla un dron cuadricóptero con el microcontrolador ESP32. A lo largo del desarrollo fue creando videos donde contaba su aprendizaje y fue interactuando con la comunidad de YouTube, recibiendo consejos y probando nuevas opciones.



Figura N° 3: *Captura del proyecto de pratikPhadte, cortesía de [3]*

Otro proyecto desarrollado por un usuario de la comunidad de YouTube es ESP-FLY del usuario Max Imagination [4] el cual se muestra en la figura 4.

El proyecto presenta un mini dron cuadricóptero que tan solo pesa menos de treinta gramos y permite incorporar una cámara. También utiliza un microcontrolador ESP32.



Figura N° 4: *Captura del proyecto del usuario Max Imagination, cortesía de [4]*

En cuanto a drones comerciales un ejemplo es el DJI AGRAS T100, un dron diseñado para la agricultura, capaz de realizar tareas de izaje, riego y dispersión de semillas.

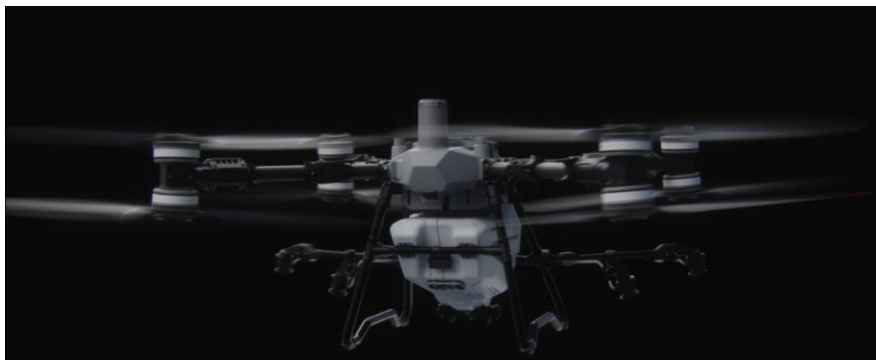


Figura N° 5: *Imagen del dron DJI AGRAS T100, cortesía de [5]*

En este proyecto se diseñará un prototipo de dron y su controlador utilizando un regulador cuadrático lineal (LQR) diseñado en MATLAB a partir del modelado matemático del cuadricóptero.

1.1 Objetivos

1.1.1 Objetivo general

- Diseñar e implementar un prototipo funcional de un cuadricóptero con partes modulares.

1.1.2 Objetivos específicos

- Realizar un diseño de chasis del prototipo con el objetivo de minimizar costos.
- Obtener el modelo matemático del dron.
- Diseñar un controlador y realizar simulaciones.
- Diseñar el circuito electrónico necesario para el control de vuelo, los sensores y motores.
- Diseño del mando remoto.
- Implementar el prototipo de forma física.
- Realizar pruebas de funcionamiento.

1.2 Alcances y limitaciones

Se desarrollará un prototipo funcional capaz de elevarse, mantenerse en vuelo y aterrizar. Para ello se diseñará un chasis que sea capaz de ser impreso en 3D, con los montajes modulares para los motores y los demás componentes que conforman al dron. Se diagramará un circuito electrónico para controlar los motores y se programará un sistema de control de vuelo para el dron. El objetivo de este controlador es que sea sencillo y económico, permitiendo prescindir del uso de cámaras y sistemas de posicionamiento por coordenadas. Si bien el prototipo no será capaz de navegar autónomamente, permitirá realizar diferentes pruebas de circuitos y de estrategias de control de vuelo.

1.3 Metodología

Con la finalidad de alcanzar los objetivos planteados en este trabajo se propone la siguiente metodología:

- Realizar un diseño de chasis del prototipo con el objetivo de minimizar costos.

- Investigar tipos de drones y las partes que los componen.

Se realizará una investigación sobre el funcionamiento de los drones, incluyendo los conceptos físicos a tener en cuenta, las variables que se deben controlar para lograr un prototipo funcional

- Seleccionar el tipo de dron para el prototipo.

Se ponderarán las distintas configuraciones posibles y se seleccionara la más viable

- Diseñar un boceto del chasis.

Se procederá a realizar un primer boceto del dron con el objetivo de ir editando el mismo a lo largo de las etapas de diseño según se requiera.

- Definir partes modulares y no modulares.

En esta actividad se definirán que piezas del dron serán modulares y que permitirán al ser intercambiables. Para ello se realizarán bocetos en 3D y se diseñarán los mismos en el software de diseño. A su vez, se seleccionarán las partes que no serán intercambiables

- Definir acoplamiento para los módulos.

Se harán pruebas de diversas uniones para seleccionar la o las formas de acoplar los módulos.

- Diseñar los módulos para las piezas a utilizar.

Una vez realizadas las actividades anteriores se podrá dar por terminada la definición de los módulos.

- Obtener el modelo matemático del dron.
 - Esquematizar el dron en plantas y hallar sus modelos matemáticos.

Se hallarán las ecuaciones que describen el comportamiento físico del dron para luego diseñar una estrategia de control.

- Hallar las funciones de transferencia de las plantas

Se calcularán las funciones de transferencias que representarán el comportamiento físico del dron.

- Diseñar un controlador y realizar simulaciones.

- Definir el esquema de control a utilizar.

En esta actividad se seleccionará la estrategia de control a abordar para lograr el control de vuelo del dron.

- Diseñar un controlador

Una vez seleccionada la estrategia de control, se comenzará con el diseño de los controladores para las diversas funciones de transferencias que modelan los comportamientos del dron.

- Realizar simulaciones.

En esta etapa se simularán los controladores y la planta en SIMULINK para evaluar el funcionamiento de la misma.

- Diseñar el circuito electrónico necesario para el control de vuelo, los sensores y motores.

- Seleccionar los motores.

Se harán diversas búsquedas para hallar las opciones disponibles y se realizara una selección en base a un análisis de conveniencia según costo-beneficio.

- Definir sensores y seleccionarlos

Con el controlador ya diseñado, se determinarán los sensores necesarios para su funcionamiento y se evaluarán los diferentes modelos disponibles de estos.

- Diagramar circuito electrónico en software

Mediante el uso del software se harán esquemas de circuito con la finalidad de definir los componentes necesarios.

- Seleccionar las baterías que alimentaran al dispositivo.

Se elegirá una batería en base a su peso, precio, disponibilidad y capacidad.

- Diseño del mando remoto o Investigar las partes necesarias.

Se buscará información y se decidirá cuáles son las partes necesarias para un mando funcional.

- Seleccionar componentes.

Se decidirá que componentes utilizar para que el mando sea efectivo y de bajo costo.

- Diseñar circuito electrónico.

Se diagramará en KiCAD el circuito necesario para que los sticks y botones funcionen y se pueda transmitir la información que generan al microcontrolador.

- Seleccionar forma de comunicación con el dron.

Se investigarán las formas de comunicación disponibles entre los microcontroladores y se determinara cual es la más efectiva.

- Escribir programa para la placa de desarrollo.

Se realizará la programación de los joysticks y el medio de comunicación con el dron para que este reciba las referencias deseadas por el usuario.

- Diseñar carcasa en CAD 3D.

Se utilizará FUSION 360 para diseñar la geometría 3D de la carcasa para su posterior impresión.

- Implementar el prototipo de forma física.

- Construir el circuito electrónico.

Se soldarán los componentes a la placa perforada, comprobando con un multímetro que estén conectados de manera efectiva.

- Realizar conexionado de los componentes.

Se realizarán los cableados entre los sensores, la placa, los motores y el microcontrolador.

- Imprimir el chasis y sus módulos.

Se utilizará una impresora 3D para construir el chasis y los módulos necesarios, realizando ajustes a la impresora donde sea necesario.

- Montar los motores.

Se colocarán los motores en sus acoples correspondientes al chasis.

- Programar y configurar las placas de desarrollo.

Se procederá a introducir el controlador y la configuración de sensores en la placa de desarrollo.

- Realizar pruebas de funcionamiento.

- Verificar funcionamiento de los sensores.

Se utilizará un osciloscopio para corroborar el correcto funcionamiento de los sensores utilizados.

- Comprobar el funcionamiento del control y realizar afinaciones necesarias.

Se evaluará la estabilidad del sistema con el controlador en funcionamiento y se realizarán los ajustes necesarios para que funcione correctamente.

- Medir con osciloscopio las acciones de control.

En esta actividad se busca determinar que las acciones de control ejercidas por el microcontrolador sean las esperadas.

- Realizar prueba de vuelo.

Se comprobarán las características de vuelo del dron realizando diferentes evaluaciones de funcionamiento en el prototipo.

- Redactar el informe del proyecto.

Se escribirán los avances del proyecto a medida que se van realizando los distintos objetivos hasta finalizar el proyecto mediante la revisión y corrección del informe.

1.4 Marco teórico

1.4.1 Controlador LQR

El regulador cuadrático lineal (LQR) es un algoritmo matemático que minimiza una función de costos, generalmente definida por las desviaciones de los parámetros clave definidos por el usuario. Este algoritmo encuentra un controlador de retroalimentación de estados apropiado para minimizar las desviaciones no deseadas.

1.4.2 Motores brushless DC

Los motores sin escobillas o brushless de corriente continua (BLDC) son motores en donde el rotor consiste en un imán permanente, con las bobinas ubicadas en el estator. Como las bobinas no se mueven no se necesita de un conmutador o escobillas.

A diferencia de un motor con escobillas donde la conmutación de las bobinas se produce con la rotación y la alineación de los carbones con los contactos de estas, en un motor brushless la conmutación de las bobinas se hace electrónicamente. Para lograr un buen funcionamiento del motor y controlar la velocidad, se requiere un controlador que ajuste la magnitud y dirección de la corriente en las bobinas. Este proceso complejo se resuelve mediante el uso de controladores electrónicos de velocidad (ESC), los cuales pueden implementar diversas estrategias de control y formas de estimar la rotación.

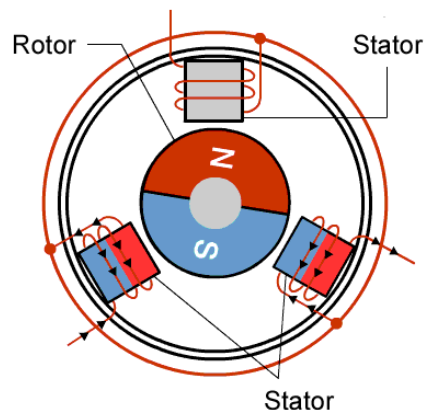


Figura N° 6: Esquema de motor BLDC, cortesía de [6].

1.4.3 Modulo IMU

La unidad de medición inercial (IMU) es un sensor que contiene acelerómetros y giroscopios para obtener información sobre la orientación y velocidad de un objeto, son utilizadas comúnmente para controlar aviones y drones. Comúnmente se utiliza una fusión de estos dos sensores, en conjunto con un barómetro para medir la altura y un módulo de GPS para calcular la posición y actitud del dron.

1.4.4 Controlador electrónico de velocidad (ESC)

Los ESC convierten la tensión DC de la batería en una señal trifásica adecuada para motores BLDC, regulando el voltaje y frecuencia de acuerdo con las órdenes del controlador de vuelo. Para lograr un control preciso de velocidad, muchos ESC utilizan técnicas de control avanzadas como el control vectorial, lo cual mejora la eficiencia y optimiza la respuesta dinámica.

1.4.5 Impresión 3D FDM

La impresión 3D FDM (modelado por deposición fundida) es un proceso de manufactura aditiva por extrusión de material. Una bobina de filamento plástico se alimenta a través de un extrusor que lo calienta, funde y extrude a través de una boquilla para conformar el objeto a fabricar, conforme a las instrucciones dadas por un CNC (control numérico por computadora).

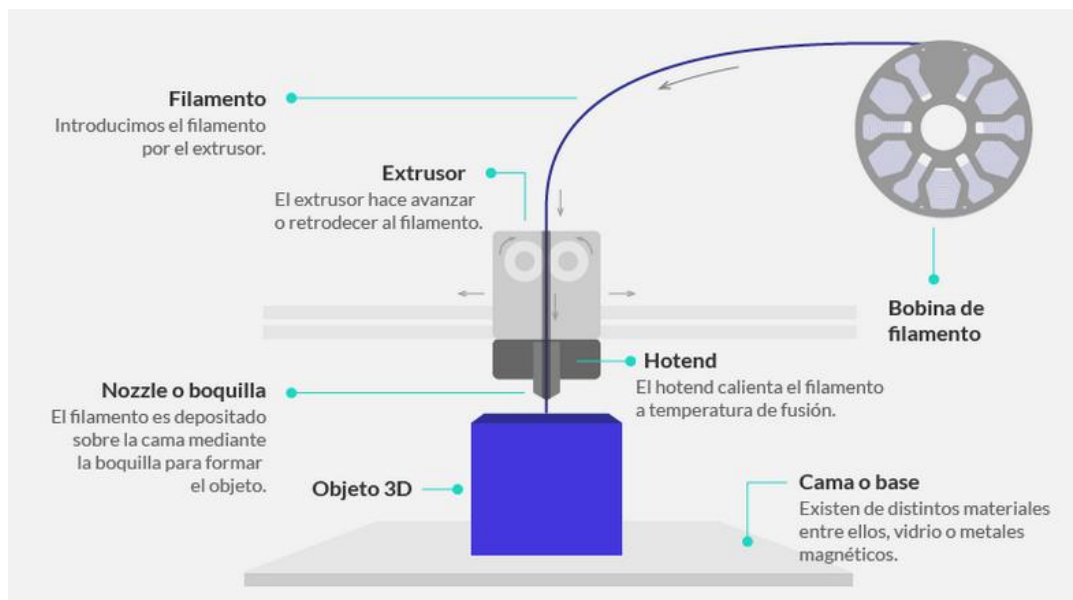


Figura N° 7: Esquema de impresión FDM, cortesía de [7].

1.4.6 Conceptos básicos sobre el movimiento de un dron

A continuación, se explican 4 conceptos fundamentales para entender el estado y movimiento de un dron:

- **Yaw:** Corresponde al giro del dron en el eje vertical Z, permitiendo orientar el frente del dron de izquierda a derecha sin alterar la posición del dron. Para lograr este giro, se aumentan las revoluciones del par de motores que generan el torque en el sentido del Yaw que se quiere realizar y se reducen las revoluciones del otro par de motores. Este torque que generan los motores es producido como reacción a la rotación de la hélice de cada motor y se puede aumentar haciendo que un par de motores opuestos gire más rápido y el otro par más lento.
- **Pitch:** Es la inclinación hacia delante o hacia atrás del dron, en dirección al eje X. Con esto se controla el avance o retroceso de este. Este movimiento se efectúa mediante el aumento de las fuerzas de un par de motores ya sea el trasero o el delantero y la disminución de las fuerzas en el par opuesto.
- **Roll:** Corresponde a la inclinación hacia los lados en el eje Y, que es perpendicular a los ejes X y Z. Este permite los desplazamientos laterales restantes. Esto se logra aumentando las fuerzas en el par izquierdo o derecho y disminuyendo el par contrario.
- **Hover:** Esto no es una rotación como los anteriores, sino la capacidad de mantener una posición estable en el aire, en otras palabras, si el dron se encuentra quieto flotando en el aire se lo considera "en hover".

Los movimientos descritos se presentan en la figura 8.

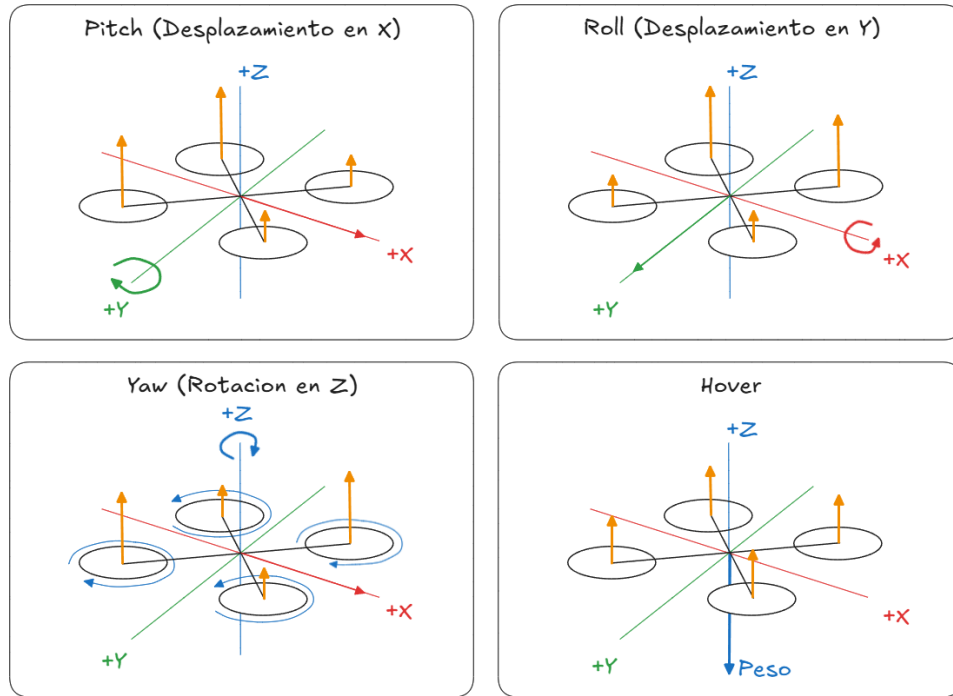


Figura N° 8: *Conceptos de movimiento del cuadricóptero.*

El conjunto de los tres ángulos de rotación (Roll, Pitch y Yaw) se denomina actitud del dron. Esta define su orientación en el espacio con respecto a un sistema de referencia inercial y es una variable fundamental para el control de vuelo, ya que determina la dirección y estabilidad del vehículo.

1.4.7 Grados de libertad de un dron

Grados de libertad (DOF por sus siglas en inglés) hace referencia al número de ejes y sensores combinados para balancear un avión, helicóptero o robot. [12]

En el caso de un cuadricóptero los grados de libertad más comunes son:

- 6 DOF. Con un acelerómetro y un giroscopio de 3 ejes cada uno.
- 9 DOF. Agregando un magnetómetro (brújula) a la configuración anterior.
- 10 DOF. Se agrega un barómetro para realizar una medición de la altura del dron.
- 11 DOF. Se adiciona un módulo GPS.

1.4.8 Tensor de inercia del cuadricóptero

El tensor de inercia es una matriz simétrica que describe cómo se distribuye la masa de un cuerpo alrededor de sus ejes principales de rotación. Determina la resistencia del dron a cambiar su velocidad angular en cada uno de sus ejes.

En un sistema rígido como un cuadricóptero, el tensor de inercia I se expresa como:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Para estructuras simétricas (como un cuadricóptero tipo “+” o “X”), los productos de inercia suelen ser pequeños y se pueden aproximar a cero dando una forma diagonal. [20]

1.4.9 Matrices de rotación

La orientación del dron puede describirse mediante los ángulos de Euler: θ , ϕ y ρ .

Las matrices de rotación permiten transformar un vector desde el sistema cuerpo (body frame) al sistema de referencia inercial (world frame) o viceversa.

Las rotaciones elementales alrededor de los ejes \mathbf{X} , \mathbf{Y} y \mathbf{Z} son:

Rotación Roll (eje \mathbf{X}):

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

Rotación Pitch (eje \mathbf{Y}):

$$R_y(\phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}$$

Rotación Yaw (eje \mathbf{Z}):

$$R_z(\rho) = \begin{bmatrix} \cos \rho & -\sin \rho & 0 \\ \sin \rho & \cos \rho & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación compuesta (orden **Z-Y-X**, típica en drones) es:

$$R = R_z(\rho) R_y(\phi) R_x(\theta)$$

Esta matriz se utiliza para convertir aceleraciones, fuerzas y velocidades entre ambos sistemas de referencia, siendo esencial para el modelado del dron y su control.

1.4.10 Velocidades angulares del cuadricóptero

Los giroscopios de la IMU miden directamente las **velocidades angulares** del dron en los ejes del sistema cuerpo: p , q y r .

Estas velocidades no coinciden directamente con las derivadas de los ángulos de Euler, es por ello que se utiliza la matriz de transformación $T(\theta, \phi)$.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T(\theta, \phi) \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\rho} \end{bmatrix}$$

Donde:

$$T(\theta, \phi) = \begin{vmatrix} 1 & \sin(\theta)\tan(\phi) & \cos(\theta)\tan(\phi) \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \frac{\sin(\theta)}{\cos(\phi)} & \frac{\cos(\theta)}{\cos(\phi)} \end{vmatrix}$$

1.4.11 Clasificaciones de drones según su masa máxima de despegue

Según la EASA (Agencia de la Unión Europea para la Seguridad Aérea) [19] existen 7 clases de drones definidas de acuerdo a su masa máxima de despegue (MTOW):

- Drones construidos de manera privada (por debajo de los 250g)

- C0 (debajo de 250g)
- C1 (debajo de 900g)
- C2 (debajo de 4kg)
- C3 (debajo de 25 kg)
- C4 (debajo de 25 kg, distintas restricciones comparado a C3)
- Drones contruidos de manera privada (por debajo de los 25kg)

La categoría abierta es la referencia para drones de uso comercial o recreativo de bajo riesgo, esta esta subdividida en 3 categorías:

- A1: Permitido sobrevolar gente, pero no multitudes. (Categorías C0, C1 y privados por debajo de 250g)
- A2: Permitido volar cerca de la gente. (Categorías C2)
- A3: Permitido volar lejos de la gente. (Categorías C3, C4 y privados por debajo de 25kg)

CAPITULO 2: MODELADO CAD 3D

En este capítulo se presenta el diseño y modelado tridimensional del dron, desarrollado íntegramente en el software Autodesk Fusion 360.

El objetivo principal de esta etapa fue obtener un diseño propio, económico y de fácil implementación, que pudiera fabricarse mediante impresión 3D, lo cual permite reducir costos y facilitar la replicación del prototipo.

Asimismo, se buscó que el diseño sea modular, de manera que los distintos componentes (motores, sensores, batería, controladores electrónicos) puedan ensamblarse y reemplazarse con facilidad.

El modelado CAD constituye la base física del prototipo y resulta esencial para su posterior análisis matemático y fabricación.

El diseño se realizará a partir de 4 motores 2212 920KV de los cuales ya se dispone (ver capítulo 5.1 para más detalles sobre estos), el fabricante recomienda 2 configuraciones de hélice:

- La primera utiliza una hélice 8045 de 8 pulgadas de diámetro, esta configuración provee 465g de empuje por motor.
- La segunda utiliza una hélice 1045 de 10 pulgadas de diámetro, esta configuración provee 642g de empuje por motor.

Para este prototipo se optó por usar las hélices 1045, si bien al ser más largas el tamaño del dron y el consumo de batería aumentara, el mayor empuje que estas proveen permitirá el uso de una batería de mayor tamaño y más libertad a la hora de realizar el diseño.

Se comenzó tomando las medidas principales de los motores, y se realizó un modelo simplificado de estos para facilitar el diseño de los brazos en los que irán montados.

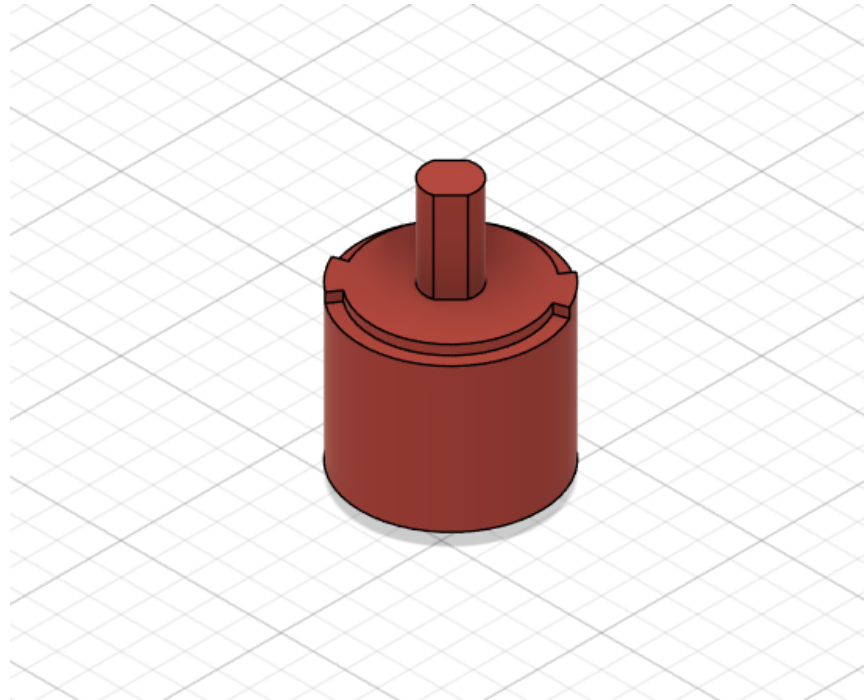


Figura N° 9: *Modelo simplificado de un motor.*

Para el diseño del brazo, se tuvo en cuenta que su longitud debería de ser de por lo menos 5 pulgadas, ya que este es el radio de las hélices 1045 utilizadas. A su vez, se modeló una ranura en la que se ubicaría el motor con agujeros para los tornillos que lo montan al brazo. Para acoplarlo al chasis, se utilizan dos tornillos M3 con cabeza hexagonal sujetos con tuercas autofrenantes. Esta elección de tuercas se hizo debido a que las vibraciones presentes en el dron pueden aflojar las tuercas normales y fueron usadas en todo el modelo por el mismo motivo. Encima del brazo se colocó la ESC, cuyo cable fue asegurado por una pieza plástica ajustada por los mismos tornillos que ajustan el brazo al chasis.

En la figura 10 se observa el conjunto que compone a los brazos del dron. Este mismo incluye, un protector de hélices desmontable, el motor, el ESC, un bloque guía de cables y el brazo.

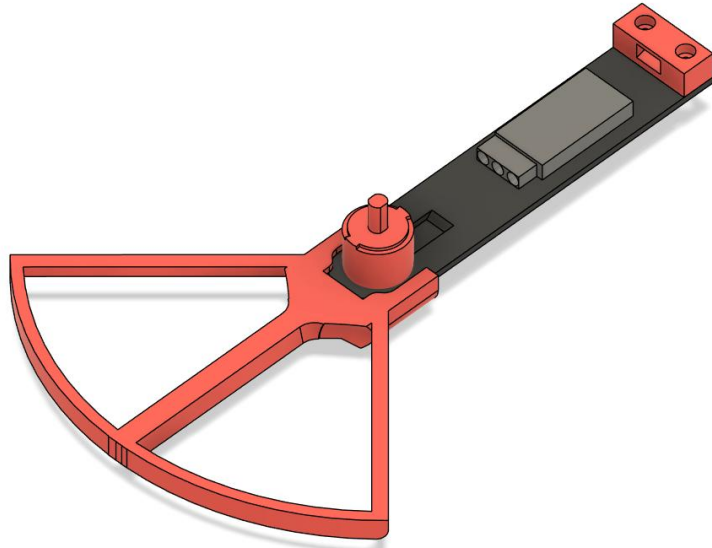


Figura N° 10: *Uno de los brazos del dron con el ESC.*

El diseño del cuerpo principal consiste en una forma en cruz, encontrándose en sus extremos, los anclajes para los brazos. El tamaño de este fue determinado principalmente por la batería, que se encuentra en la parte superior del chasis sujeta por dos abrazaderas y un tope. En la parte inferior del chasis se encuentra el compartimiento en el cual se instaló el MPU6050 y la placa de desarrollo ESP32. Esto se puede observar en la figura 11.

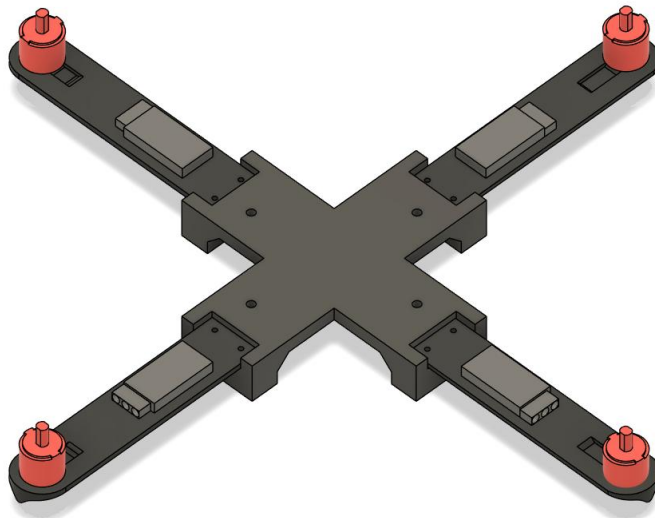


Figura N° 11: *Chasis con brazos acoplados.*

En la figura 12 se observa la batería montada al chasis mediante dos abrazaderas. Las mismas se ajustan con cuatro tornillos M5. Con esto se logra asegurar la batería al chasis manteniendo la posibilidad de removerla fácilmente para cargarla.

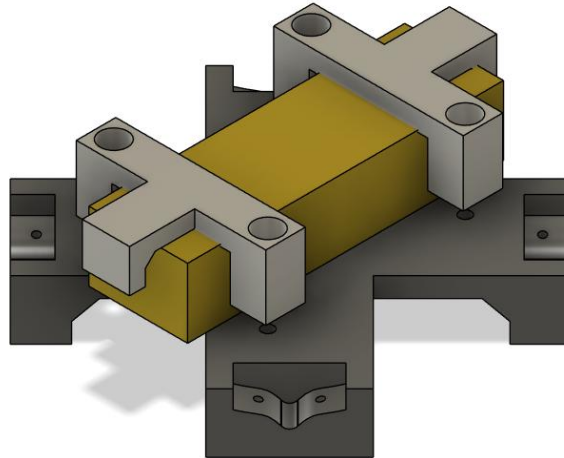


Figura N° 12: *Batería montada sobre el chasis con las bridas.*

Se observa en las figuras 13 y 14, la zona donde se montan el resto de los componentes electrónicos del dron mencionados anteriormente.

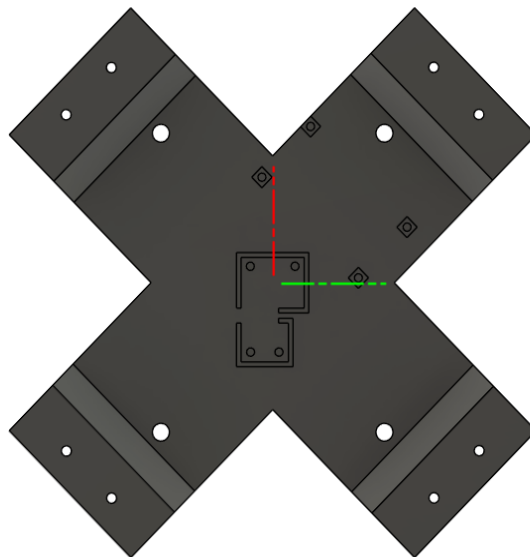


Figura N° 13: *Parte inferior del chasis.*

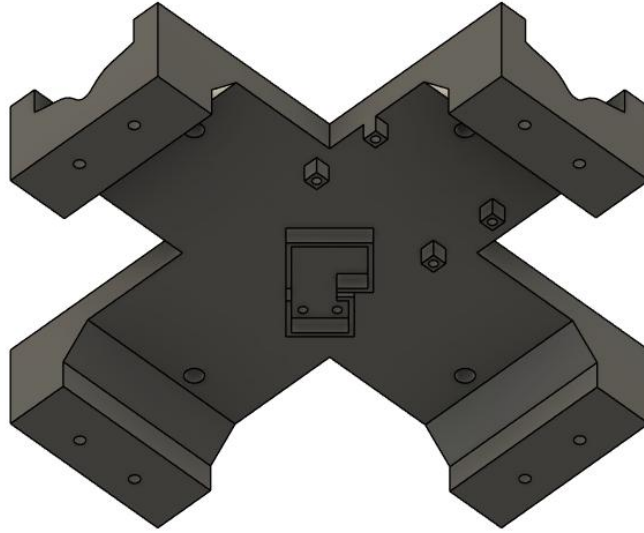


Figura N° 14: *Parte inferior del chasis segunda vista.*

Para absorber los impactos al aterrizar se instalaron cuatro patas con dos rieles sujetos a la parte inferior del dron mediante dos tornillos M3, con tuercas autofrenantes.

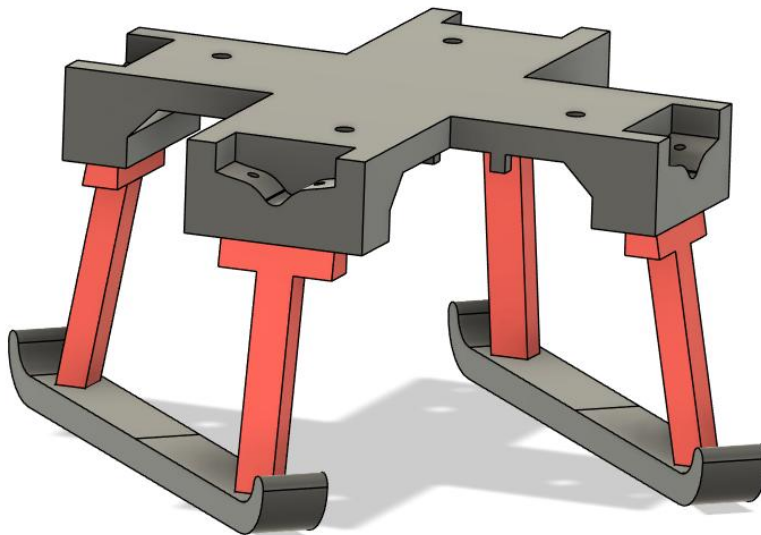


Figura N° 15: *Tren de aterrizaje.*

Una vez finalizado el diseño, se le asignaron a cada una de las partes su peso correspondiente. Nótese que las piezas plásticas difieren ligeramente en densidad debido a las particularidades de las impresiones en 3D. De esta manera, un cuerpo con mayor perímetro por unidad de volumen será más denso ya que las paredes serán más abundantes y tendrá menos relleno. El porcentaje de relleno utilizado es de 20% ya que aumentarlo no mejoraría sus propiedades físicas significativamente e incrementaría el peso de estas.

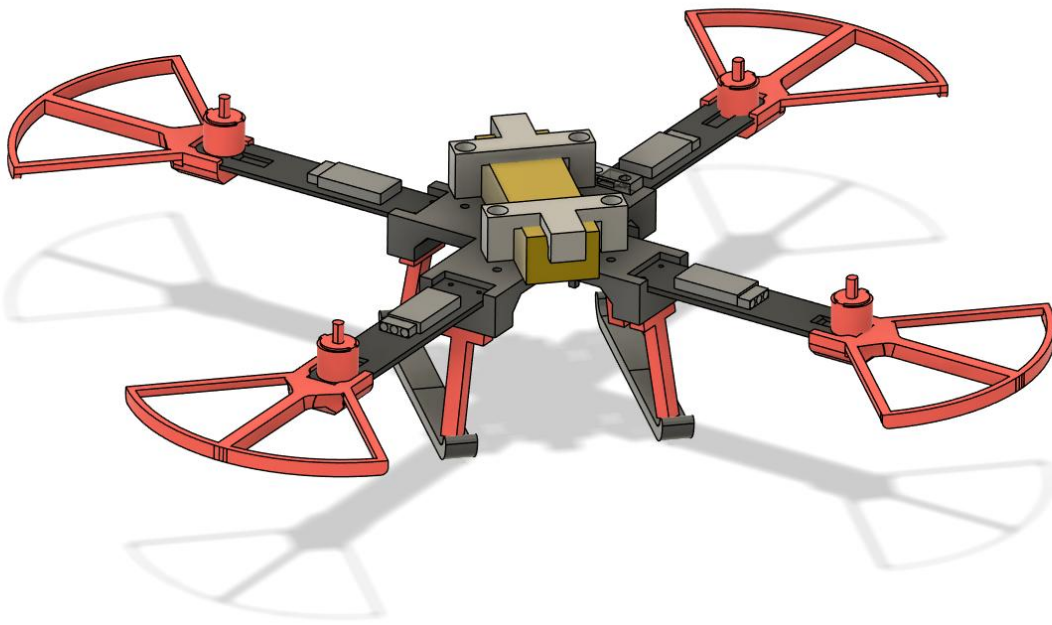


Figura N° 16: *Vista completa del dron.*

En la figura 17 se detallan las medidas máximas del dron vistas desde arriba del mismo. También se muestra la distancia de los ejes de los motores a los ejes X e Y. Estas medidas están dispuestas en milímetros.

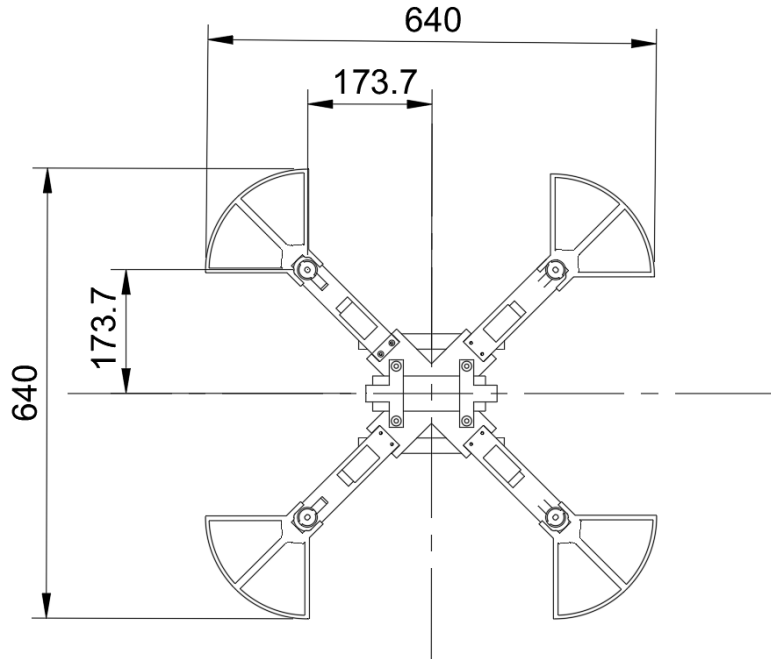


Figura N° 17: Vista superior del dron.

La altura máxima del dron desde el suelo es de 162 milímetros como se indica en la figura 18.

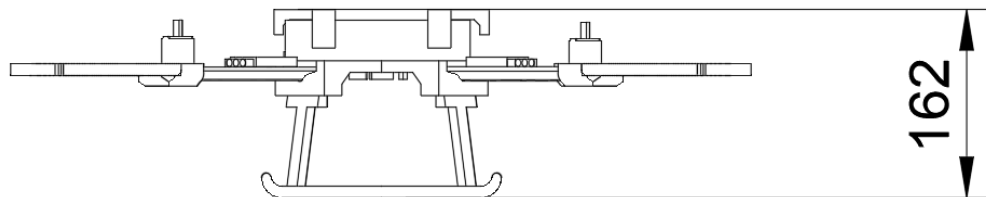


Figura N° 18: Vista lateral del dron.

Con el peso asignado a las distintas partes, el programa calcula el momento de inercia del conjunto (en $\text{kg}\cdot\text{m}^2$) que se utilizara en el modelado matemático:

$$I = \begin{vmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{vmatrix} = \begin{vmatrix} 0.01187 & 0.00011205298 & -0.00032854269 \\ 0.00011205298 & 0.01253 & 0.00052110776 \\ -0.00032854269 & 0.00052110776 & 0.02198 \end{vmatrix} \quad (1)$$

También se obtuvo un estimado del peso total del dron siendo este 1589 g.

CAPITULO 3: MODELADO MATEMÁTICO

En este capítulo se desarrolla el modelo matemático del cuadricóptero, con el propósito de describir con cierto rigor, su comportamiento dinámico y disponer de una representación adecuada para el diseño del controlador.

El objetivo de esta sección es obtener un conjunto de ecuaciones que permitan simular el movimiento del dron y analizar cómo responden sus variables de estado frente a las entradas de control. Como se mencionó en la introducción, un cuadricóptero es un sistema subactuado ya que tiene 6 grados de libertad y 4 entradas de control. Las cuales son:

- Ft: empuje resultante de los 4 motores.
- Torque en X: la diferencia de empuje entre los pares de motores que están de un lado y del otro del eje X, producen un momento en el cuerpo del dron alrededor de este eje.
- Torque en Y: la diferencia de empuje entre los pares de motores que están de un lado y del otro del eje Y, producen un momento en el cuerpo del dron alrededor de este eje.
- Torque en Z: la diferencia de velocidades angulares de los motores con giro horario y antihorario produce un momento al rededor del eje Z.

Para el modelo matemático se hicieron las siguientes suposiciones:

- La estructura del cuadricóptero es rígida y simétrica respecto al eje X.
- El centro de masa está ubicado en el centro geométrico entre los 4 motores.
- Las hélices son rígidas y no se deforman mientras giran. A su vez, producen el mismo empuje a una velocidad de rotación dada, sin que se vean afectadas por su entorno.
- La gravedad es constante y no se ve afectada por la rotación ni la altitud respecto de la tierra.

- Las alteraciones del comportamiento dinámico no son significativas debido a altas velocidades de viento.
- El *ground effect* es despreciable. [13]

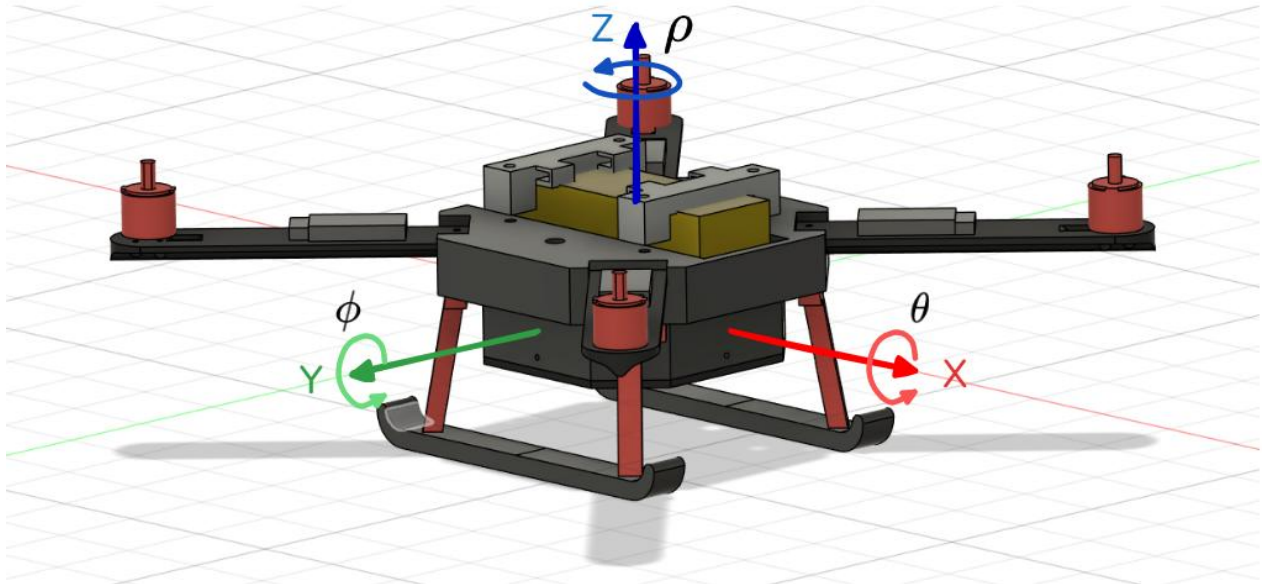


Figura N° 19: Ejes cartesianos y ángulos de Euler.

Para relacionar el sistema de referencia móvil del dron con el sistema del mundo, se requieren hacer transformaciones por medio de matrices. A continuación, se presentan las matrices de rotación y de transformaciones angulares en \mathbb{R}^3 . Estas se utilizan para pasar los vectores expresados de un sistema al otro.

$$R_{zyx}(\theta, \phi, \rho) = \begin{vmatrix} \cos(\phi)\cos(\rho) & \sin(\theta)\sin(\phi)\cos(\rho) - \cos(\theta)\sin(\rho) & \cos(\theta)\sin(\phi)\cos(\rho) + \sin(\theta)\sin(\rho) \\ \cos(\phi)\sin(\rho) & \sin(\theta)\sin(\phi)\sin(\rho) + \cos(\theta)\cos(\rho) & \cos(\theta)\sin(\phi)\sin(\rho) - \sin(\theta)\cos(\rho) \\ -\sin(\phi) & \sin(\theta)\cos(\phi) & \cos(\theta)\sin(\phi) \end{vmatrix} \quad (2)$$

$$T(\theta, \phi) = \begin{vmatrix} 1 & \sin(\theta)\tan(\phi) & \cos(\theta)\tan(\phi) \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \frac{\sin(\theta)}{\cos(\phi)} & \frac{\cos(\theta)}{\cos(\phi)} \end{vmatrix} \quad (3)$$

Los vectores que se emplearon en el modelado matemático son los siguientes:

$$v = [\dot{x} \ \dot{y} \ \dot{z}] \quad (4)$$

$$v_B = [u \ v \ w] \quad (5)$$

$$\omega = [\dot{\theta} \ \dot{\phi} \ \dot{\rho}] \quad (6)$$

$$\omega_B = [p \ q \ r] \quad (7)$$

Donde:

- v : es un vector que contiene la velocidad lineal del cuadricóptero respecto al mundo.
- ω : es un vector que contiene la velocidad angular del cuadricóptero respecto al mundo.
- $R_{zyx}(\theta, \phi, \rho)$: es la matriz de rotación.
- v_B : es un vector que contiene la velocidad lineal con referencia al cuerpo del dron.
- ω_B : es un vector que contiene la velocidad angular con referencia al cuerpo del dron.
- $T(\theta, \phi)$: es la matriz de transformaciones angulares.

Los vectores referenciados al cuerpo del dron que se mostraron anteriormente se pueden transformar del sistema de coordenadas inercial del dron al sistema de referencia del mundo. Con ello se puede lograr que el dron se mueva con respecto al mundo y no a su propio sistema de ejes inerciales, como sucedería en un dron FPV (First Person View). Esto es principalmente empleado para los drones que hacen seguimiento de trayectorias y los que se coordinan en enjambres de drones.

Del producto de la matriz de rotación con el vector v_B y el producto de la matriz de transformaciones angulares con el vector ω_B resulta el siguiente modelo cinemático:

$$v = R \cdot v_B \quad (8)$$

$$\omega = T \cdot \omega_B \quad (9)$$

$$\begin{cases} \dot{x} = w[\sin(\theta)\sin(\rho) + \cos(\theta)\cos(\rho)\sin(\phi)] - v[\sin(\rho)\cos(\theta) - \sin(\theta)\cos(\rho)\sin(\phi)] + u[\cos(\phi)\cos(\rho)] \\ \dot{y} = v[\cos(\theta)\cos(\rho) + \sin(\phi)\sin(\rho)\sin(\theta)] - w[\cos(\rho)\sin(\theta) - \cos(\theta)\sin(\rho)\sin(\phi)] + u[\cos(\phi)\sin(\rho)] \\ \dot{z} = w[\cos(\theta)\cos(\phi)] - u \cdot \sin(\phi) + v[\cos(\phi)\sin(\theta)] \\ \dot{\theta} = p + r[\cos(\theta) \cdot \tan(\phi)] + q[\sin(\theta)\tan(\phi)] \\ \dot{\phi} = q \cdot \cos(\theta) - r \cdot \sin(\theta) \\ \dot{\rho} = r \frac{\cos(\theta)}{\cos(\phi)} + q \frac{\sin(\theta)}{\cos(\phi)} \end{cases} \quad (10)$$

Conforme a la segunda ley de newton, las fuerzas actuantes sobre el cuadricóptero son:

$$m(\omega_B \times v_B + \dot{v}_B) = f_B \quad (11)$$

Donde m es la masa, $\omega_B \times v_B$ es la aceleración centrípeta y f_B es la fuerza total:

$$f_B = [f_x \ f_y \ f_z] \quad (12)$$

Las ecuaciones de Euler para dinámica de cuerpos rígidos determinan el torque aplicado al dron:

$$I \cdot \dot{\omega}_B + \omega_B \times (I \cdot \omega_B) = M_B \quad (13)$$

Donde m_B es el torque total e I es el tensor de inercia:

$$M_B = [M_x \ M_y \ M_z] \quad (14)$$

$$I = \begin{vmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{vmatrix} = \begin{vmatrix} 0.01187 & 0.00011205298 & -0.00032854269 \\ 0.00011205298 & 0.01253 & 0.00052110776 \\ -0.00032854269 & 0.00052110776 & 0.02198 \end{vmatrix} \quad (1)$$

Entonces el modelo dinámico del cuerpo del dron resulta:

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} - pw + ru) \\ f_z = m(\dot{w} + pv - qu) \\ M_x = [\dot{p} \cdot I_{xx} + \dot{q} \cdot I_{xy} + \dot{r} \cdot I_{xz}] + [qp \cdot I_{xz} + q^2 \cdot I_{yz} + qr \cdot I_{zz} - (rp \cdot I_{xy} + rq \cdot I_{yy} + r^2 \cdot I_{yz})] \\ M_y = [\dot{p} \cdot I_{xy} + \dot{q} \cdot I_{yy} + \dot{r} \cdot I_{yz}] + [p^2 \cdot I_{xz} + pq \cdot I_{yz} + pr \cdot I_{zz} - (rp \cdot I_{xx} + rq \cdot I_{xy} + r^2 \cdot I_{xz})] \\ M_z = [\dot{p} \cdot I_{xz} + \dot{q} \cdot I_{yz} + \dot{r} \cdot I_{zz}] + [p^2 \cdot I_{xy} + pq \cdot I_{yy} + pr \cdot I_{yz} - (qp \cdot I_{xx} + q^2 \cdot I_{xy} + qr \cdot I_{xz})] \end{cases} \quad (15)$$

Las fuerzas externas que actúan sobre el dron se pueden describir como:

$$f_B = f_t \cdot \hat{e}_3 + f_p - mgR^T \cdot \hat{e}_z \quad (16)$$

Donde:

- f_t : es el empuje generado por las hélices.
- \hat{e}_3 : es el versor en el eje **Z** del dron.
- $f_p = [f_{px}, f_{py}, f_{pz}]^T$: son las fuerzas producidas por perturbaciones.
- \hat{e}_z : es el versor en el eje **Z** inercial.

Los momentos externos que actúan sobre el dron están dados por:

$$M_B = \tau_B - g_a + \tau_p \quad (17)$$

Donde:

- g_a : son los momentos giroscópicos causados por la rotación de los motores y el cuerpo.
- $\tau_B = [\tau_x, \tau_y, \tau_z]^T$: son los torques producidos por las hélices sobre el cuerpo.
- $\tau_p = [\tau_{px}, \tau_{py}, \tau_{pz}]^T$: son los torques producidos por perturbaciones sobre el dron.

A modo de simplificación del modelo, se tomaron todas las perturbaciones externas en un vector de perturbaciones y se consideró que el controlador del dron buscará corregir los efectos inesperados producidos por las mismas.

El modelo dinámico del dron obtenido de las ecuaciones anteriores es el siguiente:

$$\begin{cases} f_{px} + mg \cdot \sin(\phi) = m(\dot{u} + qw - rv) \\ f_{py} - mg \cdot \cos(\phi) \cdot \sin(\theta) = m(\dot{v} - pw + ru) \\ f_t + f_{pz} - mg \cdot \cos(\phi) \cdot \cos(\theta) = m(\dot{w} + pv - qu) \\ \tau_x + \tau_{px} = [\dot{p} \cdot I_{xx} + \dot{q} \cdot I_{xy} + \dot{r} \cdot I_{xz}] + [qp \cdot I_{xz} + q^2 \cdot I_{yz} + qr \cdot I_{zz} - (rp \cdot I_{xy} + rq \cdot I_{yy} + r^2 \cdot I_{yz})] \\ \tau_y + \tau_{py} = [\dot{p} \cdot I_{xy} + \dot{q} \cdot I_{yy} + \dot{r} \cdot I_{yz}] + [p^2 \cdot I_{xz} + pq \cdot I_{yz} + pr \cdot I_{zz} - (rp \cdot I_{xx} + rq \cdot I_{xy} + r^2 \cdot I_{xz})] \\ \tau_z + \tau_{pz} = [\dot{p} \cdot I_{xz} + \dot{q} \cdot I_{yz} + \dot{r} \cdot I_{zz}] + [p^2 \cdot I_{xy} + pq \cdot I_{yy} + pr \cdot I_{yz} - (qp \cdot I_{xx} + q^2 \cdot I_{xy} + qr \cdot I_{xz})] \end{cases} \quad (18)$$

3.1 Dinámicas de los actuadores:

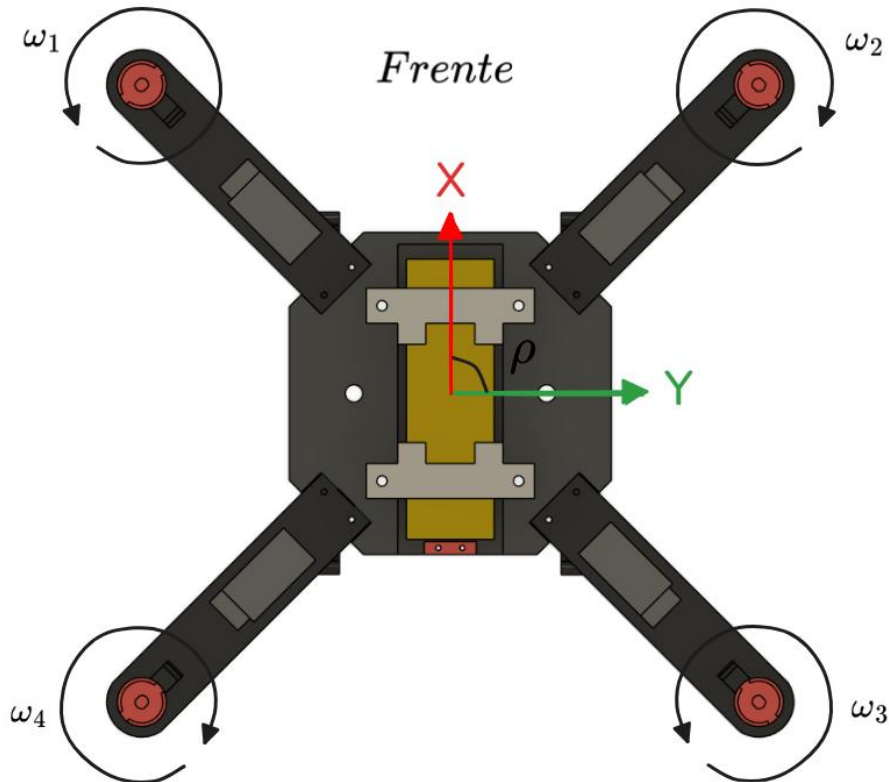


Figura N° 20: Vista superior del dron.

La fuerza resultante de cada uno de los motores va a ser igual a la sumatoria de las cuatro acciones de control salientes del controlador, fuerza en z y los 3 torques en x, y, z con sus respectivos signos, divididas por 4. Siendo l la distancia de los motores al centro del dron, las ecuaciones resultantes son las siguientes:

$$F_{\text{mot1}} = \frac{Ft + \tau_x \cdot l - \tau_y \cdot l + \tau_z}{4} \quad (19)$$

$$F_{\text{mot2}} = \frac{Ft - \tau_x \cdot l - \tau_y \cdot l - \tau_z}{4} \quad (20)$$

$$F_{\text{mot3}} = \frac{Ft - \tau_x \cdot l + \tau_y \cdot l + \tau_z}{4} \quad (21)$$

$$F_{\text{mot4}} = \frac{Ft + \tau_x \cdot l + \tau_y \cdot l - \tau_z}{4} \quad (22)$$

Se realizó la medición de empuje de los motores utilizando un banco de pruebas (ver capítulo 5 para más detalles) del que se obtuvo, por interpolación cuadrática de los valores obtenidos una ecuación que relaciona la señal PWM ingresante a el ESC con el empuje resultante del motor.

$$PWM_i = A \cdot F_{\text{mot}_i}^2 + B \cdot F_{\text{mot}_i} + C \quad (23)$$

Una vez obtenidas las fuerzas necesarias de cada motor, se ingresa a la curva interpolada con el valor de fuerza y se traduce a un valor de PWM que es posteriormente alimentado a los motores.

3.2 Modelado por variable de estado no lineal

El vector que representa los estados del dron es:

$$\mathbf{x} = [\theta, \phi, \rho, p, q, r, u, v, w, x, y, z]^T \quad (24)$$

Para simplificar el despeje, los términos no diagonales del tensor de inercia fueron despreciados por ser 2 órdenes de magnitud menores a las inercias principales.

De las ecuaciones anteriores de dinámica y cinemática se obtuvo que:

$$\left\{ \begin{array}{l} \dot{\theta} = p + r[\cos(\theta)\tan(\phi)] + q[\sin(\theta)\tan(\phi)] \\ \dot{\phi} = q \cdot \cos(\theta) - r \cdot \sin(\theta) \\ \dot{\rho} = r \frac{\cos(\theta)}{\cos(\phi)} + q \frac{\sin(\theta)}{\cos(\phi)} \\ \dot{p} = \frac{\tau_x + \tau_{px}}{I_{xx}} - qr \frac{I_{zz} + I_{yy}}{I_{xx}} \\ \dot{q} = \frac{\tau_y + \tau_{py}}{I_{yy}} - pr \frac{I_{zz} + I_{xx}}{I_{yy}} \\ \dot{r} = \frac{\tau_z + \tau_{pz}}{I_{zz}} - pq \frac{I_{xx} + I_{yy}}{I_{zz}} \\ \dot{u} = \frac{f_{px}}{m} + g \cdot \sin(\phi) - qw + rv \\ \dot{v} = \frac{f_{py}}{m} - g \cdot \cos(\phi)\sin(\theta) + pw - ru \\ \dot{w} = \frac{f_t + f_{pz}}{m} - g \cdot \cos(\phi)\cos(\theta) - pv + qu \\ \dot{x} = w[\sin(\theta)\sin(\rho) + \cos(\theta)\cos(\rho)\sin(\phi)] - v[\sin(\rho)\cos(\theta) - \sin(\theta)\cos(\rho)\sin(\phi)] + u[\cos(\phi)\cos(\rho)] \\ \dot{y} = v[\cos(\theta)\cos(\rho) + \sin(\phi)\sin(\rho)\sin(\theta)] - w[\cos(\rho)\sin(\theta) - \cos(\theta)\sin(\rho)\sin(\phi)] + u[\cos(\phi)\sin(\rho)] \\ \dot{z} = w[\cos(\theta)\cos(\phi)] - u \cdot \sin(\phi) + v[\cos(\phi)\sin(\theta)] \end{array} \right. \quad (25)$$

Este modelo obtenido corresponde a un sistema no lineal, ya que las ecuaciones de movimiento incluyen términos no lineales como las funciones trigonométricas seno(x) , coseno(x) y tangente(x), las cuales se escribieron en inglés como sin(x), cos(x) y tan(x) respectivamente.

Sin embargo, dado que en este trabajo se diseñará un controlador empleando técnicas de control lineal clásicas, es necesario linealizar el sistema en torno a un punto de equilibrio. Esta simplificación permite obtener un modelo en variables de estado lineal, adecuado para el diseño e implementación de un controlador por realimentación de estados, por ejemplo, un controlador LQR.

3.3 Linealización:

La linealización es un proceso en el que un sistema no lineal se aproxima a uno lineal en torno a un punto de equilibrio. Esto se realiza porque muchos controladores clásicos, como el LQR, requieren que el modelo sea lineal para poder diseñarse y analizar su estabilidad. En este caso, se linealiza el modelo dinámico del dron alrededor de una condición de vuelo estacionario (hover), donde las variaciones angulares y de velocidad son pequeñas. Esta simplificación permite aplicar técnicas de control lineal manteniendo una buena aproximación del comportamiento real del sistema cerca de dicho punto.

Se definió el vector de control:

$$u = [f_t, \tau_x, \tau_y, \tau_z]^T \quad (26)$$

Para la linealización se asume que $\cos(x) = 1$ y $\sin(x) = x$. Esto es válido para ángulos pequeños.

El modelo linealizado en variables de estado es:

$$f(x, u) = \begin{cases} \dot{\theta} \approx p + r\phi + q \cdot \theta\phi \\ \dot{\phi} \approx q \cdot -r \cdot \theta \\ \dot{\rho} \approx r + q \cdot \theta \\ \dot{p} \approx \frac{\tau_x + \tau_{px}}{I_{xx}} - qr \frac{I_{zz} + I_{yy}}{I_{xx}} \\ \dot{q} \approx \frac{\tau_y + \tau_{py}}{I_{yy}} - pr \frac{I_{zz} + I_{xx}}{I_{yy}} \\ \dot{r} \approx \frac{\tau_z + \tau_{pz}}{I_{zz}} - pq \frac{I_{xx} + I_{yy}}{I_{zz}} \\ \dot{u} \approx \frac{f_{px}}{m} + g \cdot \phi - qw + rv \\ \dot{v} \approx \frac{f_{py}}{m} - g \cdot \theta + pw - ru \\ \dot{w} \approx \frac{f_t + f_{pz}}{m} - g \cdot -pv + qu \\ \dot{x} \approx w[\theta \cdot \rho + \phi] - v[\rho - \theta \cdot \phi] + u \\ \dot{y} \approx v[1 + \phi \cdot \rho \cdot \theta] - w[\theta - \rho \cdot \phi] + u \cdot \rho \\ \dot{z} \approx w - u \cdot \phi + v \cdot \theta \end{cases} \quad (27)$$

Para linealizar se debe establecer un punto de equilibrio \bar{x} , para el cual se eligió:

$$\bar{x} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \bar{x}, \bar{y}, \bar{z}]^T \quad (28)$$

Este punto de equilibrio se obtiene con la entrada fija \bar{u} :

$$\bar{u} = [mg, 0, 0, 0] \quad (29)$$

A partir de 27, se define la matriz A :

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}; u=\bar{u}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (30)$$

A su vez, la matriz B se define como:

$$B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{x=\bar{x}; u=\bar{u}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

Por otro lado, se definió el vector de perturbaciones d , que incluyen las fuerzas y torques no modelados.

$$d = [f_{dx}, f_{dy}, f_{dz}, \tau_{dx}, \tau_{dy}, \tau_{dz}] \quad (32)$$

A continuación, se define la matriz de perturbaciones D.

$$D = \frac{\partial f(x, u, d)}{\partial d} \Big|_{x=\bar{x}; u=\bar{u}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_{zz}} \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

El modelo lineal es expresado de la siguiente forma:

$$\dot{x} = A \cdot x + B \cdot u + D \cdot d \quad (34)$$

Como resultado el modelo linealizado resulta:

$$f(x, u) = \begin{cases} \dot{\theta} = p \\ \dot{\phi} = q \\ \dot{\rho} = r \\ \dot{p} = \frac{\tau_x + \tau_{dx}}{I_{xx}} \\ \dot{q} = \frac{\tau_y + \tau_{dy}}{I_{yy}} \\ \dot{r} = \frac{\tau_z + \tau_{dz}}{I_{zz}} \\ \dot{u} = \frac{f_{dx}}{m} + g \cdot \phi \\ \dot{v} = \frac{f_{dy}}{m} - g \cdot \theta \\ \dot{w} = \frac{f_t + f_{dz}}{m} \\ \dot{x} = u \\ \dot{y} = v \\ \dot{z} = w \end{cases} \quad (35)$$

CAPITULO 4: DISEÑO DEL CONTROLADOR Y SIMULACIÓN

En este capítulo se presenta el diseño del controlador LQR y su validación a través de simulaciones.

El objetivo fue comprobar que, a partir del modelo matemático obtenido previamente, es posible diseñar un controlador capaz de mantener la estabilidad del dron ante distintas condiciones de operación, siempre priorizando la simplicidad y bajo costo de implementación.

Las simulaciones realizadas permitieron validar el diseño antes de llevarlo a la práctica, reduciendo riesgos y asegurando que el prototipo cumpla con su fin didáctico y experimental.

Para controlar los 12 estados del modelo completo del dron sería necesario contar con sensores tales como un GPS, un magnetómetro y un barómetro, que permitan medir la posición absoluta y la orientación respecto al mundo.

Sin embargo, debido a la ausencia de estos sensores en el prototipo, se decidió reducir el sistema de control a los 9 primeros estados, correspondientes a los ángulos de Euler, las velocidades angulares y las velocidades lineales. De esta manera, se priorizó la estabilización del dron en sus variables más críticas para el vuelo, dejando los tres estados de posición a lazo abierto.

Para esto, se optó por un controlador LQR (Linear Quadratic Regulator) el cual obtiene una matriz de ganancias que minimizan una función de costos que incluye las matrices Q y R, las cuales penalizan el error de cada estado y el uso de las acciones de control, respectivamente.

4.1 Esquema de control:

El controlador es alimentado con el error en los estados, siendo este la diferencia entre el estado actual y el ideal. El estado ideal del dron consiste en tener los ángulos en cero grados y velocidad angular nula en los tres ejes (hover). Esto es debido a que, para mantener su posición, el dron no debe estar inclinado. El controlador utiliza estas entradas, y las multiplica por un vector de ganancias para obtener la acción de control.

Luego de calcular la acción de control, a la cual se suma la acción del usuario en el joystick, esta es traducida a pulsos PWM según lo establecido por la ecuación 23 para ingresarlos al controlador electrónico de velocidad (ESC). A continuación, los sensores proveen la

aceleración y velocidad angular del dron que son utilizadas para estimar su velocidad lineal por un estimador y sus ángulos mediante el filtro de Madgwick. [9]

En la figura 21 se muestra un diagrama de bloques describiendo el funcionamiento del dron como se ha explicado en este capítulo.

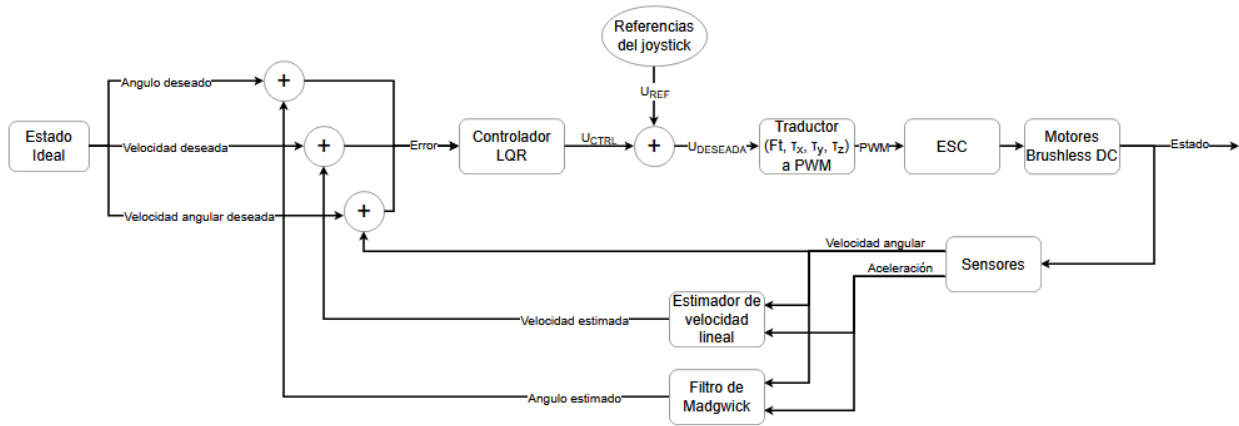


Figura N° 21: Diagrama de bloques del funcionamiento del dron.

4.2 Controlabilidad y Observabilidad:

Para verificar que el sistema sea controlable y observable, se utilizan las definiciones clásicas de la teoría de control. Se busca verificar la observabilidad del sistema en caso de que sea necesario implementar un observador para prescindir de algún sensor.

La matriz de controlabilidad se define como:

$$C = [AB \ A^2B \ \dots \ A^{n-1}B] \quad (36)$$

Donde A (30) y B (31) son las matrices del sistema en variables de estado.

El sistema es controlable si y solo si el rango de C es igual a n , siendo n la dimension del vector de estados, en este caso siendo 12 para el controlador entero y 9 sin el control posicional.

De forma análoga, la matriz de observabilidad se define como:

$$O = [C \ CA \ CA^2 \ \dots \ CA^{n-1}]^T \quad (37)$$

Donde C es la matriz de salida del sistema.

El sistema es observable si y solo si el rango de O es igual a n .

Se realizaron ambas verificaciones en MATLAB mediante el siguiente código:

```
O = [C_reducido
      C_reducido*A_reducido
      C_reducido*A_reducido^2
      C_reducido*A_reducido^3
      C_reducido*A_reducido^4
      C_reducido*A_reducido^5
      C_reducido*A_reducido^6
      C_reducido*A_reducido^7
      C_reducido*A_reducido^8
      ];
% Rango de la matriz de observabilidad del sistema reducido
R_O = rank(O);
fprintf("el rango de la matriz de observabilidad del modelo reducido es:
%d\n",R_O)

% Se calcula la matriz de controlabilidad

Controlab = [B_reducido A_reducido*B_reducido A_reducido^2*B_reducido
A_reducido^3*B_reducido A_reducido^4*B_reducido A_reducido^5*B_reducido
A_reducido^6*B_reducido A_reducido^7*B_reducido
A_reducido^8*B_reducido];
% Rango de la matriz de controlabilidad del sistema
R_C = rank(Controlab);
fprintf("el rango de la matriz de controlabilidad del modelo reducido
es: %d\n",R_C)
```

Snippet N° 1: Controlabilidad y Observabilidad.

Donde se observaron los siguientes resultados:

```

el rango de la matriz de observabilidad es: 12
el rango de la matriz de controlabilidad es: 12
el rango de la matriz de observabilidad del modelo reducido es: 9
el rango de la matriz de controlabilidad del modelo reducido es: 9

```

Figura N° 22: Resultados de Controlabilidad y Observabilidad

En ambos casos se observa que el rango es igual a la dimensión del sistema, confirmando la controlabilidad u observabilidad.

4.3 Controlador LQR:

Se buscó implementar un controlador del tipo LQR (Linear Quadratic Regulator), el cual halla una matriz de ganancias K que minimice la función de costos definida por el vector de estados x , las matrices A y B del sistema y las matrices Q y R, las cuales penalizan el error de cada estado y el uso de las acciones de control respectivamente.

Las matrices Q y R fueron establecidas como matrices identidad, esto significa que el control penalizará todos los estados y las acciones de control de la misma manera. No es necesariamente lo óptimo en la implementación física y deberá calibrarse cuando se realicen las pruebas de vuelo del dron para encontrar un conjunto de valores que proporcione un comportamiento deseable.

Como resultado, la matriz K del controlador tiene los siguientes valores:

$$K_{reducido\ 9} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 4.6557 & 0 & 0 & 1.0538 & 0 & 0 & 0 & -1.0000 & 0 \\ 0 & 4.6620 & 0 & 1.0568 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 1.0217 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (38)$$

El controlador LQR se implementa mediante una ley de control por realimentación de estados, expresada como:

$$u = -Kx \quad (39)$$

Donde u es el vector de acciones de control (fuerza total y torques), k es la matriz de ganancias obtenida mediante la función LQR, y x es el vector de estados del dron. Esta ecuación indica que el controlador calcula las acciones necesarias para minimizar el error de los estados respecto a sus valores deseados, penalizando tanto los desvíos como las acciones excesivas de control. En la implementación práctica, esta acción se combina con las entradas del usuario (desde el joystick), resultando en una corrección continua que mantiene la estabilidad del dron.

```
% matrices LQR modelo reducido

% Matriz Q: penaliza los errores de cada estado

Q_reducido = eye(9); % Inicializamos una matriz identidad de 9x9

% Penalización para la actitud
Q_reducido(1,1) = 10; % Penaliza las inclinaciones en pitch %15
Q_reducido(2,2) = 10; % Penaliza las inclinaciones en roll
Q_reducido(3,3) = 1; % Penaliza las inclinaciones en yaw

% Penalización para las velocidades angulares
Q_reducido(4,4) = 1; % Penaliza la velocidad angular en pitch %0.14
Q_reducido(5,5) = 1; % Penaliza la velocidad angular en roll
Q_reducido(6,6) = 1; % Penaliza la velocidad angular en yaw

% Penalización para las velocidades lineales
Q_reducido(7,7) = 1-0.99; % Penaliza la velocidad lineal en x
% Valores preferidos 1-0.99
Q_reducido(8,8) = 1-0.99; % Penaliza la velocidad lineal en y
% Valores preferidos 1-0.99
Q_reducido(9,9) = 1-0.99; % Penaliza la velocidad lineal en z
% Valores preferidos 1-0.99

% Matriz R: penaliza las acciones de control

R_reducido = eye(4); % Inicializamos una matriz identidad de 4x4

R_reducido(1,1) = 1; % Penaliza las acciones de control en ft
R_reducido(2,2) = 1; % Penaliza las acciones de control en tau_x
R_reducido(3,3) = 1; % Penaliza las acciones de control en tau_y
R_reducido(4,4) = 1; % Penaliza las acciones de control en tau_z

% Obtencion de la matriz de ganancias K para el controlador LQR

K_reducido_9 = lqr(A_reducido,B_reducido,Q_reducido,R_reducido);
```

Snippet N° 2: Matrices Q y R en MATLAB.

4.4 Simulación:

Para la simulación se implementó el modelo completo con las ecuaciones no lineales en Simulink. Se realizó una prueba donde el dron se encuentra inicialmente estático en hover y, a lo largo de 60 segundos, se le darán diversas referencias externas simulando un control remoto, las cuales están explicadas a continuación.

En la figura 23 se pueden ver los distintos bloques que componen la simulación del dron. En el centro está el bloque de subsistema, donde está contenido el bloque que posee las ecuaciones del modelo no lineal del dron. A la salida del subsistema "Dron" se obtienen los estados actuales de cada iteración. Estas son ingresadas al controlador luego de pasar por un zero order hold que simula el tiempo discreto del microcontrolador. Posterior a esto, se obtienen las acciones de control solicitadas por el controlador. A estas se le agregan las acciones pedidas por el usuario y son retroalimentadas a la planta.

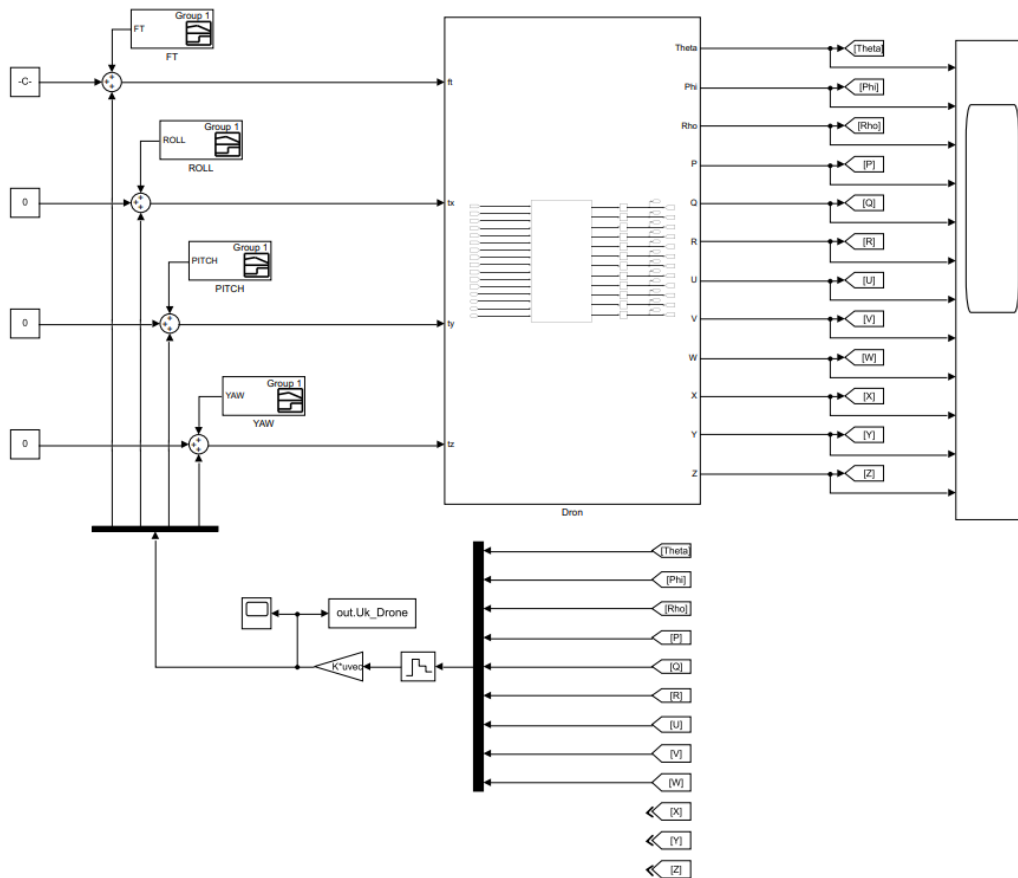


Figura N° 23: Diagrama de bloques en Simulink.

En la figura 24 se presenta el contenido del subsistema mencionado anteriormente. Este tiene como entradas los estados del dron del instante anterior y las entradas de control que vienen de la diferencia entre la referencia y las acciones del controlador. En el centro está el bloque de función que posee el modelo no lineal de la planta. Las salidas de este bloque son integradas y con ello se obtienen los estados actuales del sistema. Estos salen del subsistema y son utilizados para la siguiente iteración.

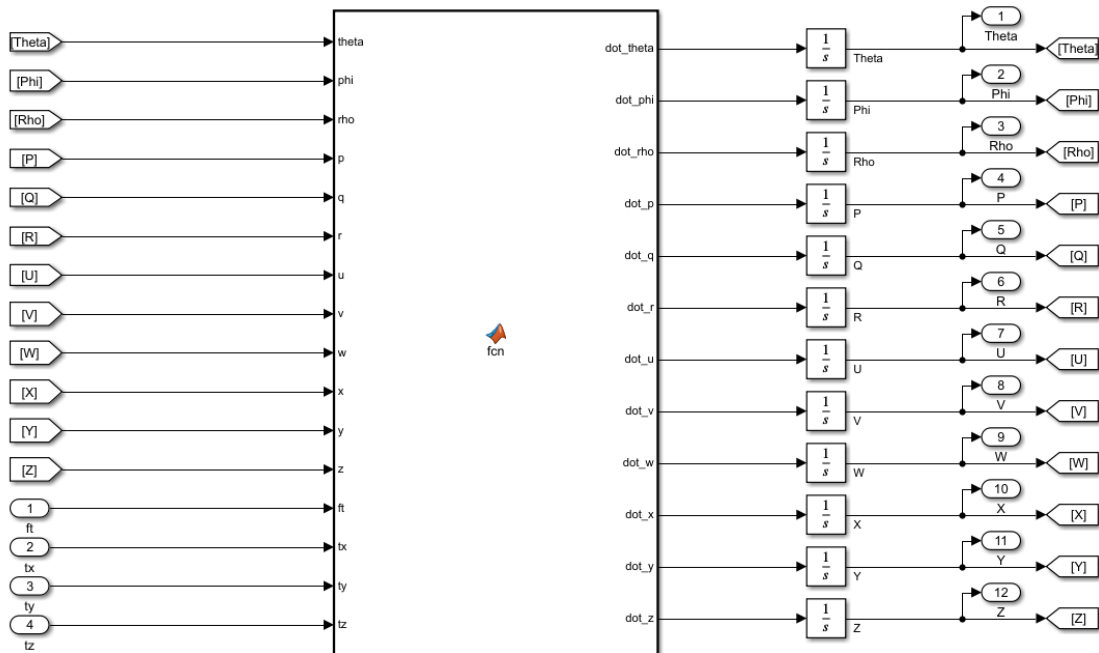


Figura N° 24: Subsistema que contiene a la planta.

En el snippet número 3 se observa el código dentro del bloque “fcn” en simulink. Este contiene el modelo no lineal del dron obtenido de la ecuación número 17.

```

m = 1.500; %aprox 1.584
g = 9.81;
b = 1; % factor de lift

I_x = 0.01187;
I_y = 0.01253;
I_z = 0.02198;

%modelo
dot_theta = p + r*(cos(theta)*tan(phi))+q*(sin(theta)*tan(phi));
dot_phi = q*cos(theta) - r*sin(theta);
dot_rho = r * (cos(theta)/cos(phi)) + q*(sin(theta)/cos(phi));

dot_p = tx/I_x - q*r*(I_z+I_y)/I_x;
dot_q = ty/I_y - p*r*(I_z+I_x)/I_y;
dot_r = tz/I_z - p*q*(I_x+I_y)/I_z;

dot_u = g*sin(phi) - q*w + r*v;
dot_v = -g*cos(phi)*sin(theta) + p*w - r*u;
dot_w = ft/m - g*cos(phi)*cos(theta) -p*v + q*u;

dot_x = w*(sin(theta)*sin(rho)+cos(theta)*cos(rho)*sin(phi)) -
v*(sin(rho)*cos(theta) -
sin(theta)*cos(rho)*sin(phi))+u*(cos(phi)*cos(rho));
dot_y = v*(cos(theta)*cos(rho)+sin(theta)*sin(rho)*sin(phi)) -
w*(cos(rho)*sin(theta) -
cos(theta)*sin(rho)*sin(phi))+u*(cos(phi)*sin(rho));
dot_z = w*(cos(theta)*cos(phi))-u*sin(phi) +v*(cos(phi)*sin(theta));

```

Snippet N° 3: Bloque "fcn".

A continuación, se muestran las referencias que simulan señales del control remoto. En este caso se compone de 4 señales:

- La señal “FT” que se usa para controlar la fuerza de empuje de los motores, que permanece en 0 hasta el segundo 4 y permanece en 1 hasta el segundo 6 volviendo a 0 nuevamente. Con estos valores se espera un ascenso del dron que comience en el segundo 4, con aceleración distinta de 0 y en el segundo 6 permanecería con aceleración igual a 0.

- La referencia "ROLL" la cual induce un torque en el eje X del dron. Se le dio una referencia de 0.7 Nm durante 5 segundos, empezando en el segundo 20.5. Este comportamiento produciría una inclinación del dron sobre el eje X y por consiguiente una velocidad lineal en Y.
- La referencia "PITCH" la cual induce un torque en el eje Y del dron. Se le dio una referencia de 0.2 Nm durante 3.7 segundos, empezando en el segundo 38.8. Este comportamiento produciría una inclinación del dron sobre el eje Y, y por consiguiente una velocidad lineal en X.
- Por último, la señal "YAW" que induce una rotación del dron sobre el eje Z. En la siguiente grafica se muestra que la referencia externa dada es de 0.2 por 3.7 segundos empezando en el segundo 49.1.

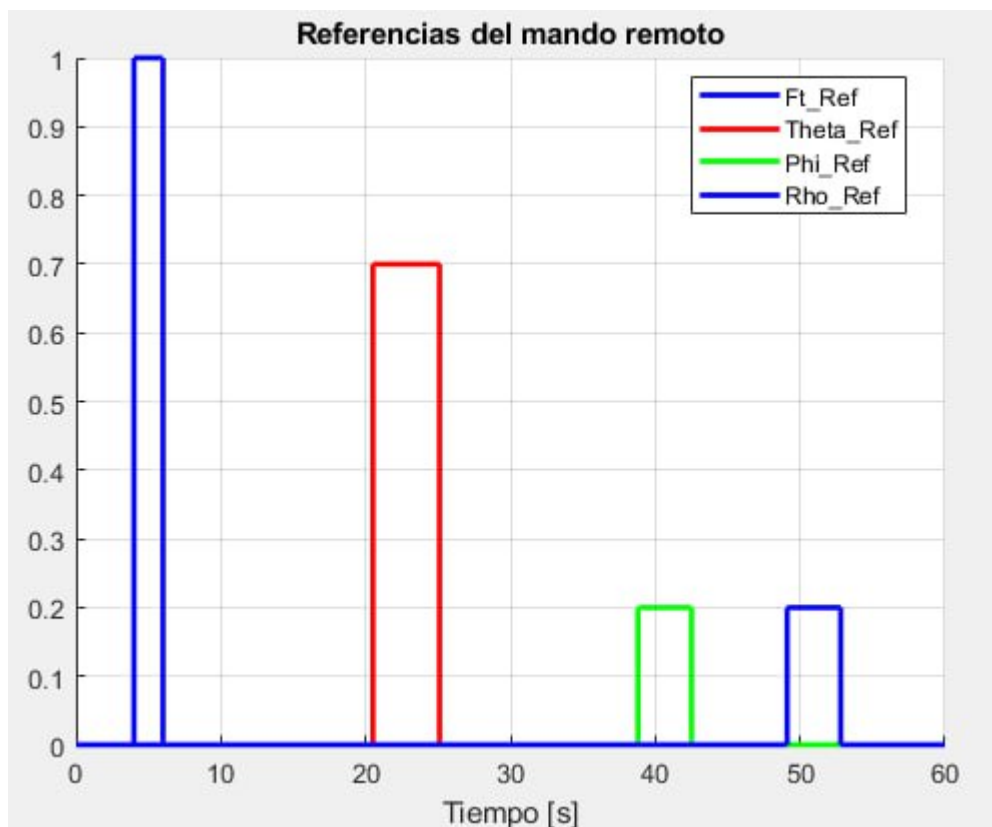


Figura N° 25: Referencias del mando remoto.

Se procedió a correr la simulación con las entradas mostradas y se observó la siguiente dinámica de los estados que se presentan en las figuras 26, 27 y 28.

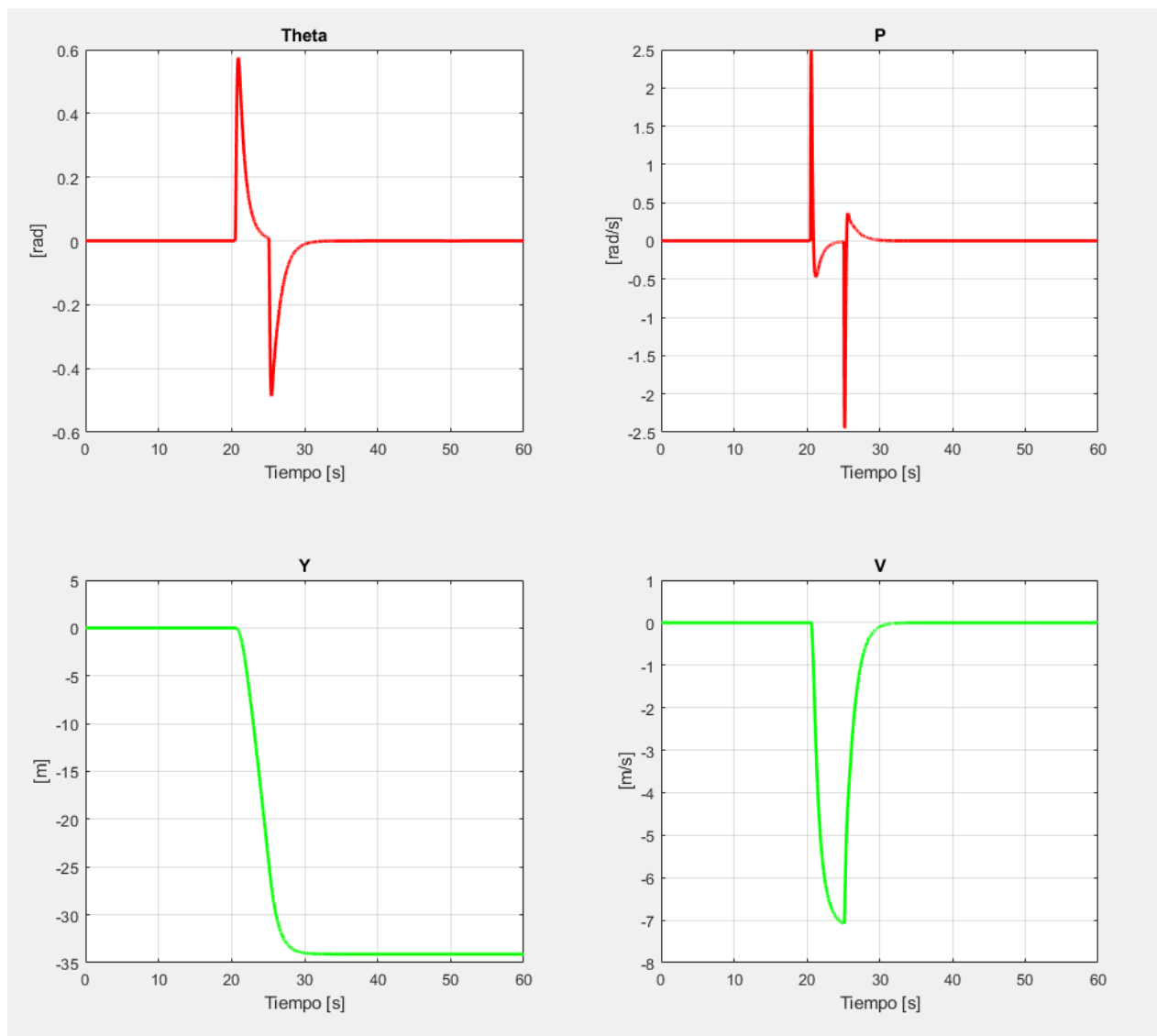


Figura N° 26: Grafico de los estados *Theta*, *P*, *Y* y *V*.

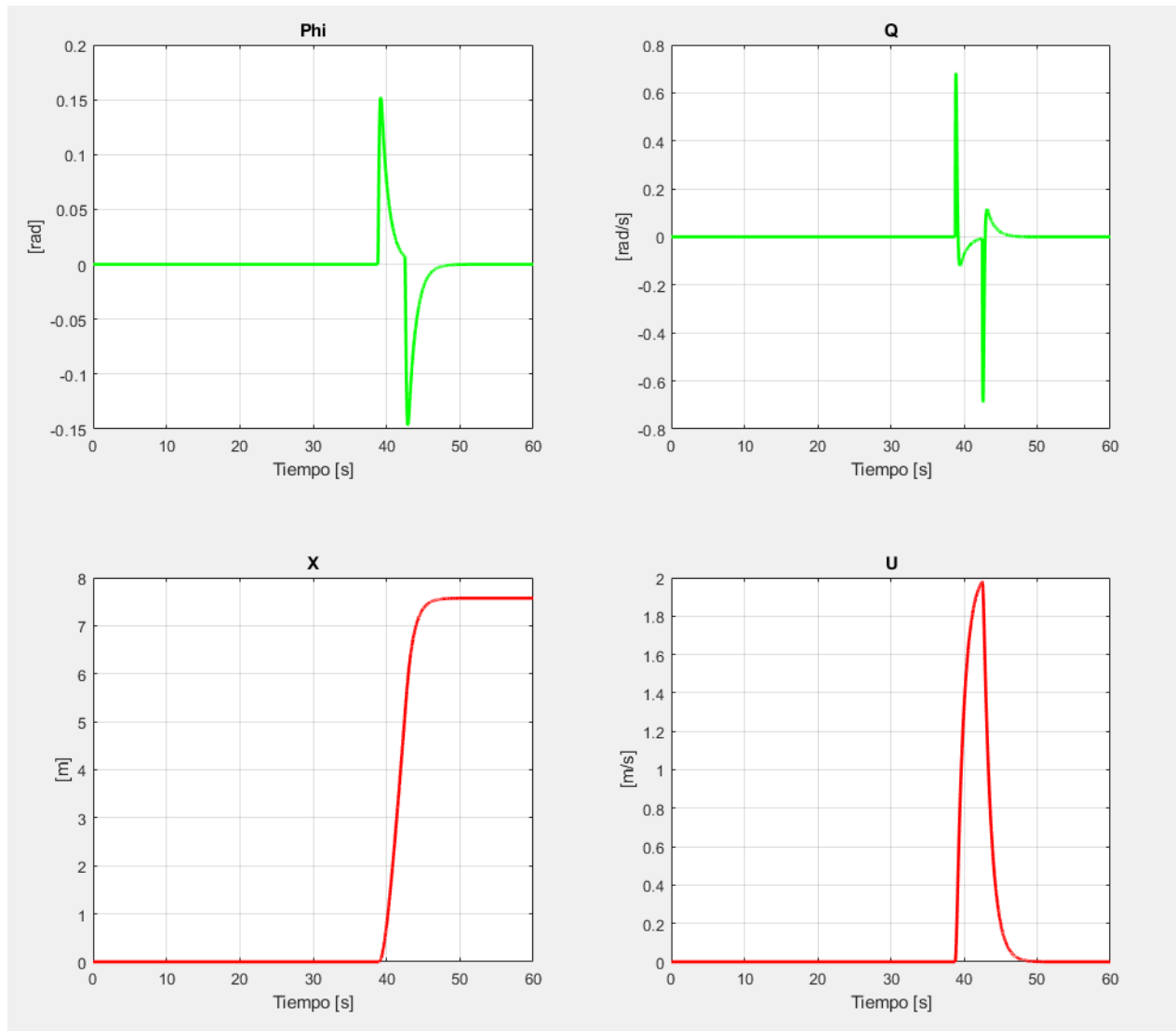


Figura N° 27: Grafico de los estados Φ , Q , X y U .

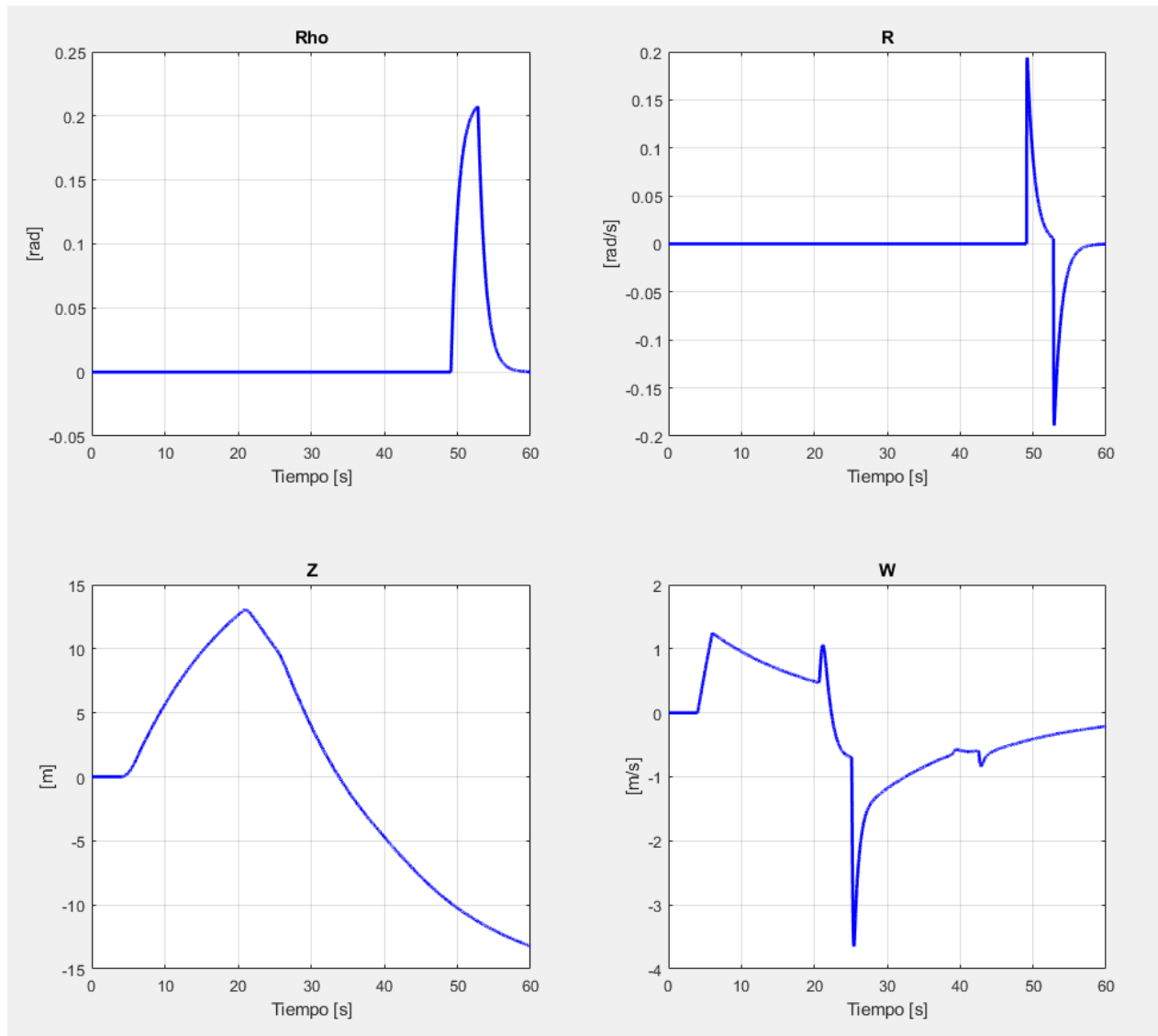


Figura N° 28: Grafico de los estados Rho , R , Z y W .

Se observan las respuestas temporales de los doce estados del dron frente a las señales de referencia aplicadas durante la simulación.

Las variables representadas corresponden a:

- θ (roll), φ (pitch), ρ (yaw): los ángulos de orientación o actitud del dron.
- p , q , r : las velocidades angulares alrededor de los ejes X, Y y Z respectivamente.
- u , v , w : las velocidades lineales en los ejes del cuerpo del dron.
- x , y , z : las posiciones en el marco inercial.

El comportamiento observado es el esperado: ante un cambio en las referencias, los ángulos varían hasta llegar a la referencia con un error mínimo. Luego regresan al cero cuando la señal del usuario vuelve a cero. Gracias a las acciones del controlador LQR no se producen oscilaciones ni sobrepasos.

Las velocidades angulares muestran un comportamiento amortiguado, sin oscilaciones significativas ni sobreimpulsos excesivos, lo que evidencia que la ganancia del controlador es adecuada. Las velocidades lineales responden de manera coherente con las inclinaciones producidas, a modo de ejemplo en el intervalo de tiempo desde los 20 a 30 segundos, se produce una rotación sobre el eje X (roll). Esto genera una velocidad lineal en el eje Y. La respuesta de la velocidad y la posición a lo largo de Y es la esperada.

En conjunto, los resultados confirman que el controlador logra mantener la estabilidad y el seguimiento de referencias sin comportamientos no deseados, validando así la correcta linealización del modelo y la sintonización inicial de las matrices

Estas señales sirvieron para simular las acciones del control remoto. Se procedió a correr la simulación y se observaron los siguientes resultados:

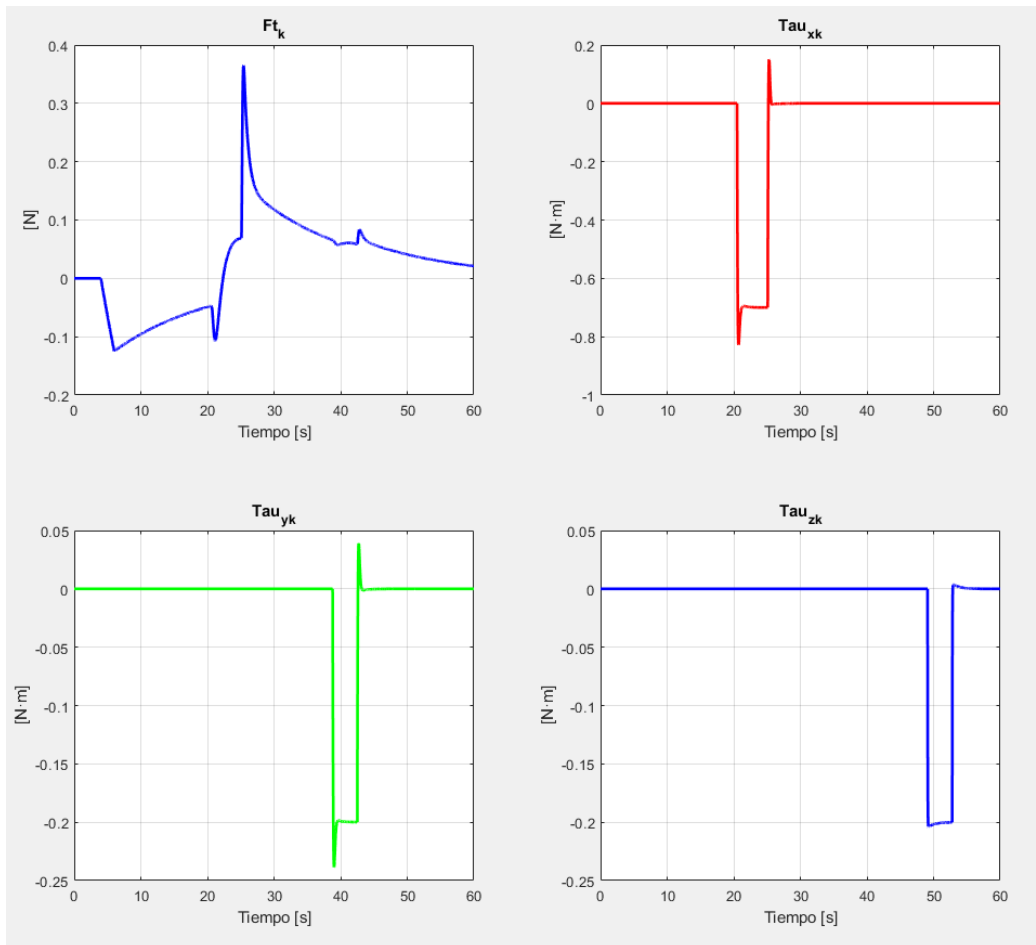


Figura N° 29: Señales de control.

Se observaron las entradas de control a los 4 motores y corresponden correctamente a las peticiones que se le hicieron al dron.

A continuación, se explicarán los resultados de cada input.

Se puede observar en la figura 30, que al solicitar un torque en X se produce un Roll. Este genera una inclinación en Theta, la cual produce que el dron tenga una velocidad en V sobre el eje Y. En el momento que el usuario regresa el torque solicitado a cero, el controlador LQR busca volver la velocidad generada en V a cero. Para ello el controlador comanda un Roll en sentido contrario al que se comandó anteriormente. Una vez que la velocidad V vuelve a ser cero, el dron se mantiene estable en hover.

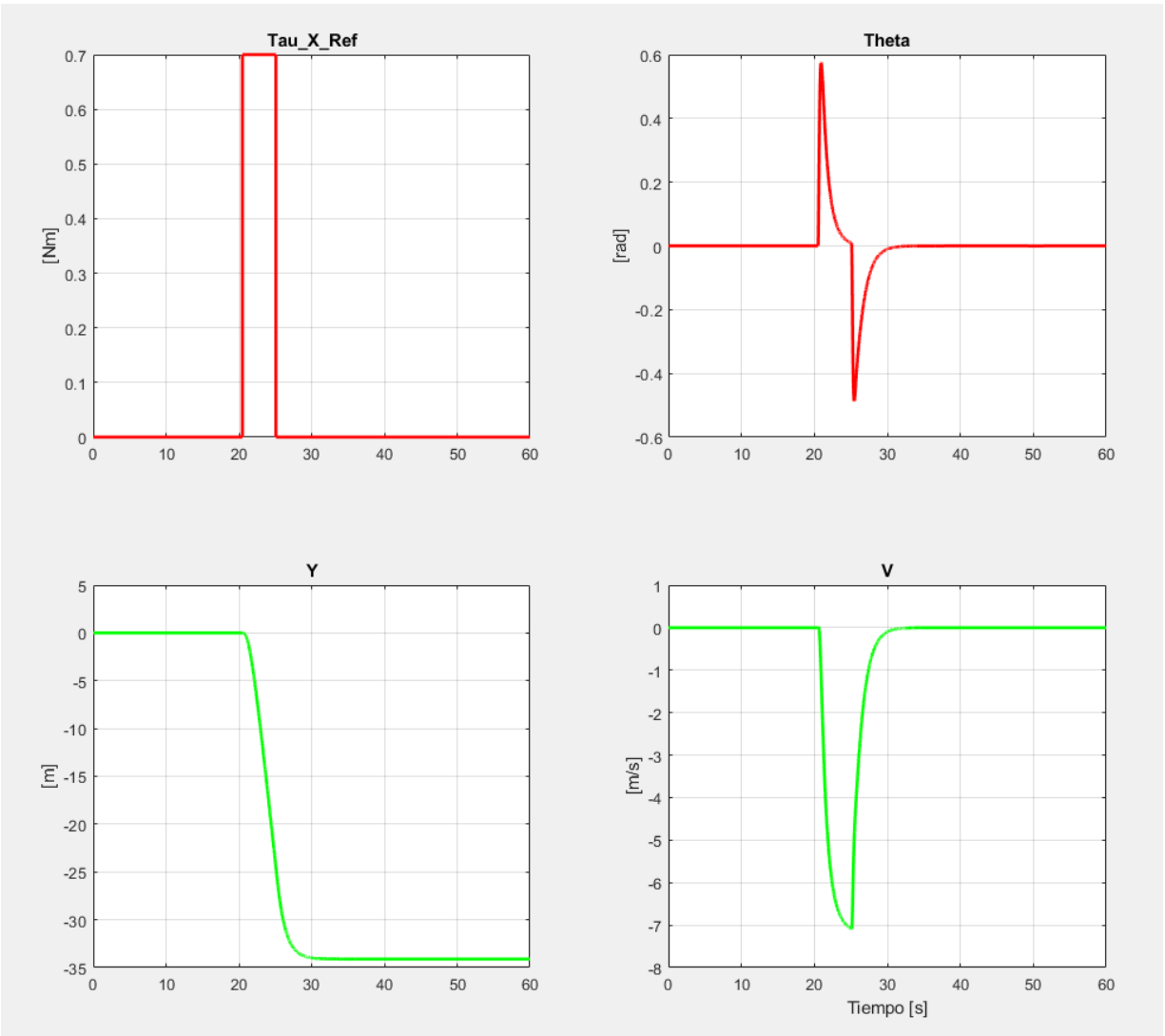


Figura N° 30: Resultados del input "ROLL".

Se puede observar en la figura 31, que al solicitar un torque sobre el eje X, se produce un Pitch. Este se puede observar por medio de la inclinación en Theta. Como resultado el dron obtiene una velocidad en U y un desplazamiento en el eje Y. Cuando el usuario deja de agregar un torque sobre el eje X, el controlador LQR intenta frenar la velocidad en Y por medio de una inclinación en el sentido contrario a la que se había producido sobre el eje X.

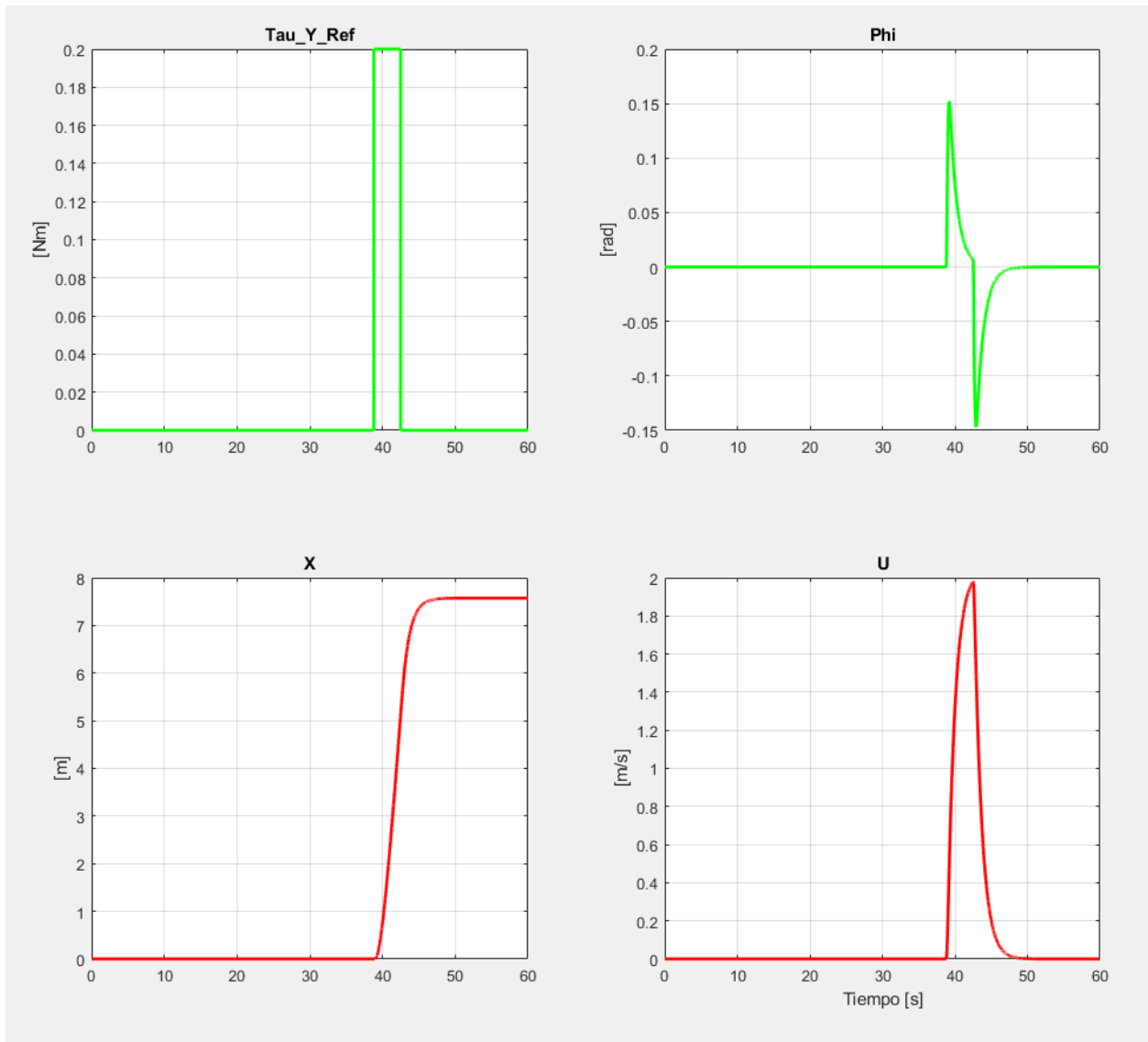


Figura N° 31: Resultados del input "PITCH"

Se observa en la figura 32, que al comandar un Yaw se produce una rotación en Rho y que, al volver a cero la referencia, el ángulo también vuelve a cero como se esperaba.

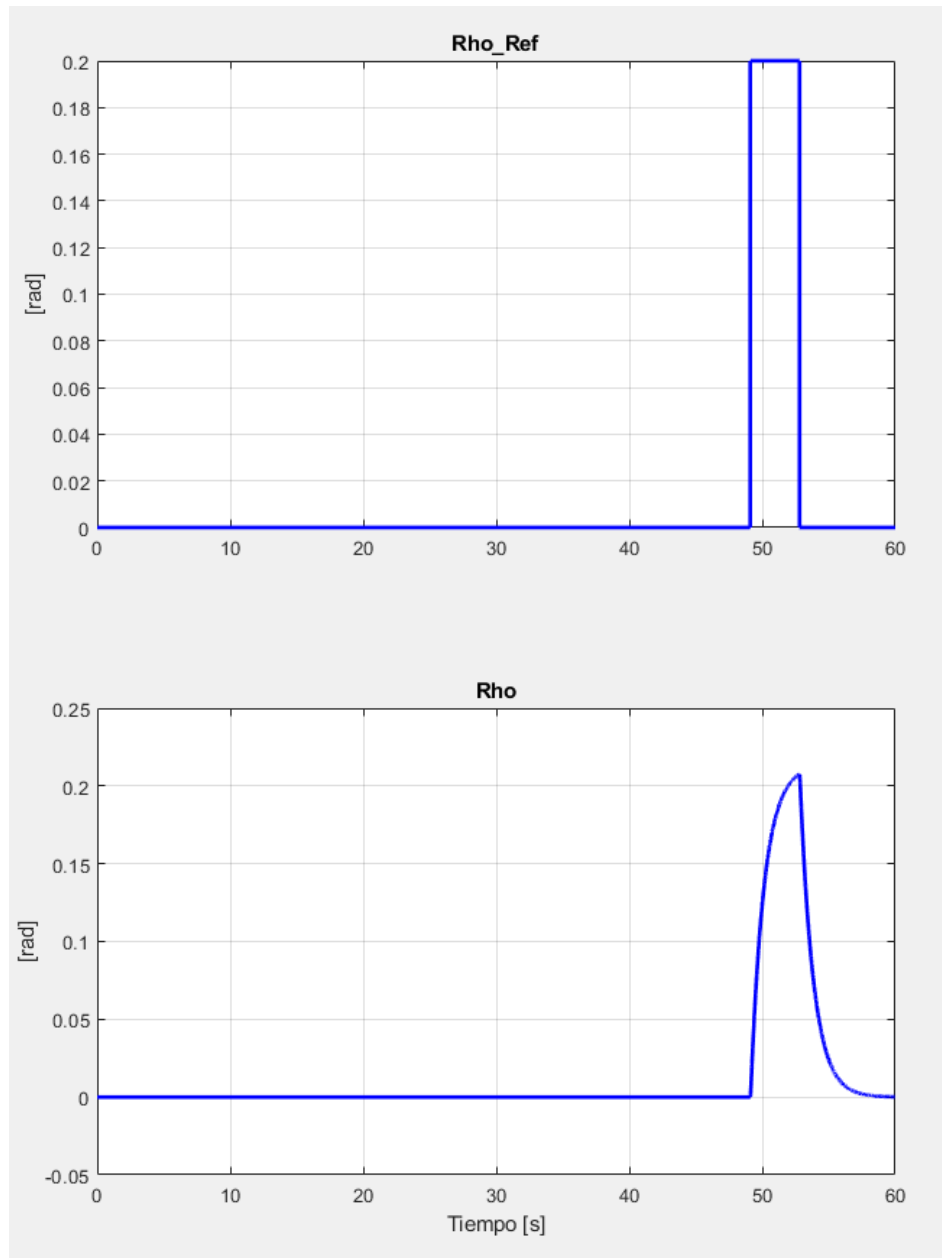


Figura N° 32: Resultados del input "Rho_Ref".

Se puede observar en la figura 33, que al solicitar un aumento del Thrust total en 1 N, el dron comienza a ascender. Al momento de volver el Thrust a hover, dejando el Ft_Ref en 0, el dron continúa elevándose, pero perdiendo velocidad. Esto sucede porque las rotaciones en los otros ángulos producen una disminución del empuje vertical. A pesar de que las fuerzas de los motores sumadas generen el equivalente al peso del dron, durante los Pitch y Roll, estas fuerzas dejan de estar alineadas con el peso del dron y, por tanto, se produce un descenso del dron.

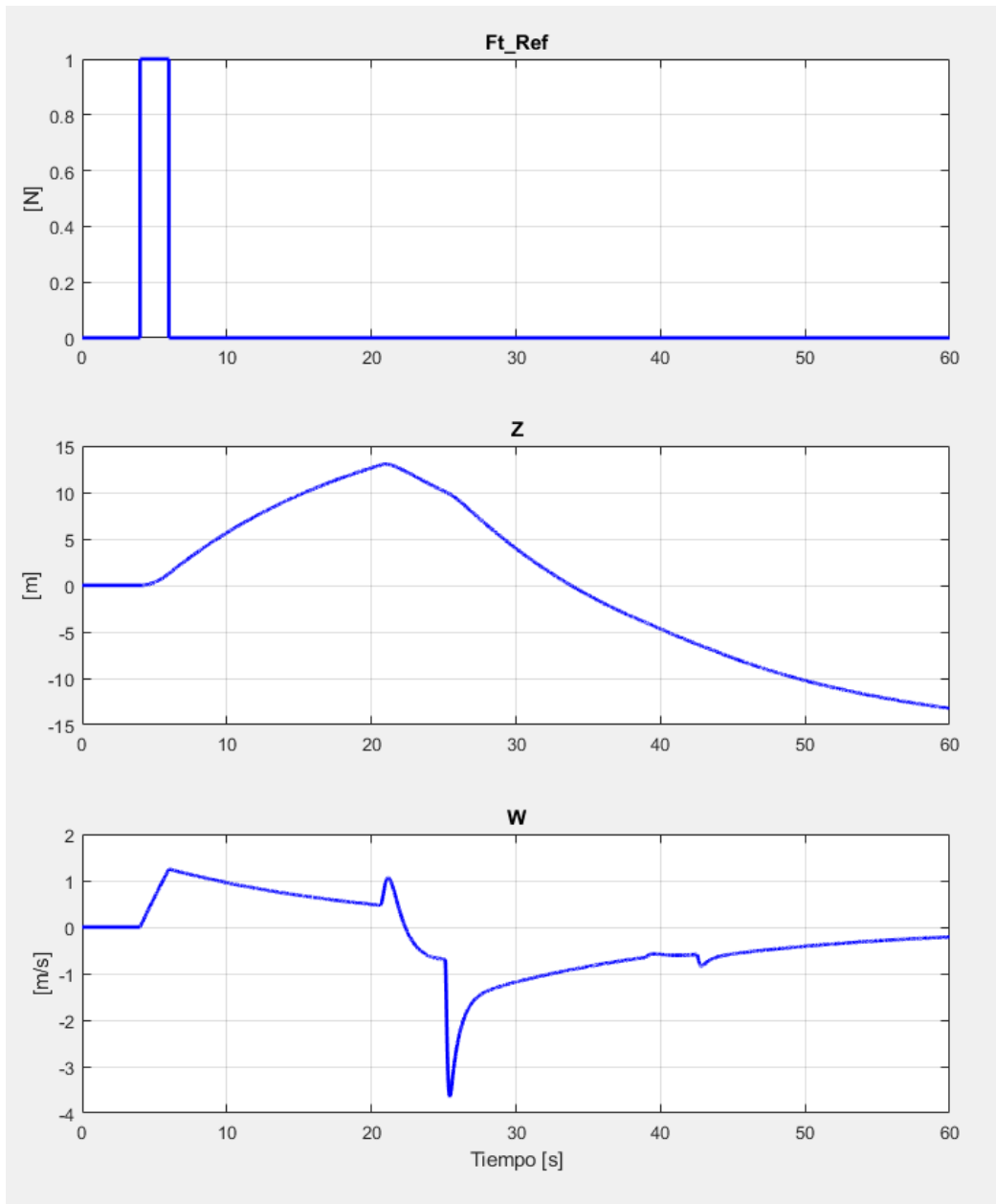


Figura N° 33: Resultados del input "Ft_Ref".

CAPITULO 5: IMPLEMENTACIÓN

En este capítulo se detalla la implementación física del prototipo, integrando el hardware seleccionado, la electrónica de control, los sensores y el software desarrollado.

El objetivo de esta etapa fue construir un dron real que materialice el diseño propio planteado en capítulos anteriores, manteniendo la premisa de que sea económico, modular y fabricado mediante impresión 3D.

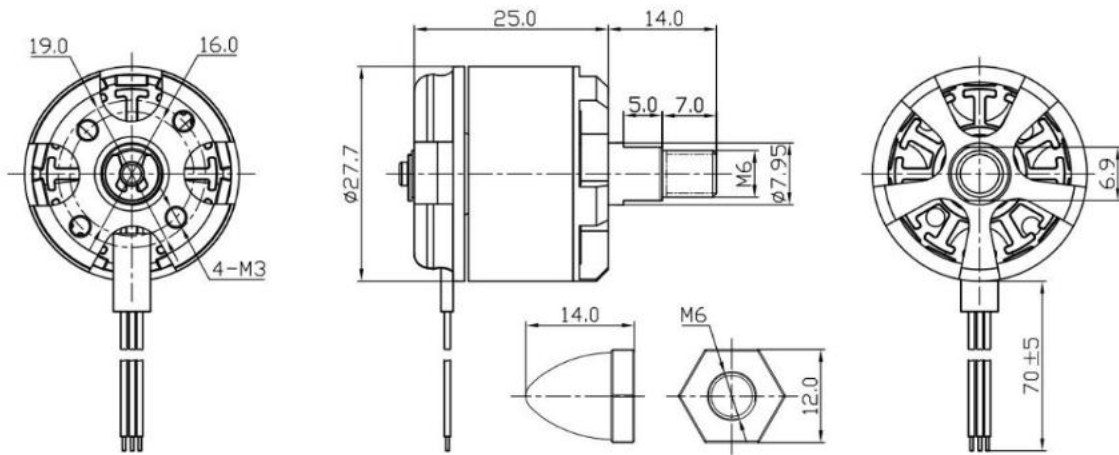
La implementación permitió validar de manera práctica las simulaciones previas y sentar la base para realizar pruebas de vuelo y futuras mejoras.

De los nueve estados a controlar, se optó por controlar a lazo cerrado los tres ángulos con sus respectivas velocidades angulares. Se eligió dejar la velocidad a lazo abierto debido a que es complejo sensorla. Inicialmente, se intentó estimarla mediante un integrador con el acelerómetro, pero el ruido presente en las mediciones de este provoca que la estimación de velocidad sea inestable y tenga un error significativo con respecto a la realidad en meros segundos de funcionamiento.

5.1 Hardware

Para la implementación física se seleccionó el siguiente hardware:

- Motor BLDC: Son los motores brushless que impulsaran al cuadricóptero, el modelo utilizado es el motor 2212 920KV que se muestra en la figura 34.



MOTOR PERFORMANCE DATA

MODEL	KV (rpm/V)	Voltage (V)	Prop	Load Current (A)	Pull (g)	Power (W)	Efficiency (g/W)	Lipo Cell	Weight (g) Approx
B2212	920	11.1	8045	7.3	465	81	5.7	2-4S	50
			1045	9.5	642	105	6.1		
	980	11.1	8045	8.1	535	90	5.9		
			1045	10.6	710	118	6.0		

Figura N° 34: Tabla técnica del motor B2212. Cortesía de [15].

- Microcontrolador: Se utilizó en kit de desarrollo ESP32 que cuenta, entre otras cosas, con un procesador de dos núcleos de 32 bits con una frecuencia de 240 MHz, conectividad inalámbrica por Wi-Fi y Bluetooth, 2 ADCs de 12 bits con hasta 18 canales. El mismo se muestra en la figura 35.



Figura N° 35: Placa de desarrollo Esp32.

- Unidad de medición inercial (IMU): Esta unidad contiene un giroscopio y un acelerómetro. La unidad utilizada es el MPU6050, que cuenta con ADCs de 16 bits y rangos de velocidades angulares de ± 250 a $2000^\circ/\text{seg}$ y de aceleraciones de ± 2 a $16g$.



Figura N° 36: Sensor MPU6050.

- Hélices: Las hélices utilizadas en los motores son las 1045, con un diámetro de 10 pulgadas y 4.5 pulgadas de paso en la hélice.
- Batería: La batería utilizada es una LiPo de 5000mAh con una clasificación C de 65.
- ESC: El controlador electrónico de velocidad elegido tiene una corriente máxima de 40 A. [16]



Figura N° 37: ESC del Dron.

A continuación, se muestran varias imágenes del prototipo físico:



Figura N° 38: *Dron ensamblado sin la batería.*



Figura N° 39: *Dron completo.*



Figura N° 40: *Dron con protectores de hélices.*

5.2 Medición del empuje generado por los motores

Para que el controlador funcione correctamente, se requiere conocer el empuje generado por los motores para un cierto valor de PWM ingresado. Para esto se fabricó un banco de pruebas utilizando una celda de carga y partes impresas en 3D.

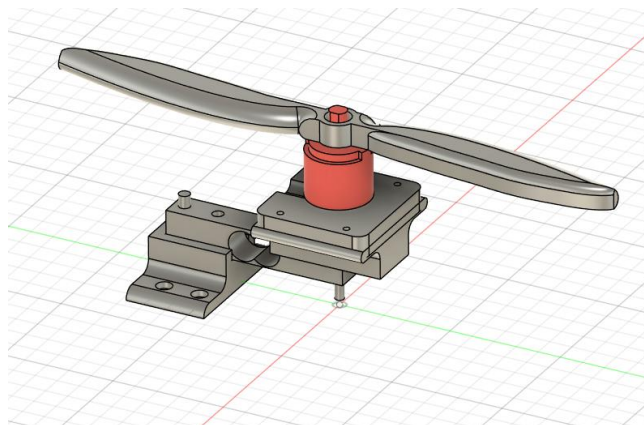


Figura N° 41: *CAD 3D del banco de motores.*

En la figura 42 se muestra el banco de pruebas utilizado para medir el empuje generado por los motores, este consiste en una celda de carga asegurada a una tabla, con una pieza plástica en la parte superior para acoplar el motor, la celda de carga es luego conectada a un microcontrolador como puede ser un Arduino o una ESP32 para realizar las mediciones deseadas.



Figura N° 42: Banco de prueba de motores.

Para realizar la medición, el motor es acoplado a la celda de carga. Luego, se procede a ingresar distintos valores PWM, en intervalos de 10 desde el valor mínimo 0 hasta el valor máximo 180 y formar una tabla con los resultados como se ve a continuación:

SERVO	Motor Thrust
0	0
10	22
20	33
30	52
40	73
50	95
60	119
70	154
80	178
90	204
100	247
110	296
120	354
130	408
140	445
150	503
160	555
170	610
180	663

Tabla N° 1: Valores de fuerza de empuje vs PWM.

Una vez obtenida la tabla de valores se procedió a interpolar cuadráticamente los datos, obteniendo la siguiente curva de empuje vs PWM. Esta ecuación cuadrática es la que usa el microcontrolador para determinar las señales que tiene que enviar a las ESC en base al empuje requerido por cada motor.

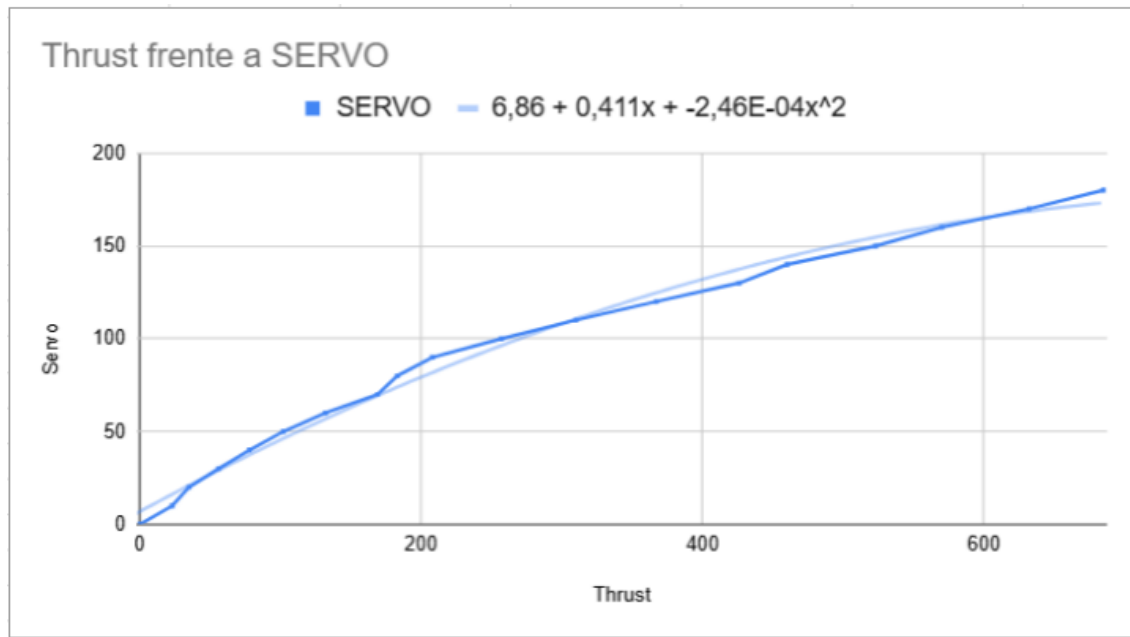


Figura N° 43: Interpolación cuadrática.

5.3 Diagrama de conexiones eléctricas

Las conexiones eléctricas en este prototipo son sencillas, cada motor es controlado por su respectivo ESC, estos son alimentados por la batería de 11.1V y reciben señales PWM de un GPIO del microcontrolador. El microcontrolador recibe datos de velocidad angular y aceleración del MPU6050, que es alimentado por la propia salida de 3.3V del ESP32.

Se optó por alimentar el microcontrolador con una batería independiente a la alimentación de los motores porque los altos consumos de estos provocaban ocasionalmente una pérdida de energía en el microcontrolador. En la figura 44 se muestra un diagrama con las conexiones:

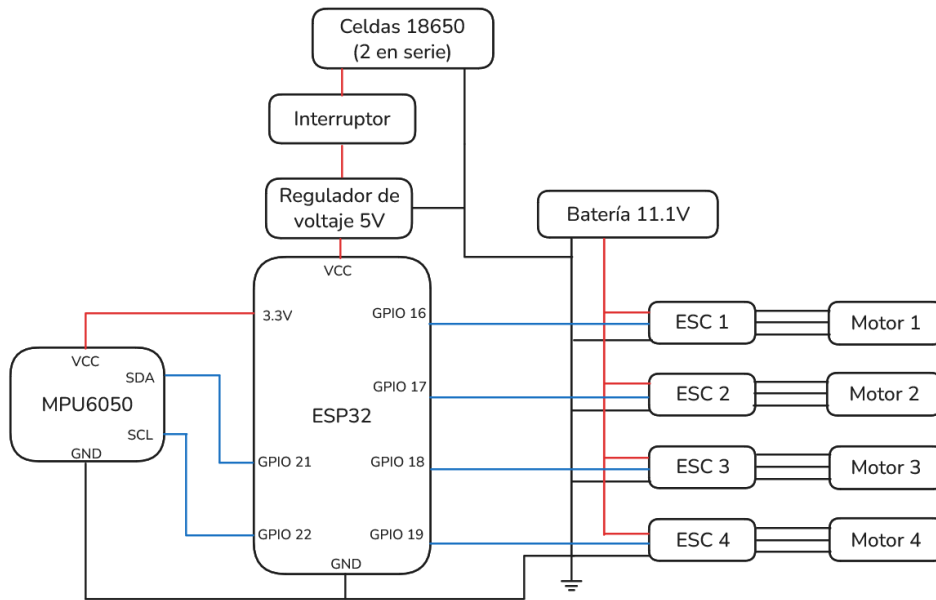


Figura N° 44: *Diagrama de conexiones eléctricas.*

5.4 Mando remoto

Para enviar referencias de control al dron se diseñó un mando remoto, el cual utiliza una ESP32. Esta se comunica con la ESP32 del dron mediante el protocolo de comunicación ESP-NOW [17].

Este protocolo permite que los dispositivos de Espressif Systems (la empresa que desarrolla los microcontroladores ESP32) se comuniquen de forma inalámbrica, ya sea mediante Bluetooth o WiFi.

De este modo se diseñó un mando en el software CAD3D Fusion360, el cual se ve en la figura 45. Si bien se estableció en el anteproyecto que se iba a realizar un circuito en una placa perforada para el mando, se determinó que debido al pequeño número de conexiones que se debían realizar, esto no sería necesario.

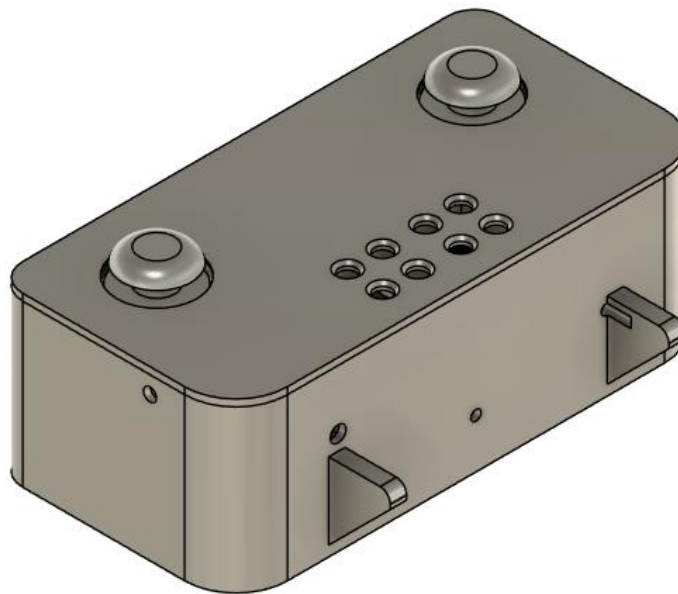


Figura N° 45: *Mando remoto.*

El mando posee dos joysticks analógicos que, a su vez tienen un pulsador cada uno y una botonera con ocho pulsadores.

Con los joysticks analógicos se leen las cuatro acciones de control que puede solicitar el usuario, y los botones son programables según lo necesario y lo que se añada al dron.

También dispone de 8 botones programables, entre los cuales se utiliza uno para armar y desarmar los motores para poder conectar y desconectar la batería sin la posibilidad de que se enciendan los motores.

El diagrama de conexiones eléctricas del mando remoto se muestra en la figura 46.

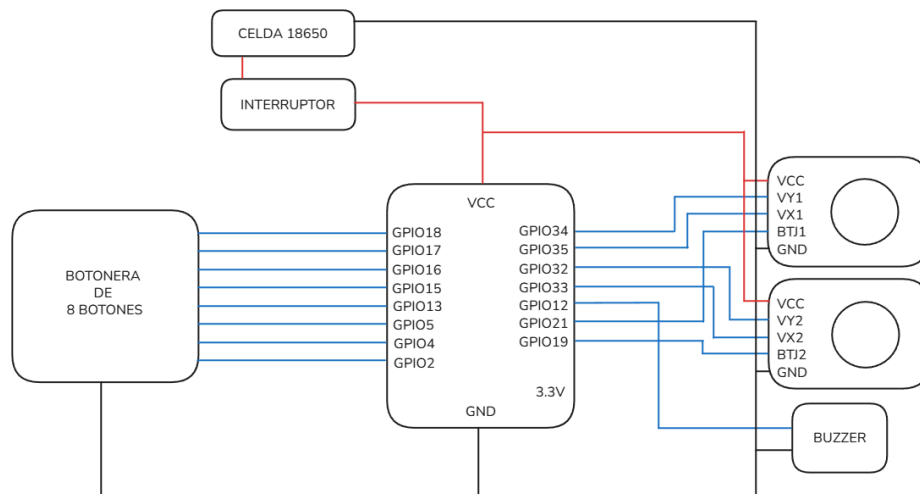


Figura N° 46: Diagrama de conexiones eléctricas del mando remoto.

5.5 Programación del microcontrolador

Un esquema de mando FPV se refiere a que las acciones Pitch, Roll y Yaw se realizan de forma directa, es decir que, si el usuario comanda un pitch hacia el frente, el dron se inclinara hacia delante sin importar su orientación, esto no es problemático cuando una cámara se encuentra montada al dron, ya que el movimiento será hacia delante desde el punto de vista de esta.

En este caso, el prototipo no tiene una cámara montada, por lo que el usuario mira al dron desde un punto fijo, esto es problemático, ya que si por ejemplo tiene un Yaw de 180°, un pitch hacia delante desde el punto de vista del dron, se vería como un pitch hacia atrás desde el punto de vista del usuario.

Debido a esto, se implementó un modo de conversión de FPV a Cartesiano. Para ello se leen las acciones que envía el mando remoto y se las transforma por medio de la matriz de rotación mostrada en la ecuación (2).

De esta forma, aunque el dron realice un Yaw, es decir, una rotación sobre su eje vertical, el usuario puede seguir asociando las acciones del mando con la dirección a la que se movía el dron previo al Yaw.

Se utilizó el software Arduino IDE para la escritura del código, la compilación y subida del programa al ESP32.

Para conseguir sensar los ángulos se empleó el filtro de Madgwick, que combina los datos del acelerómetro y giroscopio del IMU. Logra esto usando un descenso de gradiente para optimizar un cuaternión que orienta los datos del acelerómetro a una referencia de gravedad conocida. Este cuaternión se promedia con peso y se integra con el cuaternión del giroscopio y la orientación anterior. El resultado luego se normaliza y se convierte en ángulos de Euler. [9]

El filtro de Madgwick es un algoritmo de fusión sensorial diseñado para estimar la orientación tridimensional de un cuerpo rígido utilizando los datos del acelerómetro y el giroscopio. Su principal ventaja es la eficiencia computacional y la baja latencia, lo que lo hace ideal para sistemas embebidos como el ESP32. El algoritmo estima un cuaternión de orientación mediante un método de descenso de gradiente, corrigiendo las desviaciones del giroscopio con la información de gravedad proveniente del acelerómetro. A partir de este cuaternión se obtienen los ángulos de Euler (Roll, Pitch, Yaw) que representan la actitud del dron en tiempo real.

```

// Lectura de los sensores
now = millis();
dt = (now - lastTime) / 1000.0; // tiempo delta en segundos
lastTime = now;

// Leer aceleración y giro

mpu.getMotion6(&rawAx, &rawAy, &rawAz, &rawGx, &rawGy, &rawGz);

ax = rawAx / 16384.0;
ay = rawAy / 16384.0;
az = rawAz / 16384.0;

gx = rawGx / 131.0;
gy = rawGy / 131.0;
gz = rawGz / 131.0;

// Filtro de orientación
filter.updateIMU(gx, gy, gz, ax, ay, az);
roll = filter.getRoll();
pitch = filter.getPitch();
yaw = filter.getYaw() -180.0; // inicia en 180 se resta
                                // 180 como un offset

// Convertir a radianes
rollRad = roll * DEG_TO_RAD;
pitchRad = pitch * DEG_TO_RAD;
yawRad = yaw * DEG_TO_RAD;

```

Snippet N° 4: Código lectura y filtrado del sensor MPU6050.

Una vez obtenidos todos los datos de los sensores, se los introdujo en el controlador, obteniendo así los valores de fuerza total y torque en los 3 ejes.

```

// -K * X

ft_K = W;
tau_x_K = theta*4.6557 + P*1.0538 - V;
tau_y_K = phi*4.662 + Q*1.0568 + U;
tau_z_K = rho + R*1.0217;

// Restar los valores de -K * X con los de referencia

ft = ft_ref + ft_hover - ft_K;
tau_x = tau_x_ref - tau_x_K;
tau_y = tau_y_ref - tau_y_K;
tau_z = tau_z_ref - tau_z_K;

```

Snippet N° 5: Controlador LQR.

Luego se calcula la fuerza que tiene que realizar cada motor, a su vez se determina que motor tiene que ejercer la mayor fuerza, si esta excede la fuerza máxima que puede hacer el motor, se resta este exceso de todos los motores. Si bien esto provoca una pérdida de fuerza de empuje y por lo tanto altura, mantiene el torque solicitado. Esto se consideró necesario ya que estos casos se producirán cuando el dron obtenga una inclinación considerable y se determinó que conservar la estabilidad del prototipo es más importante que mantener su altura.

Por último, se insertan los datos obtenidos en una ecuación cubica producto de la interpolación en la figura 43, que convierte estos datos en instrucciones PWM para los cuatro motores. Nótese que la librería que se está utilizando es una librería orientada a servomotores, y por lo tanto utiliza valores de 0 a 180 siendo 180 un PWM con un ciclo de trabajo de 1.

```
// Actualizar valores de motores

ft_g = ( (ft)/(cos(pitchRad) * cos(rollRad)) ) * 1000 / 9.81;
// pasaje de fuerza en newtons a gramos
f_tau_x = tau_x * 1000 / (9.81 * 1); // pasaje de torques en Nm a gramos
f_tau_y = tau_y * 1000 / (9.81 * 1);
f_tau_z = tau_z * 1000 / (9.81 * d);

mot1 = (-f_tau_z + ft_g + f_tau_x - f_tau_y) / (4);
mot2 = (f_tau_z + ft_g - f_tau_x - f_tau_y) / (4);
mot3 = (-f_tau_z + ft_g - f_tau_x + f_tau_y) / (4);
mot4 = (f_tau_z + ft_g + f_tau_x + f_tau_y) / (4);

// Hallamos el valor maximo de los 4 motores

motmax = max(mot1, mot2);
motmax = max(motmax, mot3);
motmax = max(motmax, mot4);

if (motmax > 679)
{
    motmax = motmax - 679; // Diferencia de excedente
    mot1 = mot1 - motmax; // Restamos excedente de forma igual a todos
                        // los motores para no perder el torque deseado
    mot2 = mot2 - motmax;
    mot3 = mot3 - motmax;
    mot4 = mot4 - motmax;
}
/**/
mot1 = mot_a * (mot1 * mot1 * mot1) + mot_b * (mot1 * mot1) + mot_c * (mot1) + mot_d;
mot2 = mot_a * (mot2 * mot2 * mot2) + mot_b * (mot2 * mot2) + mot_c * (mot2) + mot_d;
mot3 = mot_a * (mot3 * mot3 * mot3) + mot_b * (mot3 * mot3) + mot_c * (mot3) + mot_d;
mot4 = mot_a * (mot4 * mot4 * mot4) + mot_b * (mot4 * mot4) + mot_c * (mot4) + mot_d;
```

Snippet N° 6: Código de cálculo de fuerzas y transformación a PWM.

El algoritmo de control se implementó en el microcontrolador ESP32 mediante una estructura cíclica dentro del bucle principal (loop) del programa. En lugar de utilizar interrupciones por hardware, se optó por un control temporal utilizando la función “millis()”, que mide el tiempo transcurrido desde el arranque del sistema.

En cada iteración del loop, se evalúa si el tiempo actual menos el tiempo de la última ejecución es igual o supera al período de muestreo definido. Si esta condición se cumple, se ejecuta un ciclo completo de control y se actualiza el tiempo de la ejecución del control para la siguiente comparación. De esta forma se logra una frecuencia de muestreo de 250 Hz, equivalente a un período de muestreo de 4 ms.

Dentro de este bloque controlado por tiempo se ejecutan las siguientes etapas:

1. Lectura de sensores: se obtienen los valores brutos de aceleración y velocidad angular del IMU MPU6050.
2. Filtrado de orientación: se aplican los datos al filtro de Madgwick para obtener los ángulos de Euler (roll, pitch, yaw).
3. Cálculo del control: se construye el vector de estados y se aplica la ley de control $u = -Kx$
4. Suma con acción del usuario: se agregan las referencias provenientes del joystick o mando remoto.
5. Conversión de torques a fuerzas: se calculan las fuerzas equivalentes para cada motor en función de $Ft, \tau_x, \tau_y, \tau_z$.
6. Traducción a señales PWM: las fuerzas se convierten en señales PWM mediante la ecuación polinómica interpolada previamente, y se envían a los ESC para accionar los motores.

Este procedimiento garantiza una ejecución periódica y estable del lazo de control, manteniendo la sincronización entre la lectura de sensores y la actualización de las salidas.

A continuación, se muestra un diagrama de flujo simplificado que resume el funcionamiento del algoritmo implementado:

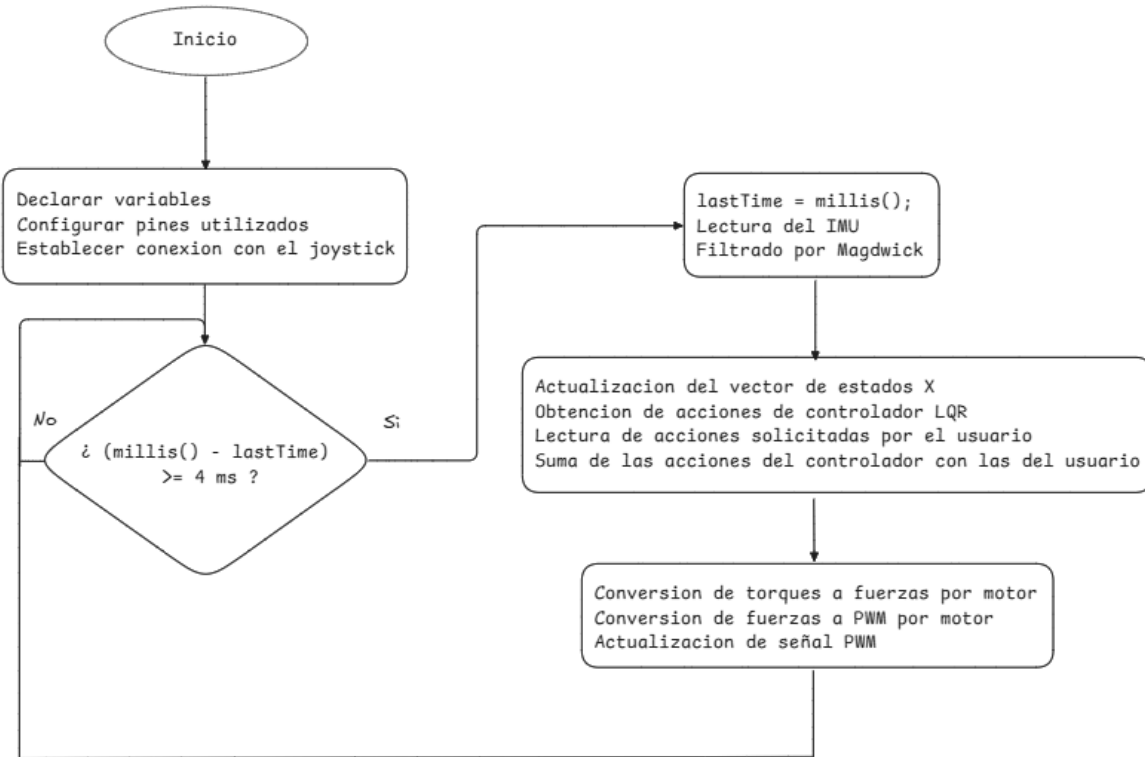


Figura N° 47: Diagrama de flujo del funcionamiento del dron.

5.6 Pruebas de vuelo

Las pruebas de vuelo fueron realizadas en el predio de futbol de la facultad. Se encendió el dron y se procedió a realizar la secuencia de arranque los motores. Una vez realizado esto, se lo alzo al vuelo.



Figura N° 48: Dron listo para despegar.

Durante el vuelo que duro unos minutos, se observó que el viento arrastraba al dron en la dirección de este. Esto sucedió por la ausencia de un controlador de posición y velocidad, por lo que se esperaba que esto sucediera. Así mismo el usuario pudo contrarrestar este arrastre con las señales enviadas desde el joystick.



Figura N° 49: *Dron en vuelo.*

Luego se procedió a aterrizarlo en el césped. El aterrizaje no fue perfecto debido a la poca experiencia del piloto y a la falta de un controlador de posiciones y velocidades lineales. Es por esto que no se logró mantener el dron en una posición fija y al momento del aterrizaje el dron tenía unas leves velocidades lineales. A pesar de esto, la construcción solida del dron no tuvo problema absorbiendo el impacto.



Figura N° 50: *Dron descendiendo.*

Una vez aterrizado se desconectó la batería y se comprobó que el prototipo no presentara daños finalizando así las pruebas de vuelo.



Figura N° 51: *Foto con el Dron.*

5.7 Análisis de costos y cronograma de ejecución

A continuación, se presenta una tabla con el costo en dólares de los componentes del dron, cabe destacar que no se incluyen los costos de mano de obra para ensamblar el prototipo ni el costo del uso de ninguna herramienta.

<i>Componente</i>	<i>Cantidad</i>	<i>Precio por unidad</i>	<i>Precio total</i>
<i>Kit motores 2212 920KV + 4 hélices 1045 + 4 ESC de 30A</i>	<i>1</i>	<i>95 USD</i>	<i>95 USD</i>
<i>MPU6050</i>	<i>1</i>	<i>3 USD</i>	<i>3 USD</i>
<i>ESP32</i>	<i>2</i>	<i>10 USD</i>	<i>20USD</i>
<i>Modulo Joystick</i>	<i>2</i>	<i>1.5 USD</i>	<i>3 USD</i>
<i>PLA x1kg</i>	<i>1</i>	<i>12 USD</i>	<i>12 USD</i>
<i>PETG x 1kg</i>	<i>1</i>	<i>12 USD</i>	<i>12 USD</i>
<i>Celda 18650</i>	<i>3</i>	<i>4 USD</i>	<i>12 USD</i>
<i>Batería 3S 5000 mAh 65C</i>	<i>1</i>	<i>76 USD</i>	<i>76 USD</i>
<i>Componentes varios (Interruptores, cables, botones, tornillos, etc.)</i>	<i>1</i>	<i>15 USD</i>	<i>15 USD</i>
		<i>TOTAL:</i>	<i>248 USD</i>

Tabla N°2: Análisis de costo del prototipo

Dentro de las alternativas comerciales de un precio similar, se encuentra el DJI Mini 2 SE [18], un dron compacto de uso recreativo, con estabilización electrónica, autonomía aproximada de 30 minutos y cámara integrada cuyo precio es de entre 350 y 400 USD.

En comparación con este modelo, el prototipo desarrollado en este proyecto presenta un costo considerablemente menor. Si bien carece de funciones avanzadas como posicionamiento GPS, cámaras o algoritmos de navegación autónoma, incorpora un sistema de control propio implementado en un microcontrolador ESP32 y un diseño modular impreso en 3D, lo que lo convierte en una plataforma accesible para fines didácticos, experimentales y de investigación.

Además, la fabricación mediante impresión 3D no solo reduce los costos, sino que facilita la reparación y reemplazo de piezas, permitiendo adaptar el chasis y los módulos según sea necesario.

CAPITULO 6: CONCLUSIÓN Y TRABAJO A FUTURO

El trabajo permitió cumplir el objetivo principal de diseñar e implementar un prototipo de cuadricóptero propio, económico, modular y fabricado mediante impresión 3D, orientado a fines didácticos y experimentales. El diseño CAD y la fabricación por impresión 3D posibilitaron reducir costos y facilitar la replicación del prototipo, asegurando su accesibilidad para pruebas y estudios en el ámbito académico.

Como parte del desarrollo, se realizó un modelado matemático del dron, a partir del cual se diseñó un controlador LQR, con el fin de estabilizar el sistema y validar su funcionamiento en vuelo. Este controlador se implementó en un microcontrolador ESP32 y se integró con un sensor IMU MPU6050, logrando estabilizar el dron ante perturbaciones externas y seguir referencias de manera satisfactoria.

Las pruebas realizadas en el simulador fueron exitosas y demostraron que el controlador es capaz de estabilizar el dron. A su vez, proveyeron la información necesaria para generar criterios de estabilidad deseados a la hora de implementar el cuadricóptero.

En la implementación, las pruebas experimentales confirmaron que el sistema es capaz de seguir referencias de ángulos con precisión y compensar perturbaciones externas de orientación, demostrando la efectividad del controlador implementado.

En conclusión, se alcanzó el diseño de un dron propio y económico, junto con la implementación de un sistema de control que garantiza su operación estable. A futuro, se propone avanzar en la estimación de velocidades lineales y en la incorporación de nuevos sensores que permitan ampliar las capacidades de navegación autónoma. Además, se propone montar un joystick que no regrese al centro en el eje vertical, de forma que permita mantener una referencia de hover fija mecánicamente y moverla en función de lo requerido.

GLOSARIO

Actitud	Conjunto de ángulos que describen la orientación del dron respecto al marco de referencia inercial, normalmente definidos por los ángulos de Euler: roll, pitch y yaw.
LQR (Linear Quadratic Regulator)	Controlador óptimo lineal que minimiza una función de costo cuadrática para estabilizar un sistema.
Linealización	Proceso de aproximar un sistema no lineal por uno lineal en torno a un punto de equilibrio.
IMU (Inertial Measurement Unit)	Unidad de medición inercial que combina acelerómetros y giróscopos para estimar la orientación y el movimiento.
ESC (Electronic Speed Controller)	Dispositivo que regula la velocidad de los motores brushless mediante señales PWM.
PWM (Pulse Width Modulation)	Técnica de modulación utilizada para controlar la potencia entregada a los motores variando el ciclo de trabajo de una señal.
Hover	Estado de vuelo en el cual el dron mantiene su posición y altitud sin desplazarse.
Cuaternión	Representación matemática de la orientación espacial que evita los problemas de singularidad de los ángulos de Euler.
Pitch	Inclinación del dron hacia adelante o atrás.
Roll	Inclinación lateral del dron.
Yaw	Giro del dron sobre su eje vertical.

REFERENCIAS BIBLIOGRÁFICAS

- [1] “Archived: Build Your Own Multicopter”, Ardupilot, consultado el 23 de oct 2025. Disponible en: <https://ardupilot.org/copter/docs/build-your-own-multicopter.html>
- [2] “ESP32 Flight Controller,” GitHub, consultado el 23 de oct 2025. Disponible en: <https://github.com/pratikPhadte/ESP32-Flight-controller->
- [3] Imagen extraída de un video de YouTube, “How to Make a Drone at Home,” consultado el 23 de oct 2025. Disponible en: <https://www.youtube.com/watch?v=dMRDzicSvXk>
- [4] Imagen extraída de un video de YouTube, “DIY Quadcopter Frame Design,” consultado el 23 de oct 2025. Disponible en: https://www.youtube.com/watch?v=V_mZsiZcy7s
- [5] DJI Agriculture, “DJI Agras T100,” consultado el 23 de oct 2025. Disponible en: <https://aq.dji.com/t100>
- [6] “What are Brushless DC Motors”, Renesas, consultado el 5 de oct 2025. Disponible en: <https://www.renesas.com/en/support/engineer-school/brushless-dc-motor-01-overview>
- [7] “La impresión 3D en FDM”, Academia Hellbot, consultado el 5 de oct 2025. Disponible en: <https://academia.hellbot.xyz/lecciones/la-impresion-3d-en-fdm/>
- [8] F. Sabatino, “Quadrotor control: modeling, nonlinear control design, and simulation,” Master’s thesis, KTH Royal Institute of Technology, Stockholm, Sweden, Jun. 2015, consultado el 5 de oct 2025. Disponible en: https://www.kth.se/polopoly_fs/1.588039.1600688317%21/Thesis%20KTH%20-%20Francesco%20Sabatino.pdf
- [9] Sebastian O. H. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” Ph.D. thesis, University of Bristol, Bristol, U.K., Apr. 2014, consultado el 5 de oct 2025. Disponible en: <https://x-io.co.uk/downloads/madgwick-phd-thesis.pdf>
- [10] K. Ogata, “Modern Control Engineering,” Quinta edición, Pearson, 2010.
- [11] Rahul H V, R Rachana, Vaishnavi L G, Nesara V, Savyasachi G K, Alfred Vivek D’Souza, 2022, “Electronic Speed Controllers: A Review,” INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICEI – 2022 (Volumen 10 – Edición 11).
- [12] “What is Degrees Of Freedom, 6DOF, 9DOF, 10DOF, 11DOF”, consultado el 5 de oct 2025. Disponible en: <https://web.archive.org/web/20160218110500/http://playground.arduino.cc/Main/WhatIsDegreesOfFreedom6DOF9DOF10DOF11DOF>
- [13] “What is Ground Effect,” consultado el 5 de oct 2025. Disponible en: <https://pilotinstitute.com/what-is-ground-effect/>

- [14] Sitio de desarrollo de Ardupilot, consultado el 5 de oct 2025. Disponible en: <https://ardupilot.org/dev/>
- [15] Racer.lt, "Racerstar Racing Edition 2212 BR2212 920KV 2–4S Brushless Motor for 350–380–400 Frame Kit," consultado el 5 de oct 2025. Disponible en: <https://www.racer.lt/item/racerstar-racing-edition-2212-br2212-920kv-2-4s-brushless-motor-for-350-380-400-frame-kit-3507.html>
- [16] ESC similar a la utilizada: Suppo Model, "40A Brushless Motor Speed Controller," consultado el 30 de oct 2025. Disponible en: <https://www.suppomodel.com/ESC/40A.html>.
- [17] Espressif Systems, "ESP-NOW: Low-Power, Connectionless Communication Solution," consultado el 5 de oct 2025. Disponible en: <https://www.espressif.com/en/solutions/low-power-solutions/esp-now>
- [18] DJI, "DJI 2 MINI SE", consultado el 18 de nov 2025. Disponible en: <https://www.dji.com/global/mini-2-se>
- [19] "Open Category — Low Risk — Civil Drones", consultado el 24 de nov 2025. Disponible en: <https://www.easa.europa.eu/en/domains/drones-air-mobility/operating-drone/open-category-low-risk-civil-drones>
- [20] R. Beard and T. McLain, "Quadrotor Dynamics and Control," Brigham Young University, 2012.

APÉNDICE

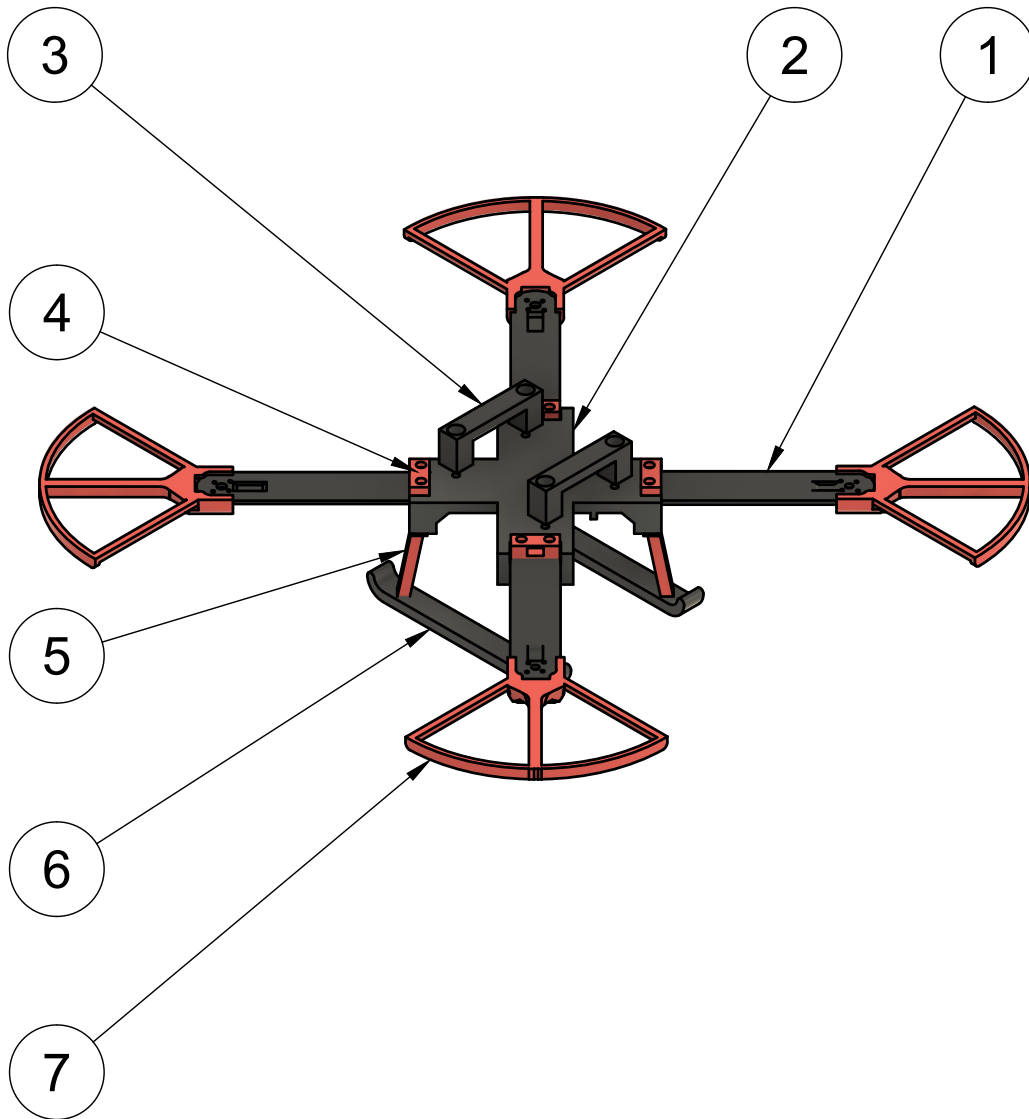
Datasheet ESP32: https://documentation.espressif.com/esp32_datasheet_en.pdf

Especificaciones del motor brushless 2212 920KV 2-4S: <https://www.hawks-work.com/pages/brushless-motor-2212>

Datasheet MPU-6050: <https://www.alldatasheet.es/html-pdf/517744/ETC1/MPU-6050/97/1/MPU-6050.html>

Manual ESC 40A: https://graysonhobby.com/pdf/grayson-40A_manual.pdf

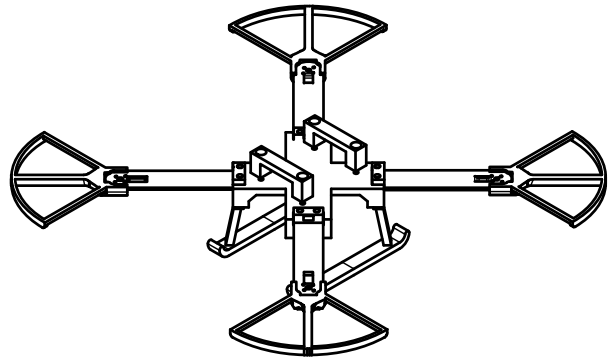
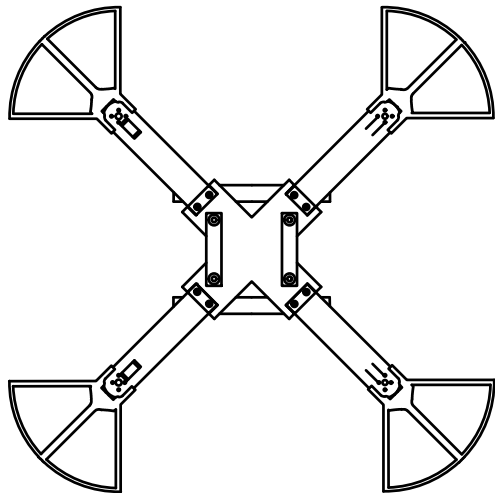
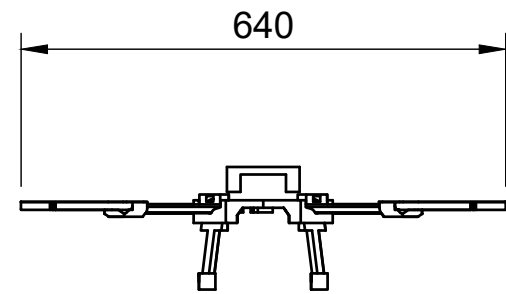
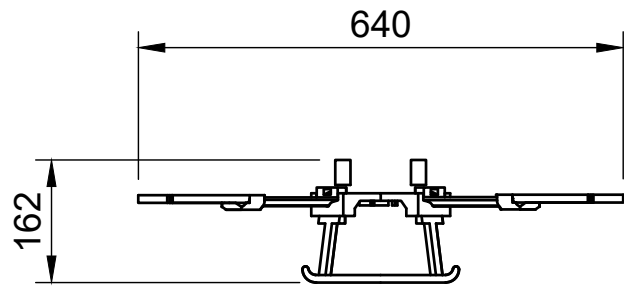
ANEXOS



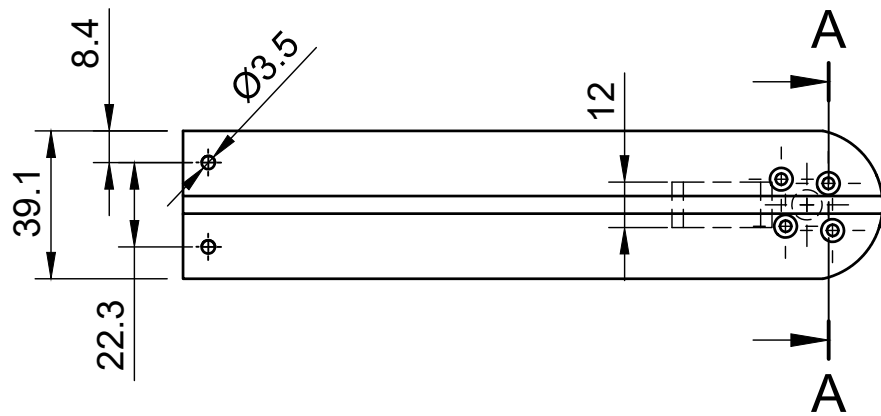
Lista de piezas

Elem ento	Canti dad	Número de pieza	Material
1	4	Brazo del motor	Plástico PET
2	1	Chasis	Plástico PLA
3	2	Abrazadera	Plástico PLA
4	4	Tope Brazo	Plástico PLA
5	4	Patas	Plástico PLA
6	2	Riel	Plástico PLA

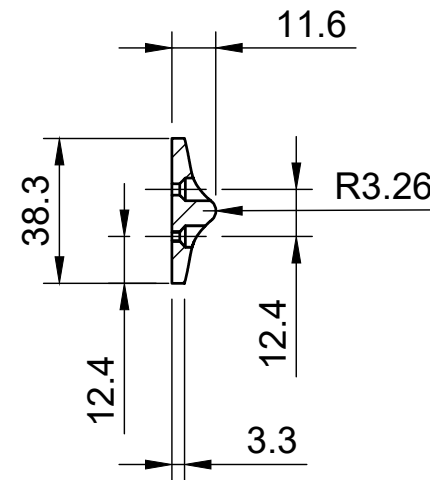
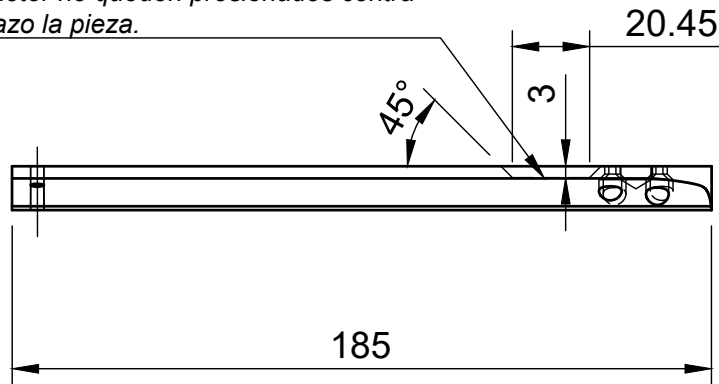
Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.00	
Vista de general de conjunto	Rev. 0	Fecha de emisión 3/11/2025	Escala 1:6
			Hoja 1/9



Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.01	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:10	Hoja 2/9
Vista general de conjunto			



Esta ranura se hizo para que los cables del motor no queden presionados contra el brazo la pieza.

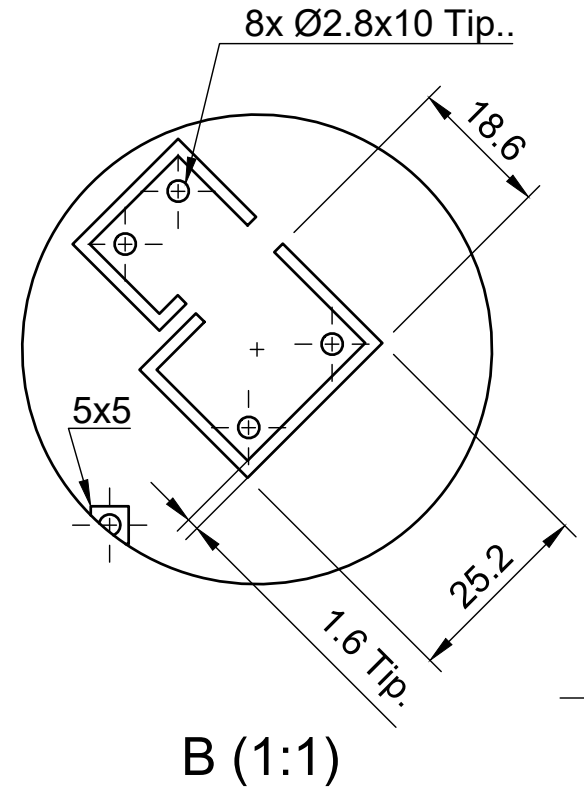
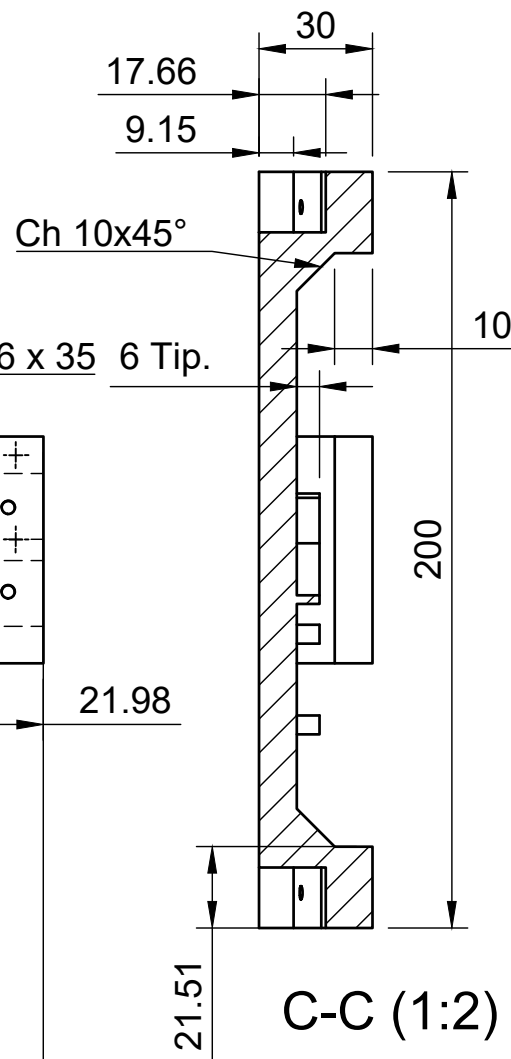
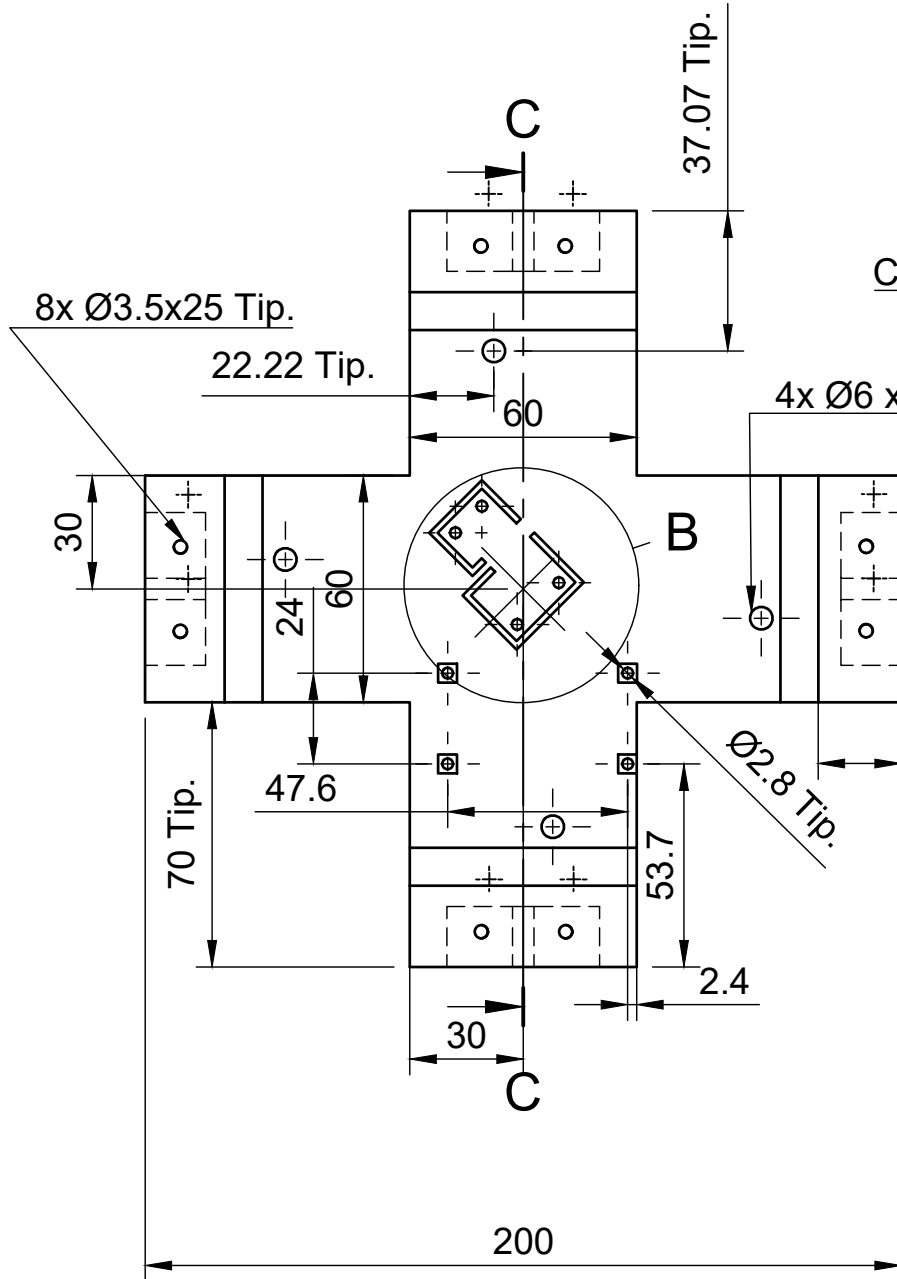


A-A (1:2)

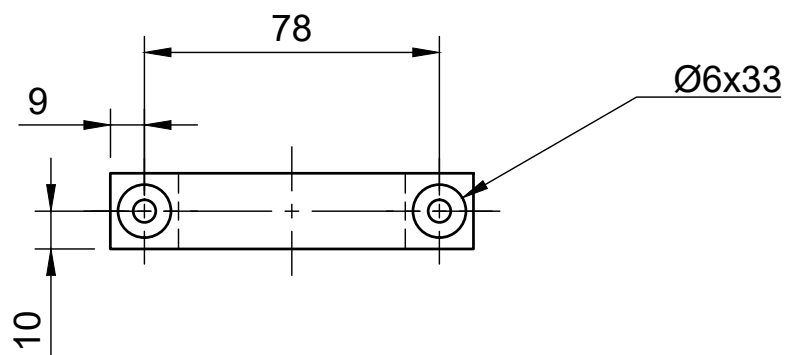
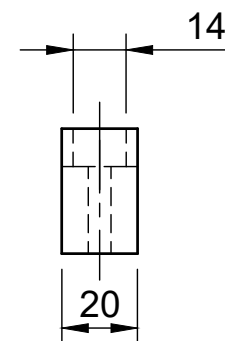
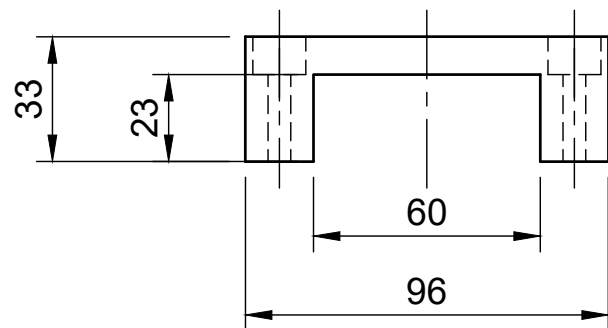
Nota:

No se detallan los agujeros para el montaje del motor ya que este varía según el modelo de motor a utilizar.

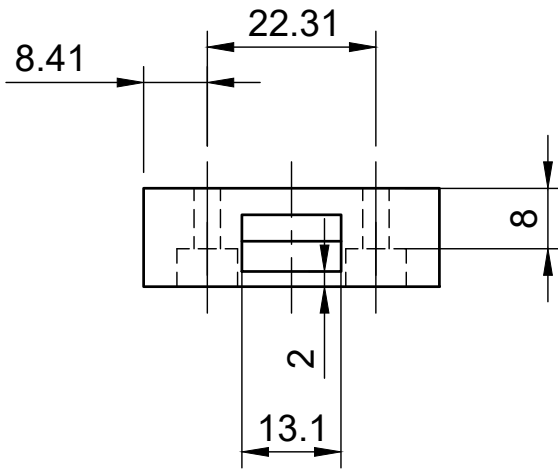
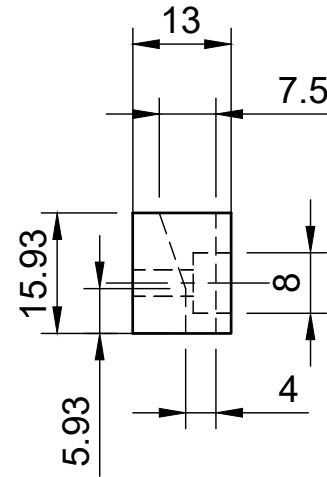
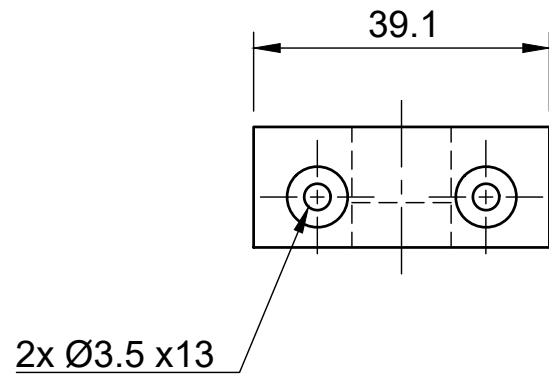
Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.02	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:2	Hoja 3/9
Brazo del motor 1			



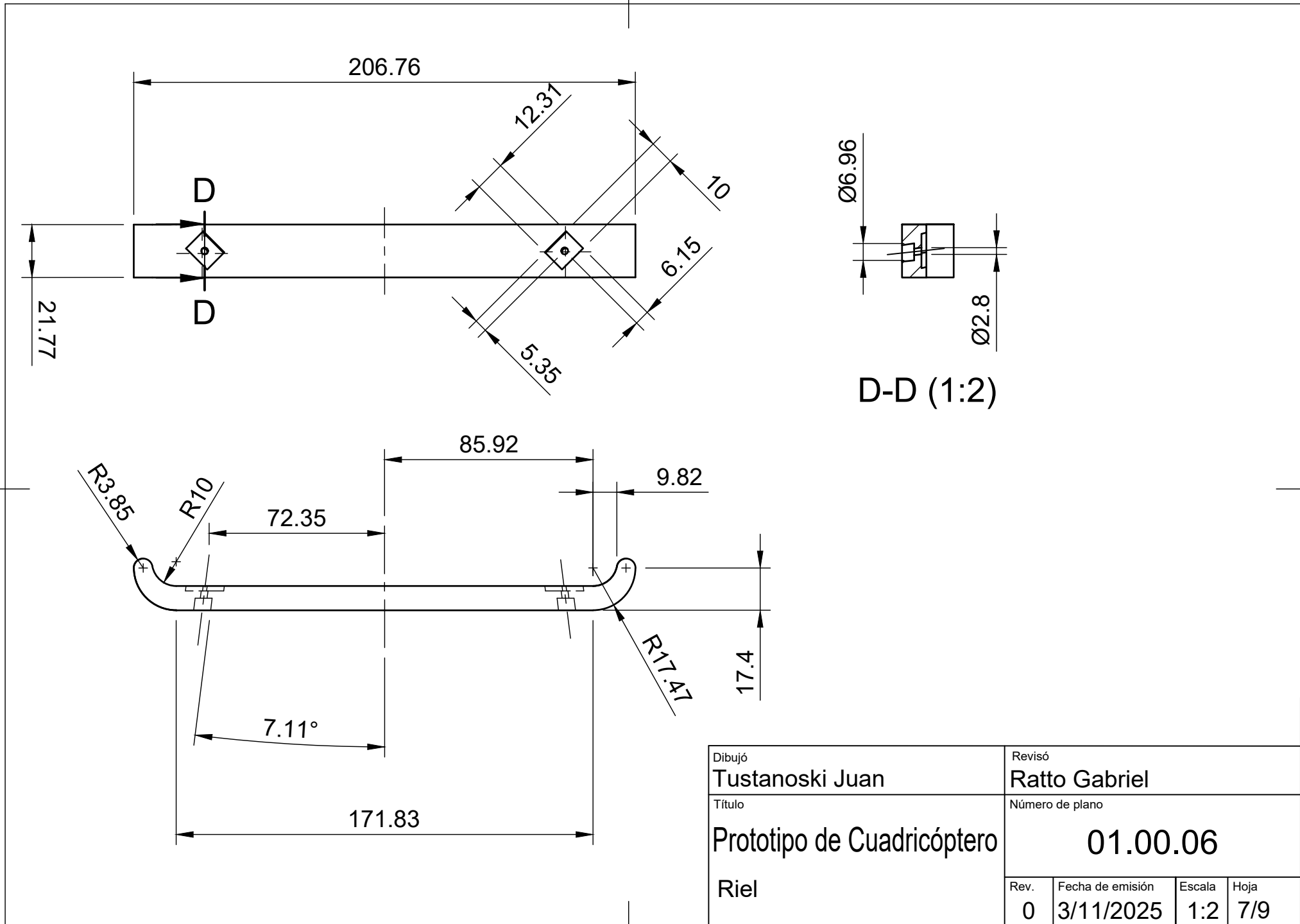
Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.03	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:2	Hoja 4/9
Chasis			

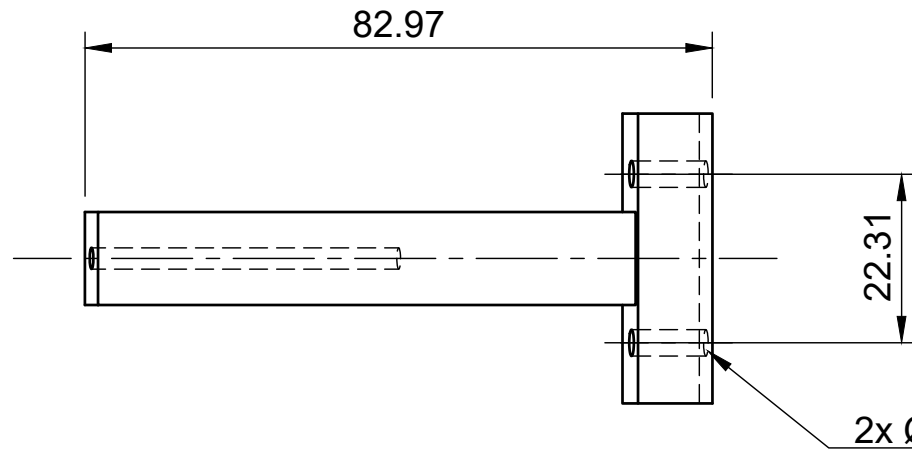


Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.04	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:2	Hoja 5/9
Abrazadera			

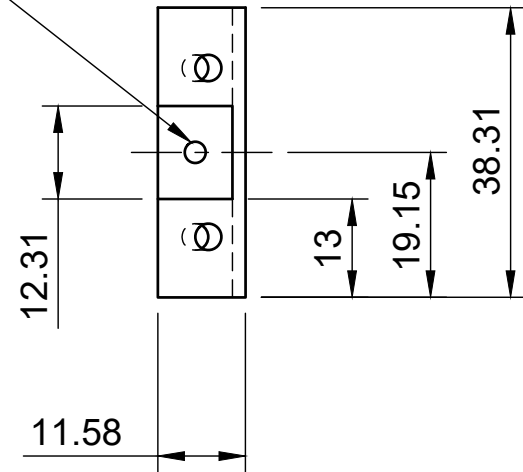


Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.05	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:1	Hoja 6/9
Tope Brazo			

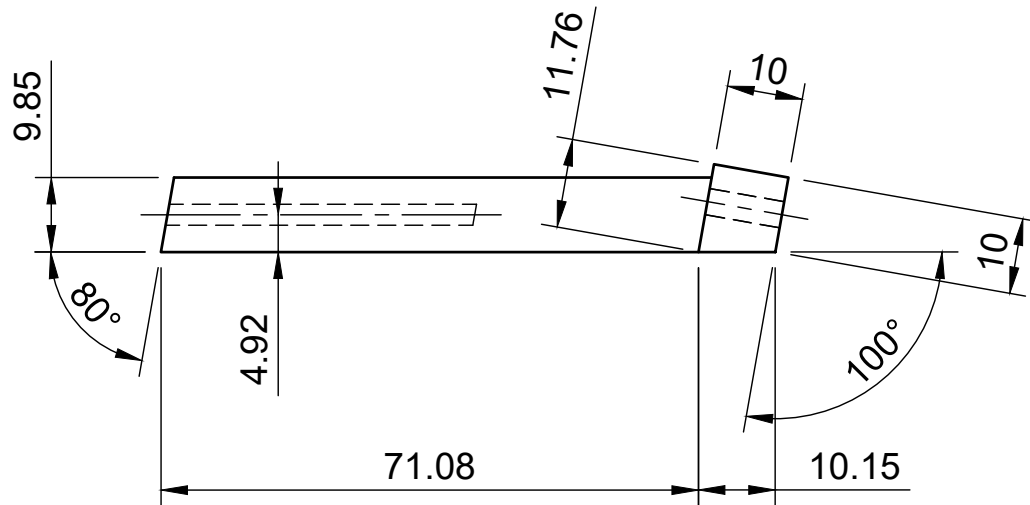




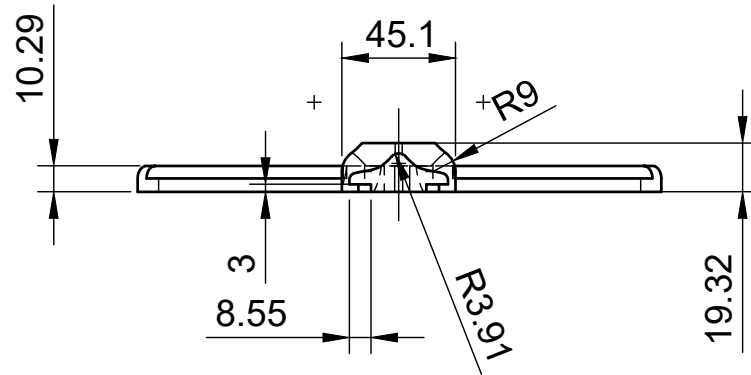
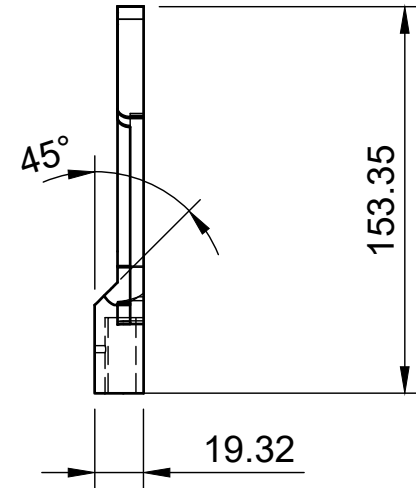
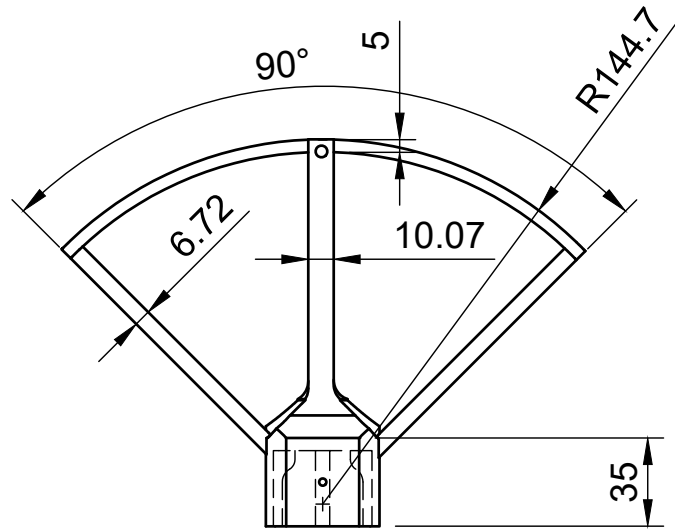
Ø2.8 x 40



2x Ø3.5 x 10



Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.07	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:1	Hoja 8/9
Patas			



Nota:

Se detallan solo las cotas más importantes debido a que hay múltiples radios y chaflanes que no son necesarios para la pieza. Estos quedaron después de múltiples pruebas y ya no tienen propósito. Se puede simplificar el diseño.

Dibujó Tustanoski Juan		Revisó Ratto Gabriel	
Título Prototipo de Cuadricóptero		Número de plano 01.00.08	
Rev. 0	Fecha de emisión 3/11/2025	Escala 1:3	Hoja 9/9
Protector			