



POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es



Francisco José Anguita Chamorro

05 TRABAJO FIN DE MASTER

INDUSTRIALES

TRABAJO FIN DE MASTER

ATERRIZAJE DE DRONES EN PLATAFORMAS MÓVILES. MODELADO Y CARACTERIZACIÓN

FEBRERO 2025

**Francisco José Anguita
Chamorro**

DIRECTOR DEL TRABAJO FIN DE MASTER:
**Pascual Campoy Cervera
Rafael Pérez Seguí**



POLITÉCNICA

AGRADECIMIENTOS

A mi tutor Pascual Campoy, por darme la oportunidad de realizar este trabajo en el seno del grupo que dirige.

A los chicos del CVAR, por la ayuda y motivación que me han prestado estos meses, y en especial a Rafael Pérez, que me ha acompañado y guiado durante todo este trayecto.

Y como siempre a mis padres, los pilares en los que me apoyo a cada paso que doy.

RESUMEN EJECUTIVO

Los Vehículos Aéreos no Tripulados (UAV), también llamados drones, son sistemas robóticos aéreos muy populares en la actualidad, usados en campos tan diversos inspección en industria, en cine y fotografía o en búsqueda y rescate. Su aplicación a problemas cada vez más complejos exige métodos de control más robustos que permitan un vuelo estable cuando el dron opera en las regiones más no lineales de su dinámica.

Con el fin de lograr un control que permita maniobras complejas como el aterrizaje en entornos hostiles se diseñan controladores más robustos capaces de absorber las perturbaciones que afectan al vuelo del dron. Estos controladores se basan en el modelado del dron para el cómputo de las instrucciones de control que mantienen al vehículo en el aire.

La obtención de un modelo más preciso y cercano al sistema real tiene un impacto positivo en el desempeño del controlador. Esta mejora en el modelado se consigue mediante la caracterización de los principales parámetros implicados en el modelo dinámico del dron.

Alcance

El objetivo de este trabajo es conseguir el aterrizaje autónomo de un dron en una plataforma móvil. Para conseguir un control de vuelo que permita esta maniobra se propone una metodología de caracterización del dron, seleccionando las características del dron a modelar y particularizando el método sobre un dron real y una plataforma de simulación que permitan realizar los experimentos que prueban y validan la metodología aplicada.

Se propone el aterrizaje autónomo sobre un barco como caso de uso. El barco es una plataforma móvil con la dificultad añadida del movimiento ascendente y descendente debido al oleaje. Los experimentos de aterrizaje en entorno marítimo y en plataforma real verifican el método que se propone en este trabajo.

Metodología

El desarrollo de la propuesta de metodología parte de un estudio del trabajo realizado en materia de caracterización y de un análisis del modelo dinámico del dron para seleccionar los parámetros más importantes y los métodos para su caracterización.

Con las técnicas elegidas se diseñan los experimentos con los que se calculan los parámetros seleccionados. Se separan los experimentos que sirven para mejorar la caracterización del dron real de los que sirven para el modelado del dron en simulación.

La caracterización se ha particularizado para el dron sobre el que se trabaja y sobre un simulador, de manera que se dispone de dos drones para la realización de experimentos de vuelo, uno real y uno simulado.

Los experimentos de vuelo permiten comparar el seguimiento de trayectorias de cada dron y analizar su similitud. Se preparan una zona de vuelo y un entorno de simulación marítima para realizar experimentos de aterrizaje que sirvan como verificación final del sistema propuesto.

Resultados y conclusiones

Los valores caracterizados del dron y su traslado a simulación han sido validados por los experimentos de seguimiento de trayectorias, de los que se han obtenido valores mínimos de error en los vuelos de los drones simulado y real. De aquí se concluye que los experimentos de cálculo de parámetros de la metodología propuesta son válidos.

Los resultados obtenidos en experimentos de aterrizaje muestran que se ha realizado con éxito la maniobra sobre distintas plataformas móviles en distintos entornos, verificando que la metodología de caracterización logra el aterrizaje autónomo del dron.

La contribución principal de este trabajo es una metodología completa de caracterización de un dron para conseguir su vuelo y aterrizaje autónomo, basándose en la implementación con el framework de robótica aérea Aerostack2 y el simulador Gazebo para realizar los experimentos que demuestran que la metodología es válida para el propósito que se propone.

Palabras Clave

UAV, cuadricóptero, caracterización, modelo dinámico, control de vuelo, aterrizaje autónomo

Códigos UNESCO

120326 SIMULACIÓN
120702 SISTEMAS DE CONTROL
330104 AERONAVES
330118 ESTABILIDAD Y CONTROL DE AERONAVES

ÍNDICE

AGRADECIMIENTOS	3
RESUMEN EJECUTIVO	5
ÍNDICE DE TABLAS	12
ÍNDICE DE FIGURAS	14
1. Introducción	15
1.1. Motivación	15
1.2. Objetivos	16
2. Estado del Arte	17
2.1. Control en el vuelo de drones multirrotor	17
2.2. Caracterización y modelado de drones	18
2.3. Vuelo de drones en simulación	19
3. Fundamento Teórico	21
3.1. Modelo cinemático y dinámico del dron quadrotor	21
3.2. Control de vuelo de drones	25
3.3. Componentes de un dron cuadricóptero	26
3.4. El framework Aerostack2	27
3.5. El simulador Gazebo	29
4. Caracterización y modelado del dron	31
4.1. Componentes del dron real	31
4.2. Obtención de los parámetros de caracterización	32
4.2.1. Dimensiones y peso	32
4.2.2. Cálculo de la inercia mediante péndulo bifilar	33
4.2.3. Obtención de curvas de empuje y parámetros del motor	34
4.3. Caracterización en simulación	34

4.3.1.	Modelado del dron en el simulador Gazebo	35
4.3.2.	Modelado del entorno marítimo	36
4.4.	Caracterización en el dron real	37
5.	Experimentos	40
5.1.	Experimentos con Péndulo Bifilar	40
5.2.	Experimentos con 'thrust stand'	41
5.3.	Pruebas de verificación del modelo	42
5.4.	Pruebas de aterrizaje en simulación	43
5.5.	Vuelos en real	44
6.	Resultados	45
6.1.	Resultados obtenidos en experimentos para inercia	45
6.2.	Resultados obtenidos en experimentos para curva de 'thrust'	46
6.3.	Comparación de vuelo de dron real y en simulación	48
6.4.	Resultados de aterrizaje en simulación	49
6.4.1.	Ajuste del algoritmo en entorno simple	49
6.4.2.	Aterrizaje en entorno marítimo simulado	51
6.5.	Resultados de aterrizaje real	52
7.	Discusión	53
8.	Conclusión	55
8.1.	Conclusiones	55
8.2.	Trabajo Futuro	55
ANEXOS		59
ANEXOS		60
A.	Planificación y Presupuesto	60
A.1.	Planificación	60
A.2.	Presupuesto	62

B.	Impacto social, impacto medioambiental y Agenda 2030	63
B.1.	Impacto social, aspectos éticos y legales	63
B.2.	Impacto medioambiental	63
B.3.	Desarrollo sostenible y Agenda 2030	64

ÍNDICE DE TABLAS

4.1. Lista de componentes del dron	31
4.2. Masa y dimensiones del frame 'ZMR250' del dron, medidas sobre el modelo de Blender	33
6.1. Resultados obtenidos en los experimentos de inercia en el eje Z	46
6.2. Resultados obtenidos en los experimentos de inercia en el eje X	46
6.3. Resultados obtenidos en los experimentos de inercia en el eje Y	46
6.4. Coeficientes de ajuste del mapa de 'thrust' para la interfaz de plataforma	47
6.5. Coeficientes aerodinámicos y parámetros obtenidos de los motores	48
6.6. Error cuadrático medio entre las trayectorias del dron real y del simulado en los experimentos de seguimiento y comparación	49
A.1. Costes de los recursos humanos empleados en el proyecto	62
A.2. Costes de los equipos empleados en el proyecto	62
A.3. Costes total del desarrollo del proyecto	62

ÍNDICE DE FIGURAS

3.1. Esquema de un cuadricóptero con variables de estado y principales fuerzas de los motores	21
3.2. Esquema de un motor con las principales fuerzas que produce	23
3.3. Esquema de la arquitectura del framework de robótica aérea Aerostack2 [1]	28
4.1. Vista superior del dron real que se caracteriza	31
4.2. Medidas de distancia del centro del dron a cada motor y de ángulo formado por los brazos del dron tomadas sobre el modelo en Blender	32
4.3. Esquema del experimento de péndulo bifilar definido por la longitud de las cuerdas, L , y la separación entre ellas, d	33
4.4. Ejemplo de curva de thrust en la que se aprecia la relación entre señal de throttle y thrust generado	34
4.5. Arquitectura del sistema de control de vuelo del dron basado en Aerostack2 en el simulador Gazebo	35
4.6. Dron modelado en el simulador Gazebo a partir del frame 'ZMR250'	36
4.7. Modelo de barco WAM-V usado para los experimentos de aterrizaje en entorno marítimo	37
4.8. Entorno marítimo simulado en Gazebo con barco y plataforma de despegue para el dron	37
4.9. Arquitectura del sistema de control de vuelo del dron real basado en Aerostack2	38
4.10. Curva de ajuste de Roll, Pitch y Yaw en controladora de Betaflight. El punto central se corresponde con un valor de 1500 de throttle, y los valores de 1000 y 2000 son las velocidades máximas en cada sentido que se pueden enviar a la controladora	39
5.1. Montaje de péndulo bifilar para el cálculo de las inercias respecto a a) Eje X, b) Eje Y, c) Eje Z	40
5.2. Montaje del thrust stand Series 1585 de TYTO Robotics	41
5.3. Software de captura de datos RCBenchmark para los experimentos con thrust stand	42
5.4. Experimento de seguimiento de trayectoria replicado en drones real (superior izquierda) y simulado (superior derecha) con la misma referencia (inferior)	42
5.5. Experimento de aterrizaje en simulación con trayectoria grabada de plataforma móvil real	43
5.6. Experimento de aterrizaje en barco en entorno marítimo simulado	44

5.7. Experimento de aterrizaje real en plataforma móvil en la zona de vuelo del CAR Robotics Arena	44
6.1. Representación de datos obtenidos de orientación del dron en a) Yaw, b) Roll, c) Pitch	45
6.2. Puntos de máximo obtenidos para un experimento en el eje Z	45
6.3. Datos recogidos en los experimentos de thrust stand para el cálculo de curvas de thrust	46
6.4. Superficie generada por curvas de thrust a distintos niveles de batería	47
6.5. Trayectorias seguidas por el dron real (azul) y el dron simulado (rojo)	48
6.6. Comparación de las trayectorias seguidas por los drones real y simulado en a) plano XY, b) altura	49
6.7. Trayectoria 3D seguida por el dron simulado (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en plataforma móvil simulada (naranja)	50
6.8. Evolución de la distancia entre el dron simulado y la plataforma móvil a lo largo de la simulación	50
6.9. Trayectoria 3D seguida por el dron simulado (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en barco (naranja) en simulación	51
6.10. Evolución de las distancias entre el dron y el barco a lo largo de la simulación	51
6.11. Trayectoria 3D seguida por el dron real (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en plataforma móvil real (naranja)	52
6.12. Evolución de las distancias entre el dron real y la plataforma móvil a lo largo del experimento	52
A.1. Diagrama EDP de la organización de las distintas fases del trabajo	60
A.2. Diagrama GANTT con las tareas realizadas, fechas de inicio y fin de cada una y tiempo empleado en ellas	61

1. Introducción

Los Vehículos Aéreos no Tripulados (UAV), también llamados drones, han sido introducidos en multitud de sectores dada su versatilidad y capacidad de vuelo autónomo. De entre ellos, los multicopteros destacan por la sencillez de su diseño y operación y por su capacidad de realizar despegue y aterrizaje vertical (VTOL). De entre todas las configuraciones de multicoptero, la más usada es el cuadricóptero, que cuenta con cuatro propulsores dispuestos de manera simétrica.

Los avances en computación que permiten la instalación de computadoras de a bordo cada vez más potentes han impulsado la aplicación de drones multicopteros en inspección, exploración, búsqueda y rescate, agricultura, etc ??.

El vuelo de drones para misiones cada vez más específicas dificulta las maniobras que deben ser capaces de realizar: alta velocidad, paso por espacios reducidos, aterrizaje en plataforma móvil o entornos desconocidos, operación en condiciones duras (niebla, viento), etc. Cerrar el bucle de control a nivel lo más bajo posible es importante para lograr un control ágil que permita abrir el abanico de movimientos del dron. En el control a bajo nivel intervienen parámetros físicos y aerodinámicos característicos del dron que deben ser calculados para su inclusión en el modelo.

El creciente interés en la aplicación de drones también ha motivado el esfuerzo científico en autonomía de drones, diseñando nuevos algoritmos de control que puedan hacer frente a los nuevos problemas. Los controladores del estado del arte son controladores basados en modelo que operan a niveles bajos de control, acentuando la necesidad de la obtención de un buen modelo del cuadricóptero.

En la investigación de nuevos algoritmos juega un papel de vital importancia la simulación de drones. La simulación permite llevar a cabo experimentos de forma muy rápida y sin ningún riesgo, lo cual es clave al hacer pruebas en entornos complejos y desconocidos. La simulación depende totalmente del modelo simulado del dron, que debe ser lo más fiel posible al real para que los resultados en simulación sean válidos.

1.1. Motivación

Las maniobras de despegue y sobre todo de aterrizaje son las más importantes que debe realizar un dron, y también las más peligrosas. La mayoría de daños en drones se producen al realizar el aterrizaje, donde una mala planificación o un fallo en el control puede resultar en una colisión que en el peor caso puede inutilizar por completo el dron. Este riesgo aumenta aún más si las condiciones del entorno son desfavorables o si el punto de aterrizaje es móvil.

En los entornos que rodean a un barco son comunes también los vientos que pueden interferir en el vuelo del dron. Además los drones que operan en entornos marítimos a menudo dependen del barco para su recarga, volviéndose vital la capacidad de aterrizaje seguro sobre ellos.

Los controladores actuales usados para resolver el control y seguimiento de trayectorias para aterrizaje se basan en el modelo del dron para calcular las acciones de control. Además, para un controlador dado, se puede mejorar la respuesta final del sistema afinando la planta que en este caso es el modelo del dron.

El proceso de diseño de un algoritmo de aterrizaje se ve muy beneficiado de una simulación que modele de forma realista el vuelo del dron para realizar experimentos rápidos con los que ajustar todos los parámetros involucrados en el algoritmo además de los parámetros del controlador del

dron sin riesgo de dañar el hardware.

Para obtener un modelo más fiel al real se calculan y caracterizan los parámetros más determinantes en el modelo y el funcionamiento del dron como su matriz de inercia o los coeficientes aerodinámicos de sus motores.

Con el dron caracterizado y la simulación se pueden realizar experimentos que validen la fidelidad con la que el dron modelado reproduce el vuelo del dron real.

1.2. Objetivos

El objetivo principal de este trabajo es conseguir el aterrizaje autónomo de un dron cuadricóptero sobre una plataforma móvil. Para lograrlo, se propone una metodología de caracterización y modelado del dron que permita conseguir un modelo fiel del dron en simulación. Sobre la simulación se diseña el algoritmo de aterrizaje cuyo ajuste se realiza sin comprometer al dron real. Esta simulación sirve también para verificar la fidelidad del vuelo simulado, validando a su vez la metodología de caracterización propuesta. El algoritmo de aterrizaje se prueba en simulación en un entorno marítimo y en real usando el dron y una plataforma que se desplaza sobre ruedas.

Se plantean los siguientes objetivos de alcance reducido que conducen a la consecución del objetivo global:

- CARACTERIZACIÓN DEL DRON
 - Realizar un estudio del estado del arte en caracterización, para conocer los parámetros más importantes en el modelo y diseño de un dron. A partir de este conocimiento, se elabora una lista de los parámetros a modelar.
 - Analizar las técnicas más usadas para la obtención de los parámetros seleccionados, a fin de escoger los métodos más apropiados para desarrollarlos.
 - Diseñar y llevar a cabo los experimentos con los que se calcularán los parámetros que caracterizan el modelo.
- MODELADO EN SIMULACIÓN Y REAL
 - Modelar el dron en el simulador, consiguiendo un modelo capaz de llevar a cabo misiones en el entorno de simulación elegido.
 - Modelar el entorno marítimo con oleaje y viento con el que probar el aterrizaje en simulación.
 - Aplicar al dron real los resultados de la caracterización que mejoren el control de su vuelo autónomo.
 - Ajustar el controlador del dron aprovechando la simulación de vuelo.
 - Verificación del paso de simulación a real mediante la comparación de experimentos de vuelo reales y su réplica en simulación.
- ATERRIZAJE AUTÓNOMO EN PLATAFORMA MÓVIL
 - Diseñar un algoritmo de aterrizaje en plataforma móvil basado en Aerostack2 haciendo uso del modelo caracterizado.
 - Probar la robustez del aterrizaje mediante experimentos en entorno marítimo.
 - Validar el aterrizaje y la caracterización mediante experimentos reales de aterrizaje en plataforma móvil.

2. Estado del Arte

En este capítulo se realiza un análisis de las publicaciones más recientes y relevantes en los campos de aterrizaje autónomo de drones y de caracterización de los mismos. De entre las técnicas expuestas a continuación se han seleccionado las más convenientes para el desarrollo de este trabajo.

2.1. Control en el vuelo de drones multirrotor

En materia de control de vuelo de drones, los dos controladores más importantes actualmente son el 'Differential Flatness Controller' (DFC) y el 'Model Based Predictive Controller' (MPC) [2] o Controlador con Modelo Predictivo. El DFC aprovecha la propiedad de 'flatness' de la dinámica del quadrotor para generar entradas de 'attitude' que se pueden seguir de manera precisa mediante el método de 'incremental nonlinear dynamic inversion' (INDI) [3]. El MPC, por otro lado, es un algoritmo que genera acciones de control óptimas dentro de un horizonte de predicción, permitiendo la introducción de restricciones en el problema de control [4]. El 'Model Predictive Contouring Control' (MPCC) [5] fusiona el problema del cálculo de acciones de control con el de generación de trayectorias para atacarlos a la vez. En [2] se comparan ambos controladores en el contexto del vuelo ágil de drones (hasta 20 m/s). Este estudio sitúa por delante al controlador MPC en robustez y precisión en el seguimiento de trayectorias gracias a su naturaleza activa y predictiva. La principal ventaja del control reactivo del DFC reside en su menor coste computacional.

Al aterrizaje autónomo de drones se han aplicado una gran variedad de técnicas de control para sistemas no-lineales. Métodos como el 'Backstepping' o la Linealización por Realimentación se usan en [6] y [7] para cubrir las no linealidades presentes en el modelo de un quadrotor. La lógica borrosa se aplica al control en numerosos campos, el aterrizaje de drones entre ellos. En [8] [9], se diseñan Controladores de Lógica Borrosa (FLC) para lograr el aterrizaje.

El control óptimo también se aplica directamente al problema del aterrizaje en su versión no-lineal (NMPC). El controlador se combina con otras técnicas para mejorar la maniobra. En [10] se propone una cooperación entre dron y USV para lograr un aterrizaje en plataforma móvil basado en NMPC. En [11] se añade 'Constraint Aware Particle Filtering' (CAPF) al controlador para predecir las posiciones futuras de un barco sobre el que aterrizar. En [12], la Transformada Rápida de Fourier (FFT) se aplica a la información de la posición del barco recogida por una cámara para construir un modelo de predicción de una plataforma móvil.

La Inteligencia Artificial y el Aprendizaje Automático también han cogido peso en el vuelo y aterrizaje de drones. El Aprendizaje por Refuerzo (RL) y el Aprendizaje por Refuerzo Profundo (DRL), que incorpora redes neuronales, permiten realizar un control sin modelo a niveles de estado y de actuadores. El estudio comparativo realizado en [13] enfrenta al control óptimo y al aprendizaje por refuerzo en el contexto de las carreras autónomas de drones. El control mediante RL permite representar comportamientos imposibles para para control óptimo, además de ser capaz de optimizar objetivos a nivel de tarea. Sin embargo, en un ambiente menos controlado que un circuito de carreras, el RL presenta mayores problemas de generalización que incrementan enormemente la cantidad de datos y tiempo necesarios para el entrenamiento [14]. Existen aún así diversos trabajos en los que se logra el aterrizaje autónomo usando algoritmos de RL como Q-learning y Double Q-Learning [15], TD3 [16] o PPO [17].

2.2. Caracterización y modelado de drones

De acuerdo con la literatura, el método general de obtención de parámetros de caracterización es el cálculo online mediante experimentos con bancos de prueba [18]. Al seguir este método, lo normal es separar entre cálculo de parámetros del dron y cálculo de parámetros específicos de los motores. Los trabajos actuales revisados siguen esta línea, separando el trabajo realizado en cálculo de parámetros del dron como su inercia [19] o drag aerodinámico [20], de los que se centran en el modelado de los motores [21].

Del trabajo centrado en caracterización del dron se concluye una clara inclinación a la obtención de matrices de inercia [22]. La caracterización de efectos aerodinámicos es menos común por su menor impacto en el vuelo del dron y por la mayor dificultad que entrañan los métodos para su cálculo.

Al no disponerse de los medios para el cálculo de efectos y rozamientos aerodinámicos del dron como túneles de viento o CFD [23], se limita la caracterización del dron al cálculo de sus componentes principales de inercia y se estudian los métodos usados para ellos en el Estado del Arte.

En [24] se usa un péndulo bifilar para calcular las inercias del dron en los tres ejes principales midiendo el periodo de oscilación. En [19], se apoyan en visión monocular para realizar una medición más precisa del periodo de oscilación y afinar la medida de la inercia. La sencillez de este método y su fácil recreación han motivado su selección para la estimación de la inercia del dron en este trabajo.

Si no es posible el conocimiento a priori de la inercia del quadrotor, se puede estimar durante el vuelo. En [25] se usa un 'All-accelerometer', un sistema compuesto por tres acelerómetros tri-axiales. En [26] se diseña un observador y se propone un algoritmo recursivo para estimar una inercia que converge al valor real.

Como última opción respecto a la inercia, se puede realizar un control del dron que no requiere del cálculo de esta matriz ni de prácticamente ningún otro parámetro. En [27] se consigue un control en 'attitude' del dron recurriendo a la identificación de una función de transferencia para los ángulos roll y pitch. En [28] y [29] se proponen arquitecturas de control adaptativo que compensan el error y la incertidumbre que suponen el no conocer la inercia para no tener que calcularla. Estos métodos son especialmente útiles en casos en los que la inercia es variable [30]

Los parámetros que caracterizan a los motores están bastante acotados, habiendo más variedad en la forma de calcular datos de empuje con los que se obtienen los coeficientes para modelar motores. Todo el trabajo realizado en este campo pasa por el cálculo de los coeficientes de thrust y torque, además del efecto del drag propio de cada motor.

De manera teórica, la Teoría Clásica de Momento se ha aplicado para la obtención del coeficiente de empuje de un rotor. No obstante, esta teoría no modela muchos de los fenómenos aerodinámicos del propulsor, y solo es realmente válida si se asumen ángulos pequeños, lo cual no es posible si el propulsor es pequeño (en torno a 10 cm de radio) o si la velocidad del dron es elevada (por encima de los 12 m/s) [31]. Para regímenes de vuelo más agresivos en los que efectos como el drag no pueden ser ignorados, se extiende la Teoría de Momento incorporando la 'Blade Element Theory' (BET) para dar lugar a la 'Blade Element Momentum Theory' (BEMT). Esta última permite tener en cuenta factores como el 'chord' o ancho de la hélice a lo largo de su longitud y su ángulo de inclinación respecto del disco del rotor en el cálculo del empuje generado por el propulsor y que influyen en el coeficiente de empuje [32].

La descripción teórica de los multirrotores de menor tamaño es menos efectiva por la presencia de efectos aeroelásticos y porque la influencia de cualquier fuente de error es mayor [33]. Además, en ocasiones necesita el apoyo de un método experimental para la obtención de algunos datos [31]. Al igual que en la medida de parámetros más genéricos del quadrotor, los métodos experimentales consisten en el empleo de maquinaria como túneles de viento que puedan medir el desempeño del propulsor usando sensores. En [34] se obtiene una caracterización completa de un propulsor y de la interferencia del brazo del dron en la corriente de aire proyectada por el motor. En [33] se usa un túnel de viento para hacer un estudio de la influencia de la geometría del propulsor en el 'pitch', la distancia que avanza el propulsor con una revolución.

Otra alternativa a los túneles de viento son los 'thrust stand', estructuras estáticas a las que se ancla el propulsor para llevar a cabo diversas medidas. Estas estructuras están dotadas de sensores de fuerza y de velocidad para obtener datos que puedan ser usados en el cálculo de los parámetros aerodinámicos del rotor [35]. En [36] se desarrolla un 'stand' personalizado, pero existen 'stands' comerciales como el 'WF-CO-30KGF Coaxial' de Wing Flayng o el 'Series 1585' de TYTO Robotics utilizado en este trabajo. El cálculo empírico de parámetros aerodinámicos tiene la ventaja adicional de poder realizar medidas del rendimiento del propulsor para distintos valores de entrada del ESC en PWM, con lo que resulta muy sencillo calcular la curva de 'thrust' del motor, más conveniente que simplemente obtener un coeficiente de 'thrust' dada la no-linealidad en la relación entre la alimentación del motor y la velocidad de giro del mismo [37].

Un software de 'Computational Fluid Dynamics' (CFD) pueden usarse para caracterizar los rotores. Este software permite la simulación aerodinámica de los propulsores para obtener parámetros de interés y fuerzas mediante la introducción de un modelo 3D del propulsor. En [38] se usa software de este tipo para caracterizar un propulsor y comparar los resultados con los obtenidos experimentalmente en túneles de viento, validando el uso del CFD. En [23] se usa CFD para validar un nuevo diseño de hélice para propulsores que pretende minimizar el ruido en la medida de la fuerza generada.

En [39] se propone el cálculo de un 'thrust map', un mapa de valores que permite conocer con mayor exactitud el empuje generado por el motor para distintas señales de actuación y que demuestra mejorar el desempeño del control de un cuadricóptero.

Teniendo en cuenta los distintos métodos analizados, por disponibilidad y presupuesto se seleccionan los métodos de péndulo bifilar para la estimación de la inercia y de cálculo de parámetros del motor con thrust stand para modelar los coeficientes del motor.

2.3. Vuelo de drones en simulación

Se ha hecho un estudio de los principales simuladores de vuelo y de la manera en la que cada uno simula las físicas de drones y motores.

En [40] se comparan los simuladores más usados en la actualidad. Existen simuladores fotorealistas muy potentes como Isaac Sim, con gran fidelidad, o Isaac Gym y Aerial Gym, acelerados por GPU para su aplicación a algoritmos de aprendizaje por refuerzo. Sin embargo, estos simuladores son muy pesados y no son de código abierto. Además, su simulación de físicas es limitada, no permitiendo simular viento ni drag aerodinámico. El simulador Flightmare [41] también es fotorealista y requiere menos capacidad de computación que los anteriores, pero sigue siendo un simulador más enfocado al Aprendizaje Automático.

AirSim es un simulador más apropiado para simulación de físicas, con una amplia gama de sensores y efectos temporales que se han aplicado a numerosos desarrollos [42]. No obstante, AirSim es también un simulador pesado y que actualmente no está siendo mantenido y que no ha sido migrado a ROS 2.

El simulador Gazebo [43], pese a su pobre fidelidad visual, cuenta con un motor de físicas potentes con el que implementa aerodinámica e hidrodinámica y permite vuelo con enjambres de drones y manipulación aérea. Es de código abierto y cuenta con una gran comunidad que ha añadido implementaciones específicas para el vuelo de drones y la simulación de motores y vuelo. Su compatibilidad con el sistema ROS 2 sobre el que funciona el framework Aerostack2 en el que se basa este trabajo motiva su selección para los experimentos de simulación.

3. Fundamento Teórico

En este capítulo se exponen brevemente los conceptos que forman la base teórica de este trabajo y las herramientas usadas en el desarrollo del mismo.

3.1. Modelo cinemático y dinámico del dron quadrotor

Se presenta a continuación el modelo cinemático y dinámico de un cuadricóptero, así como de su sistema de actuación.

Un quadrotor es un sistema dinámico con 6 grados de libertad, que son las translaciones en los 3 ejes principales y las rotaciones respecto a los mismos en ángulos de Euler RPY, y 4 entradas de actuación que se corresponden con las velocidades angulares de los motores.

$$X = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix} \quad U = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (3.1)$$

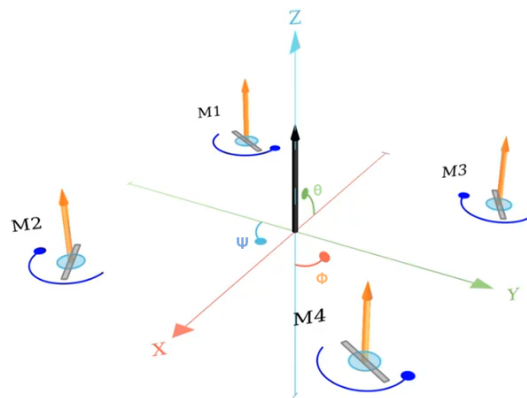


Figura 3.1: Esquema de un cuadricóptero con variables de estado y principales fuerzas de los motores

El modelo cinemático del dron en el espacio de estados queda definido por estos 6 grados de libertad, que se han separado en posición p y orientación o attitude' q del dron, y sus respectivas derivadas. La derivada de la posición p es la velocidad lineal del cuerpo del dron \vec{v}

$$\dot{p} = \vec{v} \quad (3.2)$$

A su vez, la derivada de la velocidad es la aceleración lineal del cuadricóptero \vec{a}

$$\dot{\vec{v}} = \vec{a} \quad (3.3)$$

La aceleración lineal se calcula usando la Segunda Ley de Newton

$$\vec{a} = \frac{1}{m} \vec{F} \quad (3.4)$$

donde m es la masa total del cuadricóptero y \vec{F} es la suma total de fuerzas aplicadas sobre el dron. Esta suma de fuerzas, resultante de las fuerzas aplicadas por los rotores y las fuerzas externas, se obtienen más adelante al hacer el análisis del modelo dinámico.

La derivada de 'attitude' q depende de la velocidad angular del cuadricóptero $\vec{\omega}$ y se obtiene mediante el producto Hamiltoniano

$$\dot{q} = \frac{1}{2} \cdot q \times \vec{\omega} \quad (3.5)$$

Para calcular la derivada de la velocidad angular del dron se recurre al momento angular. El momento angular \vec{L} de un cuerpo rígido es el producto de su velocidad angular ω_B y su matriz de inercia I

$$\vec{L} = I \cdot \omega_B \quad (3.6)$$

Considerando el efecto de Coriolis, la derivada del momento angular se puede calcular como

$$\frac{d\vec{L}}{dt} = I \cdot \frac{d\vec{\omega}}{dt} + \omega_B \times (I \cdot \omega_B) \quad (3.7)$$

Por otro lado, la derivada del momento angular angular también es la suma de todo los momentos que actúan sobre el cuerpo

$$\frac{d\vec{L}}{dt} = I \cdot \frac{d\vec{\omega}}{dt} = \sum_{i=1}^n \vec{\tau}_i \quad (3.8)$$

donde τ_i es el momento aplicado sobre el cuerpo por la i -ésima fuerza. La suma de momentos que provocan los motores y las fuerzas externas también se obtienen en el posterior análisis dinámico. Combinando 3.7y 3.8 y reorganizando términos se obtiene una expresión para el cálculo de la aceleración angular

$$\frac{d\vec{\omega}}{dt} = I^{-1} \cdot \left[\sum_{i=1}^n \vec{\tau}_i - \omega_B \times (I \cdot \omega_B) \right] \quad (3.9)$$

Con las derivadas calculadas como ecuaciones de estado, queda definido el modelo del dron

$$\dot{X} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ \frac{1}{2} \cdot \mathbf{q} \times \vec{\omega} \\ \frac{1}{m} \vec{F} \\ I^{-1} \cdot (\sum \vec{\tau}_i - \vec{\omega} \times I \cdot \vec{\omega}) \end{bmatrix} \quad (3.10)$$

Los motores, por su parte, se pueden modelar como sistemas de primer orden con una constante de tiempo τ y una referencia de velocidad angular deseada ω_{des}

$$\frac{d\omega}{dt} = \frac{1}{\tau} (\omega_{des} - \omega) \quad (3.11)$$

Las fuerzas y momentos que producen los motores son función de su velocidad angular y de dos parámetros aerodinámicos característicos del motor: el coeficiente de empuje k_f y el coeficiente de momento k_t

$$F_m = k_f \cdot \omega^2 \quad (3.12)$$

$$\tau_m = k_t \cdot \omega^2 \quad (3.13)$$

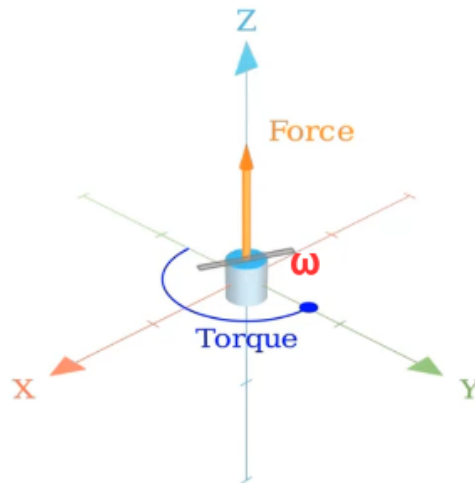


Figura 3.2: Esquema de un motor con las principales fuerzas que produce

La fuerza ejercida por cada motor se transforma al sistema de referencia del cuadricóptero a través de su matriz de rotación R_{BM} y de traslación T_{BM} . Normalmente, los motores están alineados con el sistema de referencia del cuerpo del dron, por lo que R_{BM} es la matriz identidad. Además, los motores solo producen fuerza en el eje Z del sistema de referencia inercial del dron.

Con estas consideraciones, la suma total de fuerzas y momentos producidas por los motores se expresa con

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} 0 \\ 0 \\ k_f \omega^2 \\ T_{y_{BM}} k_f \omega^2 \\ T_{x_{BM}} k_f \omega^2 \\ k_t \omega^2 + J_{motor} \alpha \end{bmatrix}_i \quad (3.14)$$

Que se puede factorizar como

$$\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} k_f \\ T_{y_{BM}} k_f \\ T_{x_{BM}} k_f \\ k_t \end{bmatrix}_i \cdot [\omega^2]_i + \begin{bmatrix} 0 \\ 0 \\ 0 \\ J_{motor} \end{bmatrix}_i \cdot [\alpha]_i = M \cdot [\omega^2]_i + J \cdot [\alpha]_i \quad (3.15)$$

donde $T_{y_{BM}}$ y $T_{x_{BM}}$ son las distancias en X e Y que separan al motor del centro de gravedad del cuerpo, J_{motor} es la inercia rotacional del motor en el sistema de referencia del motor y α la aceleración angular del motor.

Las fuerzas y momentos del entorno las producen principalmente la gravedad, el viento y la resistencia del aire. Las fuerzas medidas en el sistema de referencia del mundo se resumen a continuación

$$\vec{F}_W = \vec{F}_{gW} + \vec{F}_{dragW} + \vec{F}_{stochasticW} \quad (3.16)$$

$$\vec{F}_{gW} = \vec{Z}(m \cdot g) \quad (3.17)$$

$$\vec{F}_{dragB} = -\frac{1}{2} \rho A C_D \vec{v}_B^2 = -k_{drag} \cdot \vec{v}_B^2 \quad (3.18)$$

$$\vec{F}_W = R_{WB} \cdot \vec{F}_{dragB} \quad (3.19)$$

$$\vec{F}_{stochasticW} = \vec{N}(0, \sigma^2) \quad (3.20)$$

La fuerza de drag se corresponde con la resistencia del aire, y el signo negativo de \vec{F}_{drag} se debe a la oposición de esta al movimiento del dron. Las fuerzas de perturbación debidas a causas desconocidas de $\vec{F}_{stochastic}$ se modelan como una distribución normal de media 0 y desviación típica σ .

En cuanto a los momentos externos, se pueden separar de igual forma en resistencia aerodinámica $\vec{\tau}_{aeroB}$ y ruido estocástico $\vec{\tau}_{stochasticB}$

$$\vec{\tau}_B = \vec{\tau}_{aeroB} + \vec{\tau}_{stochasticB} \quad (3.21)$$

$$\vec{\tau}_{aeroB} = -\vec{\omega} \times (I_{aero} \cdot \vec{\omega}) \quad (3.22)$$

$$\vec{\tau}_{stochasticW} = \vec{N}(0, \sigma^2) \quad (3.23)$$

En el modelo propuesto en la ecuación 3.10 se comprueba la relación de la dinámica del dron con su matriz de inercia y su masa. También en las ecuaciones 3.12 y 3.13 queda reflejado el papel de los coeficientes aerodinámicos en las fuerzas que mueven el cuadricóptero.

3.2. Control de vuelo de drones

Como se puede comprobar en el modelo, las 4 entradas para controlar los 6 grados de libertad hacen del cuadricóptero un sistema sub-actuado. La matriz M de la ecuación 3.15 se conoce como 'mixer matrix' o 'allocator matrix', y relaciona las velocidades de giro de los motores con las fuerzas producidas por los mismos. Las dos primeras filas de esta matriz, correspondientes a las dinámicas en los ejes X e Y, son nulas, por lo que su determinante es cero y no tiene inversa. Esto implica que el dron no puede simplemente desplazarse en X o en Y, sino que su dinámica en estas direcciones está acoplada al giro del dron en Pitch para moverse en X y en Roll para moverse en Y.

De acuerdo con la literatura, existen varios niveles a los que se puede realizar el control de vuelo de un dron. El modo de control elegido determina la referencia que recibe la controladora de vuelo. De más alto a más bajo nivel, estos son

- **Control de posición**, con referencias de

$$p = [x, y, z, \psi]$$

donde x, y, z , son posiciones a lo largo de los respectivos ejes en metros (m) y ψ es el ángulo yaw en radianes (rad).

- **Control de velocidad**, con referencias de

$$v = [v_x, v_y, v_z, \omega_z]$$

donde v_x, v_y, v_z son las velocidades lineales del dron en los respectivos ejes en metros por segundo (m/s) y ω_z es la velocidad en yaw en radianes por segundo (rad/s).

- **Control de trayectoria**, con referencias de

$$Tray = [p, v, a, \psi]$$

donde p es la posición del dron, v es la velocidad del dron, a es la aceleración lineal en metros por segundo al cuadrado (m/s^2) y ψ es el ángulo yaw.

- **Control en 'Attitude'**, con referencias de

$$Attitude = [T, \phi, \theta, \psi]$$

donde T es el empuje total o 'thrust' en Newton (N) y ϕ, θ, ψ son los ángulos roll, pitch, yaw (RPY) del dron respecto a cada uno de los ejes principales en radianes (rad).

- **Control 'Acrobatic' o 'Rates'**, con referencias de

$$ACRO = [T, p, q, r]$$

donde T vuelve a ser el empuje total y p, q, r son velocidades angulares del dron en radianes por segundo (rad/s).

- **Control a nivel de motor**, con referencias de

$$\omega_m = [\omega_1, \omega_2, \omega_3, \dots, \omega_n]$$

donde ω_i son las velocidades angulares de cada motor en (rad/s).

El control en 'Attitude' consiste en el control sobre la orientación del dron en ángulos de Euler Roll, Pitch y Yaw (RPY). En ocasiones se realiza control sobre ángulos de Roll y Pitch y sobre velocidad angular en Yaw. El control en 'ACRO' se realiza sobre las velocidades angulares del dron respecto a los tres ejes y es el control a nivel más bajo al que se puede controlar antes de llegar al motor. La última etapa de control para llegar a velocidades angulares de los motores se realiza mediante la técnica de 'Incremental Non-Linear Dynamics Inversion' (INDI).

El control a más alto nivel resulta más sencillo, pero también es menos reactivo. Las aplicaciones que requieren modos de vuelo ágiles y exigentes necesitan un control rápido y lo más cercano posible al nivel de actuador. Los controladores más usados en el estado del arte, el 'Differential Flatness Based Controller' en control reactivo y el 'Model Predictive Controller' en control predictivo, controlan a nivel de 'Attitude' o 'ACRO'.

Para hacer control a estos niveles bajos también es necesario que el dron cuente con una controladora de vuelo que acepte entradas de 'Attitude' o 'ACRO' y las traduzca a comandos de actuación para los motores.

3.3. Componentes de un dron cuadricóptero

Los cuadricópteros siguen una arquitectura general que permite su control tanto manual como autónomo y que se compone de los elementos principales que se describen a continuación [44].

El '**frame**' es la base del dron, la plataforma principal sobre la que se montan todos los componente y encargada de protegerlos en caso de impacto. Deben alcanzar un equilibrio entre ligereza, para no sumar mucha carga al dron, y resistencia, para resistir los impactos. Se construyen en plástico, aluminio o fibra de carbono.

Los **motores** generan el empuje necesario para actuar el dron y conseguir su vuelo. Los más comunes son de corriente continua sin escobillas (BLDC) o motores trifásicos. Sobre ellos se monta una **hélice**, que es la que genera el empuje gracias al movimiento del motor. El número de palas de la hélice, su longitud, el 'chord' o ancho de la pala a lo largo de su longitud y el ángulo de rotación de la pala respecto al plano de la hélice son los parámetros más importantes que determinan el rendimiento del propulsor. Es importante conocer con exactitud las fuerzas que genera el propulsor con su giro para calcular las actuaciones que consigan que el estado del dron evolucione de acuerdo con las referencias de control.

Para poder controlar la velocidad de giro de los motores se introduce una unidad de **Control Electrónico de Velocidad** (ESC por sus siglas en inglés, Electronic Speed Controller), que convierte las instrucciones de la controladora de vuelo en señales de actuación para los motores.

La **Controladora de Vuelo o 'Flight Controller Unit' (FCU)** es el cerebro del dron. Consiste en una placa dotada de la electrónica y sensores necesarios para recibir las señales de control y ajustar la velocidad de giro de los motores en función del estado del dron. Este estado lo miden usando una Unidad de Medida Inercial (IMU) con la que están equipados. Estas operaciones las realiza usando el firmware que tiene cargado y que determina cómo se debe establecer la comunicación con la controladora y el formato de los mensajes que recibe y envía.

De manera general, la controladora recibe señales normalizadas que convierte a señales de 'throttle' para los motores. Este 'throttle' representa un valor de intensidad media de señal PWM. La velocidad de giro de los motores y en consecuencia el empuje que generan no depende linealmente del valor de esta señal de 'throttle'. La controladora de vuelo ignora esta relación no-lineal

y se limita a enviar señales normalizadas a partir de las acciones calculadas por el controlador, por lo que es importante conocer de antemano las curvas de thrust de los motores.

Para enviar señales de control al dron, la controladora de vuelo también debe estar dotada de un **módulo de recepción**, generalmente de señales de radio. Con ese receptor, se puede usar un mando radiocontrol (RC) para enviar las señales de throttle que comandan al dron. También se puede usar un módulo de transmisión para enviar actuaciones calculadas por un sistema de vuelo autónomo en un ordenador desde tierra, llamado 'ground station' en la literatura.

Otra alternativa para el vuelo autónomo del dron es montar un **ordenador de a bordo** sobre el dron. En esta configuración, el ordenador es el encargado de calcular las acciones de control para el dron a partir de la información sensorial que puede recoger directamente de los sensores del dron o que puede recibir mediante un módulo de radio o Wifi. El ordenador carga todo el software necesario para realizar los cálculos. En este caso, el ordenador es también el responsable de establecer la comunicación con la controladora de vuelo y de hacerle llegar los comandos calculados. El montaje de un ordenador de a bordo reduce mucho los retardos en las comunicaciones, pero su capacidad de cómputo está limitada por su tamaño.

Finalmente, la alimentación de todos estos componentes corresponde a las baterías, de tipo LiPO con alto ratio de descarga para ayudar a conseguir las velocidades de giro necesarias en los motores. Esta alta descarga y su tamaño limitado por la capacidad de carga del dron resultan en una autonomía pobre de tiempo de vuelo que caracteriza a este tipo de aeronave.

3.4. El framework Aerostack2

El software de autonomía de vuelo empleado en este trabajo es Aerostack2. Aerostack2 es un software de código libre que permite a los desarrolladores montar arquitecturas de control para el vuelo de sistemas robóticos aéreos implementando algoritmos visión por computador, control, localización, planificación, mapeado, etc [1]. Esto permite al desarrollador centrarse en su aplicación específica sin preocuparse del problema del vuelo de la aeronave.

Aerostack2 está programado sobre ROS 2, concretamente sobre las versiones Humble y Galactic. El framework aprovecha el sistema de publicación y suscripción de ROS 2 para ejecutar los módulos como nodos programados principalmente en C++, establecer la comunicación entre los módulos mediante el sistema de publicación-suscripción y montar los servicios y acciones que permiten reajustar parámetros y controlar la ejecución de las misiones. También se sirve de mecanismos como el sistema de transformadas de la librería 'tf2' de ROS 2 para trabajar con los distintos sistemas de referencia, o las librerías y puentes que existen entre ROS 2 y el simulador Gazebo para la simulación de misiones.

La arquitectura modular de Aerostack2 permite aislar el sistema de control que compone de la plataforma a la que está dirigida. Gracias a esto la integración entre distintos drones y simuladores es muy sencilla. Esto facilita la realización de estudios comparativos entre plataformas [45].

La Figura 3.3 resume la estructura y los módulos en los que Aerostack2 divide el control de vuelo de drones

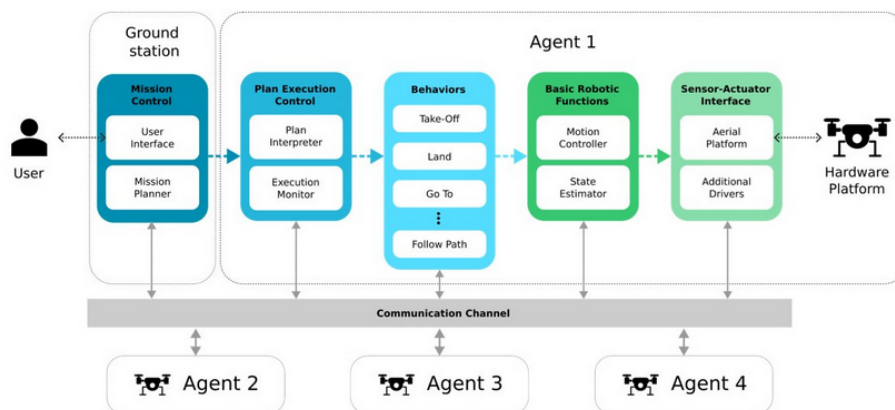


Figura 3.3: Esquema de la arquitectura del framework de robótica aérea Aerostack2 [1]

Empezando por el nivel más bajo, las interfaces de sensor-actuador son las encargadas de la comunicación directa con el hardware, ya sean los sensores, para los que se implementan diversos drivers adicionales, o las propias plataformas aéreas. Aerostack2 incorpora una clase abstracta 'AerialPlatform' que maneja todas las acciones que una plataforma debe poder realizar de forma general para poder particularizar para cada plataforma mediante la herencia de esta clase. Las plataformas en Aerostack2 son entonces las encargadas de transmitir comandos de actuación y otras peticiones de los distintos módulos de Aerostack2 a la aeronave de forma específica, y de recoger la información sensorial de esta para transmitirla a las demás capas.

Por encima de las plataformas están las 'Basic Robotic Functions', separadas principalmente en Motion Controller y State Estimator. Estos módulos se encargan de realizar las operaciones necesarias para mantener la localización y el control de vuelo del dron. Estos módulos, que son el núcleo del sistema de control, funcionan a través de la implementación de diversos plugin con una interfaz común. Esto permite adaptar cada módulo a las necesidades específicas de cada misión, y adecúan las entradas y salidas para poder estandarizar la interfaz del plugin.

Los plugin del State Estimator implementan distintos algoritmos de localización. El plugin de 'mocap_pose' permite la conexión Wifi con un sistema de 'Motion Capture' como el sistema Optitrack usado en este trabajo para obtener mensajes de la captura de movimiento del dron y realizar una estimación de su estado. Este módulo también monta el árbol de transformadas que permite localizar cada elemento respecto a los sistemas de referencia convenientes.

Los plugin del Motion Controller implementan controladores como un PID para control de velocidad o los controladores basados en modelo de Differential Flatness, usado en el control de vuelo para este trabajo, o MPC. El módulo también maneja los cambios entre los posibles modos de control configurados inicialmente.

Los Behaviors de Aerostack2 conforman una capa lógica que permite formular misiones de forma unificada y simple. Cada Behavior es una capacidad básica del vuelo de un dron, como el despegue (Take-Off), el aterrizaje (Land) o el seguimiento de caminos (Follow-Path). También incluyen habilidades como la generación de trayectorias para su seguimiento (Trajectory-Generation). La generación de trayectorias mediante Behavior se usa como base para el diseño de los experimentos de vuelo de este trabajo. La lógica de estos comportamientos está encapsulada en una interfaz común que permite iniciar, pausar, reanudar y detener los Behavior, además de un servicio 'modify' para ajustar el comportamiento.

Por encima de estos módulos Aerostack2 implementa una capa de control de ejecución de la

misión que permite a los desarrolladores escribir conjuntos de tareas que se realizan mediante la activación y desactivación de los distintos Behaviors. Para ello, Aerostack2 proporciona una API de Python que permite interactuar con el framework mediante la implementación de una clase 'DroneInterface' en la que se especifican comandos para los drones y con la que se puede obtener información del estado de cada uno. Esta API se usa en este trabajo para el diseño de la misión de aterrizaje. Otras alternativas son el Mission Interpreter, que lee tareas de un fichero en formato JSON o la posibilidad de describir la misión con un Behavior Tree compatible con la librería BehaviourTreeCPP mediante acciones de ROS 2.

Por último, Aerostack2 ofrece diversas interfaces al usuario para supervisar y controlar el vuelo. Entre ellas, herramientas para el desarrollador como el Alphanumeric Viewer que monitorea el estado del dron o el Keyboard Teleoperation que permite controlar directamente uno o varios drones.

Todas estas utilidades del framework Aerostack2, tanto en la configuración del vuelo como en la capacidad de comunicación con simulación, junto con el conocimiento previo de la herramienta han llevado a la selección de este framework como base para las implementaciones de este trabajo.

Los trabajos sobre Aerostack2 se estructuran en 'proyectos'. Cada proyecto cumple una función específica y se ha diseñado con un objetivo. En estos proyectos se incluyen los distintos ficheros de configuración para Aerostack2, la lógica externa necesaria para llevar a cabo las misiones y las propias misiones que interactuarán con el framework. En este proyecto también se incluye toda la configuración para la plataforma que se usará como interfaz con la aeronave en cuestión.

3.5. El simulador Gazebo

Gazebo es un simulador muy usado en robótica por su realismo a la hora de simular físicas precisas y modelos de ruido. Integra varios tipos de sensores y cuenta con toda una librería de puente con el sistema ROS 2 para el intercambio de mensajes [43].

El sistema de comunicación basado en publicación-suscripción de Gazebo es muy similar al de ROS 2. Los distintos modelos pueden publicar información acerca de su estado y recibir mensajes para su control.

Este simulador usa el formato SDF para cargar los distintos modelos de la simulación. En un fichero de este tipo se pueden incluir modelos de drones, como los modelos básicos incluidos en el propio simulador o modelos de drones reales creados por la comunidad. A estos modelos se les configuran 'links' para enlazar los distintos eslabones del sistema que se quiere modelar y 'joints' que configuran el tipo de movimiento entre los eslabones.

Gazebo cuenta con un sistema de físicas basado en plugins que permiten extender su funcionalidad. Entre ellos, cuenta con plugin específicos para el vuelo de drones, como el plugin 'MulticopterMotorModel', en el que se pueden configurar los distintos parámetros característicos de un motor como los coeficientes aerodinámicos o el tiempo de subida del motor. También cuenta con el plugin 'MulticopterVelocityController', que recibe la configuración de los motores para realizar un control PID de los mismos. Estos plugin pueden ser incluidos en el fichero SDF del modelo de un dron para conseguir su vuelo en simulación.

Aerostack2 cuenta con una plataforma aérea específica para Gazebo 'as2_platform_gazebo', que usa la librería 'gazebo_transport' para establecer todas las comunicaciones entre el dron simulado en Gazebo y los módulos de Aerostack2. El framework también cuenta con todo un paquete 'as2_gazebo_assets' con herramientas para construir simulaciones con Gazebo y Aerostack2 y

configurarlas cómodamente desde ficheros en formato YAML.

El conocimiento de Gazebo y su compatibilidad ya probada con Aerostack2 han sido las claves de la elección de esta herramienta.

4. Caracterización y modelado del dron

En este capítulo se exponen los distintos procedimientos que se han seguido para obtener el modelo del dron, incluyendo el fundamento teórico sobre el que se fundamentan.

4.1. Componentes del dron real

La figura 4.1 muestra el dron sobre el que se ha realizado el trabajo.



Figura 4.1: Vista superior del dron real que se caracteriza

Siguiendo la arquitectura de componentes desarrollada en el apartado correspondiente de los Fundamentos teóricos (ver 3.3, el dron cuenta con los siguientes componentes

Componente	Nombre
Frame	ZMR250
Motores	XNOVA Black Thunder
Hélices	MCK 51466
ESC	Foxeer Reaper F4
FCU	Foxeer H7 MPU6000
Ordenador de a bordo	Jetson Orin NX

Tabla 4.1: Lista de componentes del dron

Se elabora el listado de parámetros del dron a caracterizar. El modelado del dron consiste entonces en la obtención o caracterización de los siguientes elementos:

- **Dimensiones del dron:** el frame del dron es un ZMR250 fabricado en fibra de carbono. Se miden la longitud de los brazos y el ángulo formado entre ellos para introducir las dimensiones en el fichero SDF del modelo de Gazebo.
- **Peso:** la masa del dron es vital en la configuración del modelo en Gazebo y en el ajuste del controlador usado (ver 3.1).

- **Inercias:** la matriz de inercia es otro parámetros presente tanto en el modelo de Gazebo como en el ajuste del control.
- **Parámetros de los propulsores:** los coeficientes aerodinámicos del propulsor intervienen directamente en el modelado (ver 3.1). Se calculan para su configuración en el modelo de motor en Gazebo. Se usan los datos del motor para calcular curvas de thrust para la controladora (ver 3.3).
- **Modo de control, controladora:** la controladora usa un firmware concreto Betaflight 4.5.2. Se configura la plataforma de Aerostack2 con datos de la controladora para realizar los vuelos. Se configura también Aerostack2 para funcionar en modo ACRO, el modo en el que funciona la controladora.

4.2. Obtención de los parámetros de caracterización

Los primeros experimentos han servido para la obtención de los parámetros con los que luego se hará la implementación del sistema de control.

4.2.1. Dimensiones y peso

Las dimensiones del dron se han medido sobre un modelo del frame ZMR250 que se puede obtener de cualquier página web repositorio de modelos. El modelo se ha abierto y montado usando la aplicación Blender, y las dimensiones han sido tomadas usando la herramienta de medidas de este software. La figura 4.2 muestra los modelos del frame y las medidas tomadas sobre él.

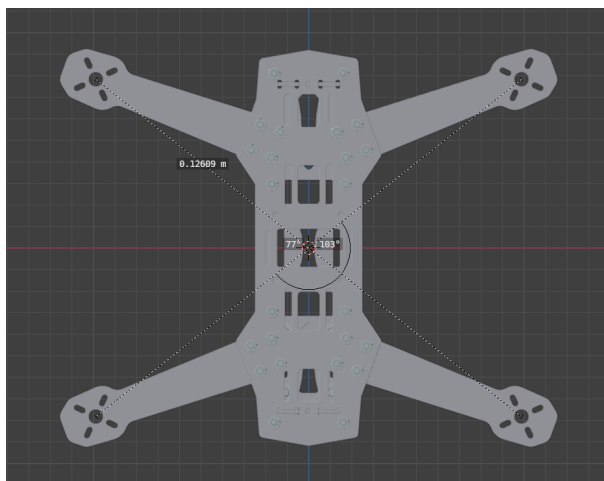


Figura 4.2: Medidas de distancia del centro del dron a cada motor y de ángulo formado por los brazos del dron tomadas sobre el modelo en Blender

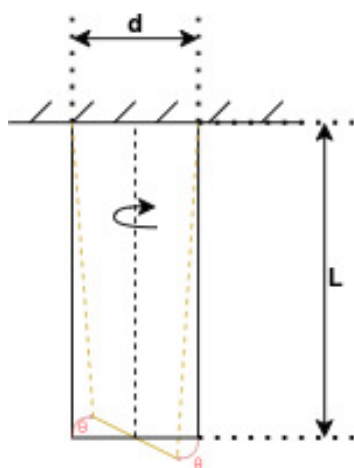
Para la masa del dron sencillamente se ha pesado usando una báscula. Los valores tomados en para estos parámetros triviales se recogen en la Tabla 4.2 a continuación

Parámetro	Valor
$m(kg)$	0,862
$L(m)$	0,12609
θ_1 (deg)	103
θ_2 (deg)	77

Tabla 4.2: Masa y dimensiones del frame 'ZMR250' del dron, medidas sobre el modelo de Blender

4.2.2. Cálculo de la inercia mediante péndulo bifilar

Para medir las inercias del dron se ha elegido el método de péndulo bifilar rotacional. Este método consiste en colgar el objeto cuya inercia se quiere medir de dos cables de igual longitud L y separados entre sí una distancia d , ambas medidas en metros. Los puntos de los que se cuelga el objeto deben elegirse de forma que el objeto quede nivelado, y que el eje que queremos medir quede alineado con las cuerdas verticales como se muestra en la figura 4.3.

Figura 4.3: Esquema del experimento de péndulo bifilar definido por la longitud de las cuerdas, L , y la separación entre ellas, d

Rotando el objeto un ángulo pequeño y liberándolo, comenzará a oscilar con un periodo dado por

$$T_0 = 4\pi \sqrt{\frac{J_{ii}L}{m_0gd^2}} \quad (4.1)$$

donde T_0 es el periodo de oscilación del péndulo, J_{ii} es el momento de inercia respecto del eje deseado, m_0 es la masa del objeto cuya inercia se quiere medir y g es la aceleración de la gravedad. De la ecuación (4.1) se despeja la siguiente expresión para el momento de inercia de cada eje

$$J_{ii} = \frac{m_0gd^2}{16\pi L} T_0^2 \quad (4.2)$$

Para lograr la mayor precisión posible en la medida del periodo de oscilación de cada experimento, se ha hecho uso del sistema Optitrack instalado en el 'CAR Arena', donde se han realizado estos experimentos. Optitrack es un sistema de captura de movimiento basado en la detección de marcas retroreflectivas mediante cámaras instaladas en distintas posiciones y alturas [46].

El software 'Motive' que acompaña al sistema Optitrack permite definir un sólido rígido a partir de los marcadores colocados sobre un cuerpo y el sistema se encarga de capturar su posición y orientación y publicarla en un tópico de ROS2 para su guardado y procesado.

4.2.3. Obtención de curvas de empuje y parámetros del motor

La controladora de vuelo MPU6000 del dron se controla usando el firmware 'Betaflight'. Betaflight recibe señales de 'throttle' normalizado entre 1000 y 2000 que traduce en señales de PWM que envía a cada motor. Para tener un conocimiento a priori del empuje que generarán los motores para las distintas señales de PWM calculadas por la controladora, se realizan ensayos con 'thrust stand' sobre los propulsores del dron.

Estos ensayos permiten calcular el empuje que genera el propulsor cuando el motor gira a una velocidad determinada. Si se mide el valor de la entrada para cada uno de esos puntos de trabajo del motor, se puede establecer una relación entre el 'throttle' aplicado y el 'thrust' generado. Esta relación queda reflejada en una 'curva de thrust' (ver Figura 4.4).

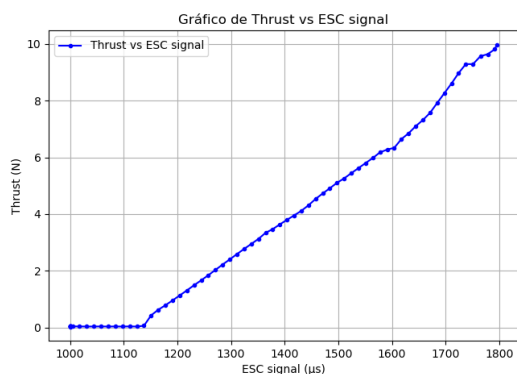


Figura 4.4: Ejemplo de curva de thrust en la que se aprecia la relación entre señal de throttle y thrust generado

Usando esta curva, un software de control de más alto nivel puede mapear una vez calculada la acción de control cuál es el valor de 'throttle' de la controladora con la que se obtendrá el empuje deseado de los motores.

Si el 'thrust stand' cuenta con sensores de velocidad angular para los motores, se pueden calcular también los coeficientes aerodinámicos del motor k_f y k_t usando las ecuaciones (3.12) y (3.13).

4.3. Caracterización en simulación

El esquema de la Figura 4.5 muestra la arquitectura del sistema de simulación de vuelo montado en Gazebo.

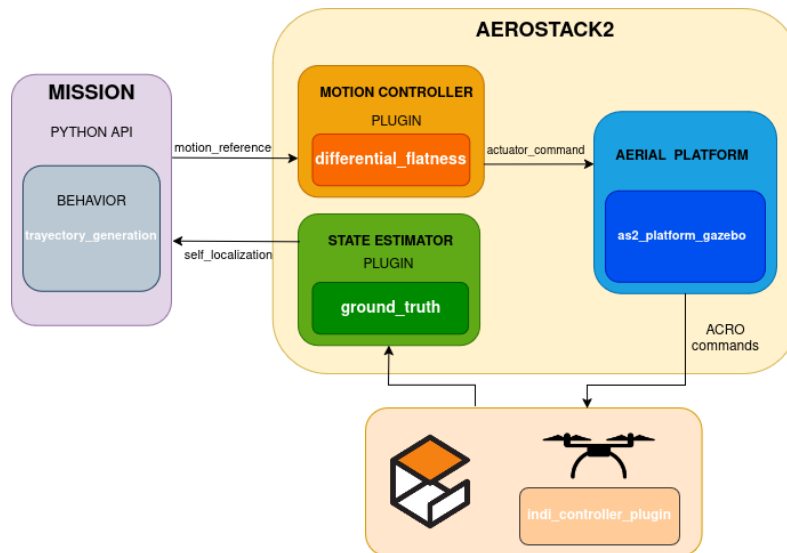


Figura 4.5: Arquitectura del sistema de control de vuelo del dron basado en Aerostack2 en el simulador Gazebo

Se ha configurado Aerostack2 para lanzar la plataforma de vuelo 'as2_platform_gazebo' con el modo de control ACRO que coincide con el modo de control que acepta el dron real. Para el Motion Controller se ha cargado el plugin del controlador Differential Flatness que genera los mensaje de 'actuator_command' que Aerostack2 envía a la plataforma. En la configuración de este plugin se ha incluido la masa del dron y el ajuste obtenido del controlador del el dron real con la intención de que el dron simulado vuele con la misma configuración que el real.

De Gazebo se obtiene directamente una media del 'Ground Truth' del dron simulado enviado directamente por el modelo. El State Estimator carga el plugin de 'ground_truth' que adapta el formato de la información recibida a un mensaje estándar de Aerostack2.

Tanto la misión de seguimiento de trayectoria como la de aterrizaje se han programado usando la Python API de Aerostack2, que recibe una serie de puntos y activa el Behavior de 'trajectory_generation' para comandar al dron.

4.3.1. Modelado del dron en el simulador Gazebo

Gazebo cuenta con unos plug-in propios para el simulado de rotores y una controladora de vuelo que realiza un control en velocidad. Además, aunque no existe ningún modelo estandarizado dentro del simulador, la comunidad ha desarrollado distintos modelos de dron que siguen una estructura similar. Siguiendo esta estructura, que es la misma que siguen los modelos de dron ya incluidos en el framework 'Aerostack2', se ha creado un fichero 'zmr250.sdf' en el que se han introducido los datos calculados en los campos de masa del dron y su matriz de inercia. Las dimensiones del dron se ven reflejadas en la posición del 'link' correspondiente a cada uno de los motores en el modelo del dron. Dado que Gazebo espera las posiciones en forma de desplazamientos en los ejes X e Y y las dimensiones tomadas sobre el dron son de longitud de los brazos y de ángulo que forman estos con el centro de gravedad del dron, la posición de cada rotor se ha obtenido con

$$dx = L * \cos(\theta) \quad (4.3)$$

$$dy = L * \sin(\theta) \quad (4.4)$$

donde dx, dy son los desplazamientos en los respectivos ejes, L es la longitud de los brazos del dron y θ el ángulo que forman los brazos con el centro del dron.

Siguiendo el sistema de templatizado de 'Aerostack2', se ha creado también un fichero 'zmr250.sdf.jinja' en el que se han incluido los plug-in de los motores y su control. El plug-in 'MulticopterMotor-Model' de Gazebo permite la configuración de diversos parámetros de un rotor para simular sus físicas. Se configuran los parámetros de 'thrust_coefficient' k_f y 'torque_coefficient' k_t , además del tiempo de subida τ y las velocidades máximas de giro del rotor.

Gazebo no tiene implementado ningún plugin para realizar control en modo ACRO. El único controlador de vuelo realiza control recibiendo referencias de velocidad. Para poder volar el dron de Gazebo usando comandos de ACRO o 'rates' se desarrolló como trabajo previo a este proyecto un plugin que implementa la etapa de INDI para convertir señales de ACRO en velocidades de los motores siguiendo la misma estructura que el plugin estándar de Gazebo.

Por último, a partir del modelo del que se han obtenido las dimensiones se ha montado un modelo 3D del frame completo que se ha exportado a formato SDF desde Blender para su uso en Gazebo (ver Figura 4.6. Este modelo es meramente visual y no afecta a la dinámica del dron durante la simulación.

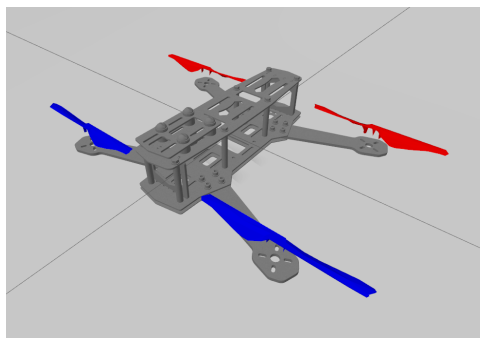


Figura 4.6: Dron modelado en el simulador Gazebo a partir del frame 'ZMR250'

4.3.2. Modelado del entorno marítimo

Para el modelado del entorno marítimo se han usado los plugin de hidrodinámicas que se implementaron para la competición MBZIRC 2021 junto con otras versiones que fueron aceptadas posteriormente para la superficie de flote de los modelos. Estos plugin permiten el ajuste de parámetros de drag y fuerza del oleaje para realizar distintas pruebas.

Se ha elegido el modelo de un Wave Adaptive Modular Vessel (WAM-V) como Vehículo de Superficie no Tripulado (USV) sobre el que realizar los experimentos en simulación (ver Figura ??). Todo modelo que deba flotar en el entorno marítimo debe incluir también los plugin de hidrodinámicas y 'buoyancy'.

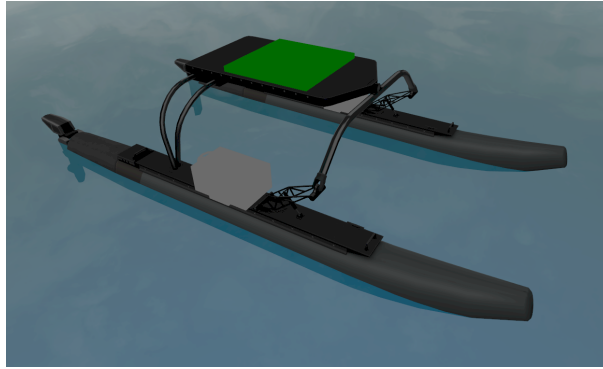


Figura 4.7: Modelo de barco WAM-V usado para los experimentos de aterrizaje en entorno marítimo

Al modelo SDF del WAM-V se han añadido también los links y joints necesarios para simular unos propulsores que pueden mover el barco usando un script de 'bash' que realiza publicaciones en topics de Gazebo a los que se suscriben estos propulsores.

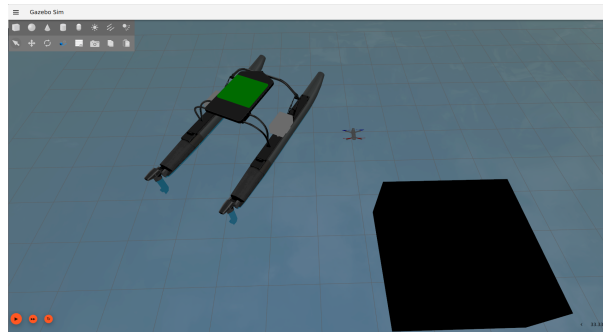


Figura 4.8: Entorno marítimo simulado en Gazebo con barco y plataforma de despegue para el dron

4.4. Caracterización en el dron real

La Figura 4.9 muestra la arquitectura usada para el vuelo del dron real.

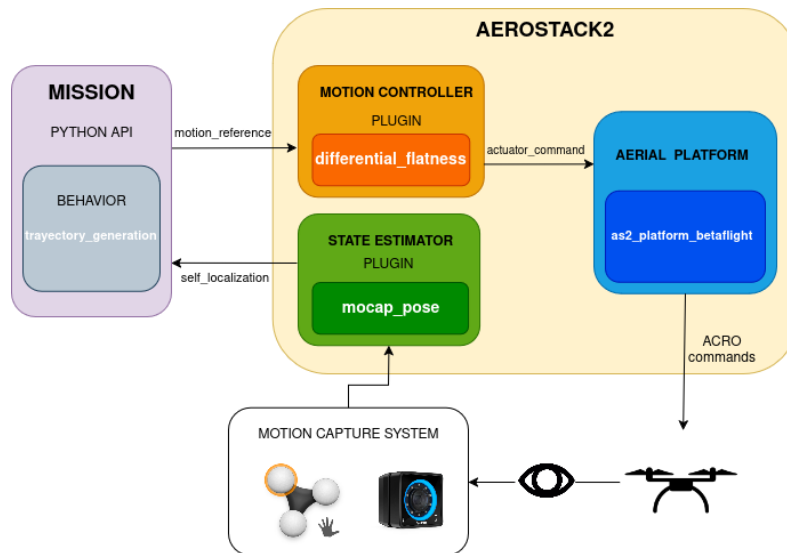


Figura 4.9: Arquitectura del sistema de control de vuelo del dron real basado en Aerostack2

Fuera del entorno de simulación la localización del dron no es trivial. Sobre el dron real se han montado marcadores reflectantes de Optitrack que permiten la detección del dron. En esta ocasión, al State Estimator de Aerostack2 se le carga el plugin 'mocap_pose' que recibe vía Wifi mensajes con la localización del dron que se encarga de nuevo de formatear en mensajes de Aerostack2.

El Motion Control se configura de manera idéntica a la arquitectura anterior de simulación. El uso de los mismos parámetros de ajuste permitirá validar la similitud en el vuelo de ambos drones.

La misión también se mantiene exactamente igual. La arquitectura modular de Aerostack2 permite que las mismas referencias se usen para comandar al dron real sin tener que ajustar nada en la misión.

El cambio más importante tiene lugar en la plataforma, que es en esencia lo que ha cambiado de un entorno a otro. Aerostack2 cuenta con una plataforma específica para el vuelo con Betaflight, 'as2_platform_betaflight', que gestiona la comunicación con cualquier controladora de vuelo que funcione con este firmware como es el caso de la Foxeer H7 con la que cuenta el dron.

El firmware de Betaflight está pensado para recibir mensajes de un mando radiocontrol mediante el protocolo Multiwii Serial Protocol (MSP). Este protocolo cuenta con un tipo de mensaje específico SET_RC en el que deserializan los comandos que se reciben del mando en 8 canales, ROLL, PITCH, YAW, THRUST (igual que un comando de ACRO si se entienden los 3 primeros campos como velocidades) y 4 canales auxiliares.

Es importante ajustar en la controladora de Betaflight cómo la controladora va a interpretar los comandos del mando, es decir, cuáles son los valores de ROLL, PITCH, YAW y THRUST que se corresponden con las salidas del mando cuando los joysticks se inclinan al máximo y la curva que relaciona todos los valores intermedios (ver Figura ??).

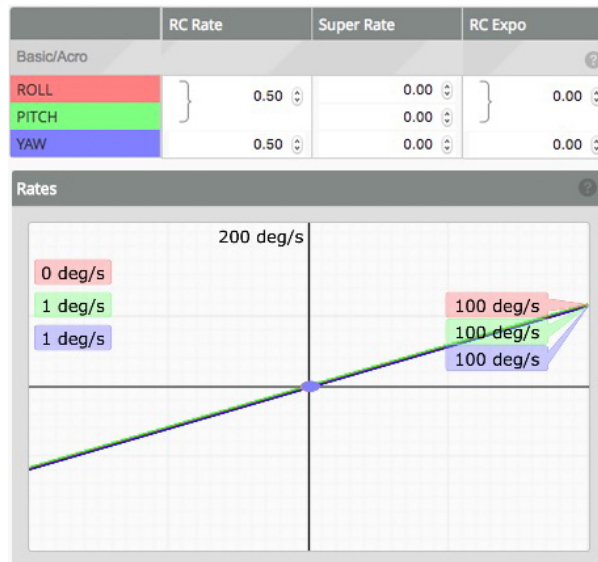


Figura 4.10: Curva de ajuste de Roll, Pitch y Yaw en controladora de Betaflight. El punto central se corresponde con un valor de 1500 de throttle, y los valores de 1000 y 2000 son las velocidades máximas en cada sentido que se pueden enviar a la controladora

Cuando se configura la plataforma de Betaflight de Aerostack2, esta interfaz hace exactamente eso. La plataforma carga una librería de comunicación MSP para comunicarse con la controladora como si fuera un mando. Cuando la plataforma recibe las instrucciones del Motion Controller, usa los valores de máximo y mínimo de velocidad angular y la pendiente de la curva configurados para traducir los comandos de ACRO en mensajes SET_RC de MSP.

Además de estos parámetros, en la plataforma se configura el 'thrust_map' calculado a partir de los datos obtenidos en experimentos de thrust stand. Esta configuración consiste en introducir los coeficientes del polinomio con el que se calculan las señales de throttle que se deben pasar a la controladora en función del valor de thrust deseado y la batería disponible.

5. Experimentos

En este capítulo se expone el diseño de los experimentos con los que se obtienen los resultados que permiten validar la metodología y el algoritmo propuestos, junto con las métricas y procedimientos con los que se evalúan esos resultados. Los dos primeros apartados se centran en la obtención de los parámetros de caracterización. El tercer apartado describe los experimentos con los que se validará el modelo obtenido tras la caracterización. En los dos últimos apartados se exponen los experimentos de aterrizaje realizados tanto en simulación como en real.

5.1. Experimentos con Péndulo Bifilar

La figura 5.1 muestra el montaje de la estructura de arco con barra fija de la que cuelgan las cuerdas que componen el péndulo. Por cada uno de los ejes para los que se quiere medir la inercia, se han fijado las cuerdas al dron de manera que la rotación se produzca exclusivamente en el eje respecto al cual se quiere medir la inercia

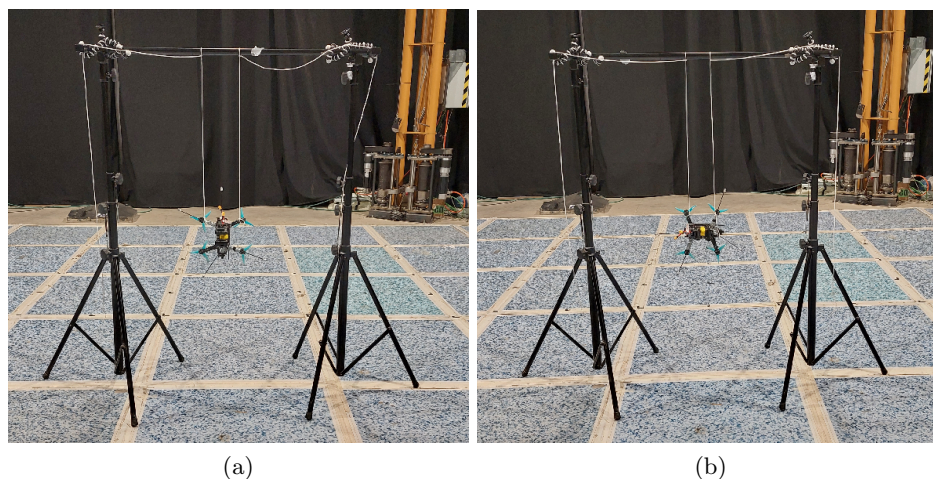


Figura 5.1: Montaje de péndulo bifilar para el cálculo de las inercias respecto a a) Eje X, b) Eje Y, c) Eje Z

Se han realizado tres experimentos por cada eje, manteniendo fija la distancia entre las dos cuerdas y variando la longitud de las mismas. Por cada experimento, se ha dejado girar el dron durante un tiempo aproximado de 20 segundos. La información de la pose del dron se ha guardado en un fichero de 'rosvbag' por cada experimento para su posterior procesado.

El procesado de los ficheros de 'rosvbag' se ha realizado con scripts de Python usando las librerías de 'bagpy' para la apertura y lectura de los datos y de 'numpy' para realizar las operaciones de filtrado de datos y cálculo de puntos máximos en la orientación del dron. Con el filtrado, se recortan los datos hasta el momento en el que el dron se libera y comienza a oscilar. Sobre los datos restantes, se aplica la función 'argrelmax()' de 'numpy', que calcula los puntos en los que se producen máximos locales. Estos puntos se corresponden con los momentos en los que el dron completa un periodo de oscilación. Un filtrado posterior elimina los 'falsos' máximos detectados cerca de los mínimos locales. Finalmente, se extraen los intervalos de tiempo comprendidos entre los sucesivos máximos para obtener múltiples medidas del periodo de oscilación sobre las que obtener una media y una desviación típica. El valor final del periodo obtenido se ha introducido

en la ecuación (4.2) del péndulo rotacional para obtener los valores de la inercia. El mismo proceso se ha repetido para cada experimento y eje, obteniendo los valores para la inercia.

5.2. Experimentos con 'thrust stand'

Para la caracterización de los motores del dron se ha usado el 'thrust stand' Series 1585 de TYTO Robotics. Este 'stand' cuenta con tres galgas extensiométricas, dos horizontales y una vertical, y con un sensor óptico para medir velocidad angular del motor.

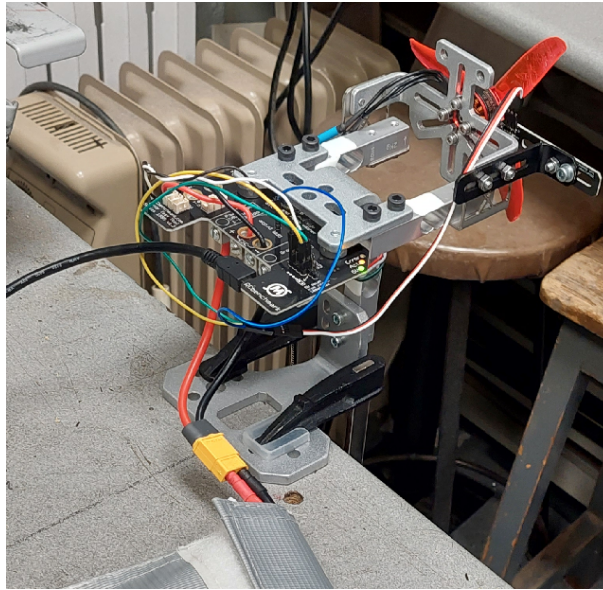


Figura 5.2: Montaje del thrust stand Series 1585 de TYTO Robotics

Sobre el stand se ha montado un variador trifásico de 45A y se ha fijado un motor como se ve en la figura 5.2. El stand y las entradas al variador se controlan mediante el software propio del fabricante RCBenchmark. Este software permite el control manual de varios rotores mediante una interfaz para el usuario y también un control automático mediante scripts programados en JavaScript. La información de los sensores se puede ver en los distintos paneles, y cuenta también con una función de captura que guarda los campos deseados y con la frecuencia deseada en ficheros CSV.

Para la caracterización de los motores, se ha programado un script de RCBenchmark que automatiza los experimentos y la lectura de datos. El script consiste en una entrada de tipo rampa ascendente con un recorrido desde un valor de 1000 de 'throttle' (0 de thrust normalizado) hasta un valor de 2000 (1 de thrust normalizado correspondiente a la máxima potencia). Tras unas décimas de segundo comienza una rampa descendente hasta el mismo valor mínimo de 1000. Aprovechando la lectura de voltaje del stand, se han realizado experimentos a distintos niveles de la batería que alimenta al motor y que es la misma que lleva el dron a bordo.

Los datos recogidos del motor para los distintos valores de 'throttle' y voltaje permiten la elaboración de un mapa de empuje que mejore el control del dron. Los datos obtenidos del sensor óptico para la velocidad de rotación permiten el cálculo de parámetros del motor para el modelo en simulación.

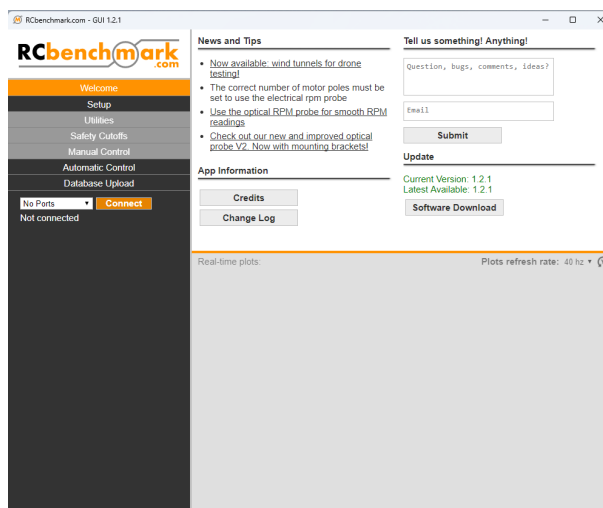


Figura 5.3: Software de captura de datos RCBenchmark para los experimentos con thrust stand

5.3. Pruebas de verificación del modelo

Para verificar la validez del modelo obtenido a partir de la caracterización del dron, se han llevado a cabo experimentos de vuelo en los que el dron real y el dron simulado han sido comandados para seguir la misma trayectoria usando la misma configuración de los parámetros del controlador.

Se ha usado como trayectoria una figura en forma de lazo generada a partir de 4 puntos. A los drones se les comanda un despegue, el seguimiento de la trayectoria y finalmente un aterrizaje. Se han realizado 3 experimentos usando esta trayectoria con el dron real, grabando toda la información para su posterior reproducción. La misma configuración inicial del dron en cada experimento se ha replicado en el entorno de simulación y se ha lanzado la misma referencia de trayectoria para el dron simulado.

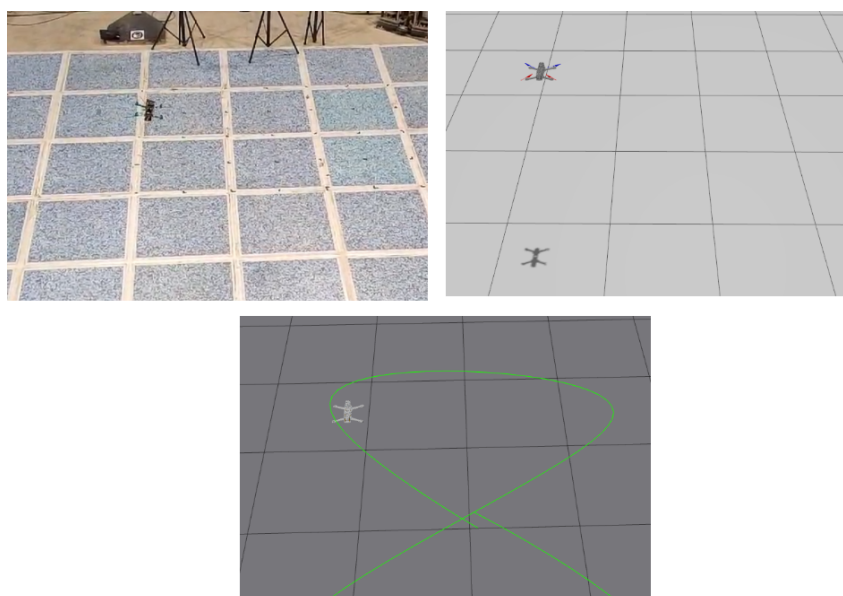


Figura 5.4: Experimento de seguimiento de trayectoria replicado en drones real (superior izquierda) y simulado (superior derecha) con la misma referencia (inferior)

Estos experimentos permiten comprobar si la metodología seguida para caracterizar el dron

resulta en un buen paso de simulación a real o 'sim-to-real'. La diferencia entre ambas trayectorias se ha medido usando un error cuadrático medio para tener un conocimiento cuantitativo de cuánto difieren el vuelo de los drones real y modelado.

5.4. Pruebas de aterrizaje en simulación

El algoritmo de aterrizaje se prueba en simulación mediante el grabado de un fichero 'rosvbag' sobre el movimiento de la plataforma móvil en 2D. Las trayectorias grabadas seguidas por la plataforma móvil se usan para mover una plataforma en el simulador Gazebo sobre la que el dron debe aterrizar.

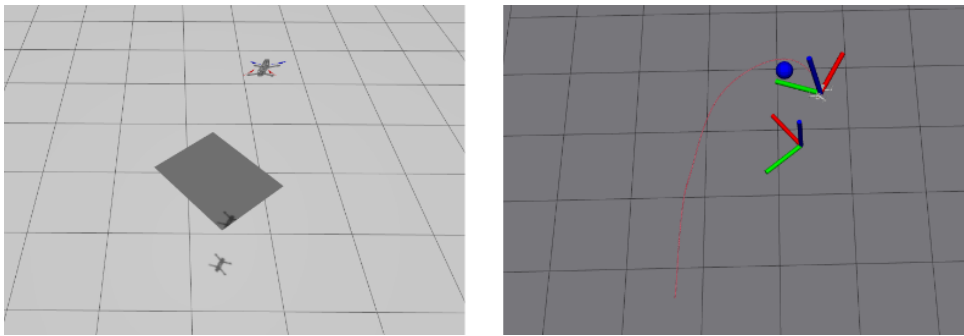


Figura 5.5: Experimento de aterrizaje en simulación con trayectoria grabada de plataforma móvil real

Se comprueba en primer lugar que el dron en simulación es capaz de aterrizar en la plataforma con un alto porcentaje de éxito, ajustando el algoritmo para ello. Los sucesivos experimentos de aterrizaje para las distintas trayectorias de la plataforma grabadas permiten determinar parámetros como el margen de distancia entre dron y plataforma para pasar de la fase de aproximación a la fase de aterrizaje.

Con el algoritmo de aterrizaje verificado, se realizan experimentos de aterrizaje en el entorno marítimo simulado en Gazebo. En este entorno se configuran distintas situaciones de intensidad de oleaje y trayectoria del barco, realizando un aterrizaje sobre el barco desplazándose en línea recta, en trayectoria curva hacia la izquierda y en curva hacia la derecha para un total de 3 experimentos.

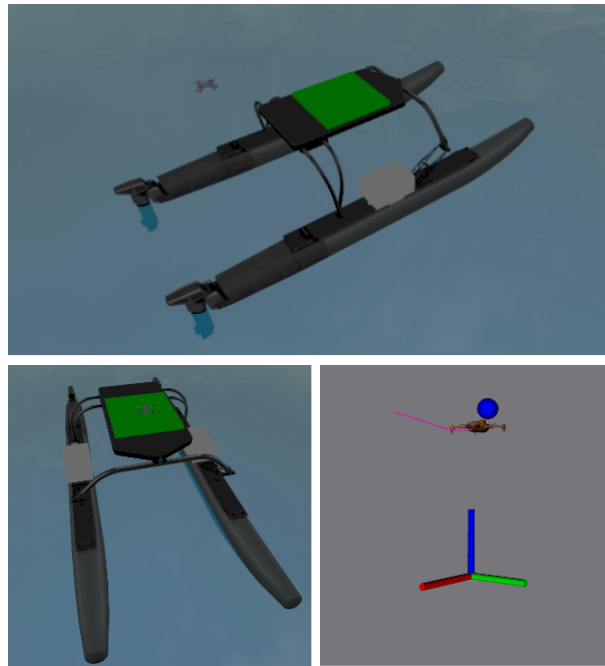


Figura 5.6: Experimento de aterrizaje en barco en entorno marítimo simulado

5.5. Vuelos en real

El algoritmo de aterrizaje afinado en simulación se prueba finalmente con el dron real en la zona de vuelo del 'CAR Robotics Arena'.

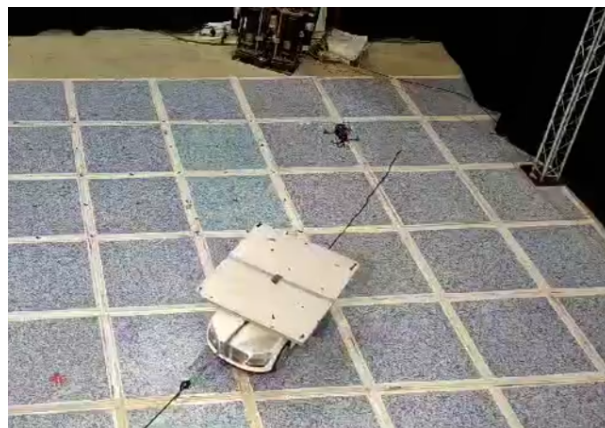


Figura 5.7: Experimento de aterrizaje real en plataforma móvil en la zona de vuelo del CAR Robotics Arena

En este último experimento, se mueve la plataforma tirando de una cuerda de la misma forma que cuando se han grabado las trayectorias que se usan para afinar el aterrizaje en simulación. Durante este movimiento, se comanda al dron un despegue y la ejecución del aterrizaje.

6. Resultados

En este capítulo se exponen los resultados obtenidos en los experimentos diseñados en el capítulo 5, obteniendo las métricas propuestas para verificar el correcto funcionamiento del sistema.

6.1. Resultados obtenidos en experimentos para inercia

Los datos obtenidos del sistema Optitrack mediante rosbag han sido representados como se muestra en la figura 6.1

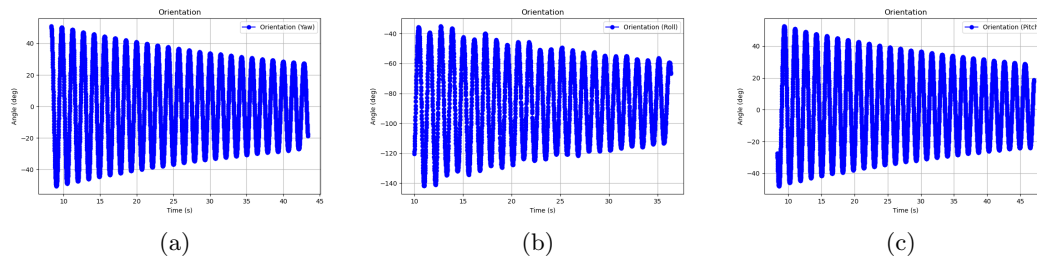


Figura 6.1: Representación de datos obtenidos de orientación del dron en a) Yaw, b) Roll, c) Pitch

Los datos se han procesado para obtener los máximos que marcan los finales e inicios de periodos de oscilación. La figura 6.2 muestra el procesado de estos datos para un experimento de giro en Yaw

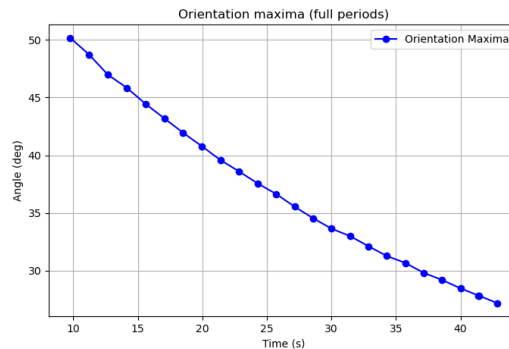


Figura 6.2: Puntos de máximo obtenidos para un experimento en el eje Z

Extrayendo exclusivamente los puntos en los que se producen los máximos y los valores de tiempo en los que se producen, se calculan los sucesivos tiempos de oscilación con los que calcular un valor medio. Para cada experimento, el valor medio de periodo se introduce en la ecuación 4.2 para calcular la inercia del dron respecto del eje del que se ha realizado el experimento. Las tablas 6.1, 6.2 y 6.3 recogen todos los datos obtenidos en los experimentos

Experiment	L(m)	d(m)	Mean Period (s)	Standard Deviation	Inertia $J_{zz}(kg/m^2)$
Experiment 1	0.77	0.185	1.43766	0.017915	0.00483
Experiment 2	0.97	0.185	1.52856	0.02965	0.00434
Experiment 3	0.62	0.185	1.25967	0.01413	0.00461

Tabla 6.1: Resultados obtenidos en los experimentos de inercia en el eje Z

Experiment	L(m)	d(m)	Mean Period (s)	Standard Deviation	Inertia $J_{xx}(kg/m^2)$
Experiment 1	0.78	0.177	1.23713	0.03899	0.00323
Experiment 2	0.895	0.177	1.17049	0.06434	0.00252
Experiment 3	0.59	0.177	1.08658	0.03251	0.00330

Tabla 6.2: Resultados obtenidos en los experimentos de inercia en el eje X

Experiment	L(m)	d(m)	Mean Period (s)	Standard Deviation	Inertia $J_{yy}(kg/m^2)$
Experiment 1	0.745	0.131	1.86017	0.05047	0.00419
Experiment 2	0.905	0.131	1.57137	0.02202	0.00378
Experiment 3	0.59	0.131	2.02438	0.04997	0.00409

Tabla 6.3: Resultados obtenidos en los experimentos de inercia en el eje Y

Los tres valores de inercia obtenidos en los experimentos de cada eje se han promediado para obtener los valores definitivos de inercia que serán introducidos en la matriz de inercia del dron. El resultados obtenido para el experimento 2 de la tabla 6.2 se ha descartado por su elevada desviación respecto del resto de valores, considerando el experimento poco representativo.

De esta manera, la matriz de inercia del dron calculada es la siguiente

$$I = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} = \begin{bmatrix} 0,0032384 & 0 & 0 \\ 0 & 0,0040254 & 0 \\ 0 & 0 & 0,00459895 \end{bmatrix}$$

6.2. Resultados obtenidos en experimentos para curva de 'thrust'

Los experimentos con el 'thrust stand' tienen como resultado hojas de datos en formato '.csv' de Excel como la que se muestra en la Figura 6.3

	A	B	C	D	E	F
1	Time (s)	ESC signal (μ s)	Torque (N·m)	Thrust (N)	Voltage (V)	Motor Speed (RPM)
2	7.8340115	1149.2842	0.012272878	-0.134200766	23.09297342	1820
3	8.511602	1171.7078	0.013055639	-0.471241986	23.07823133	2863
4	9.158186	1193.157883	0.017070629	-0.744535076	23.05883121	3387
5	9.852141	1216.36195	0.021430381	-1.041602338	23.03424635	3839
6	10.613316	1241.708742	0.026214492	-1.376239674	23.00282955	4293
7	11.25922975	1263.223758	0.030293514	-1.666226725	22.97136383	4650
8	11.893157	1284.186375	0.034197652	-1.94710852	22.93946381	4929
9	12.69237325	1310.776475	0.039126144	-2.309265057	22.89882097	3469
10	13.358035	1333.099342	0.043319922	-2.602772521	22.86235371	2725
11	14.068768	1356.922667	0.047593341	-2.919180504	22.82037964	2860

Figura 6.3: Datos recogidos en los experimentos de thrust stand para el cálculo de curvas de thrust

De entre todos los datos recogidos, se han usado los campos de 'Thrust' y 'Throttle' para realizar las curvas de 'thrust', que han sido ajustadas como polinomios de grado 2. Al añadir la dependencia con la batería 'Voltage', se logra una superficie (ver Figura 6.4) que queda reflejada en el 'thrust map' que usará la plataforma para enviar comandos precisos.

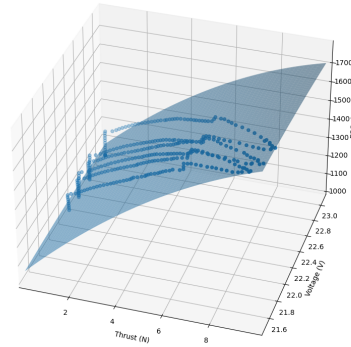


Figura 6.4: Superficie generada por curvas de thrust a distintos niveles de batería

Del ajuste de estas curvas a los datos obtenidos se consiguen los siguientes coeficientes del polinomio que se programa al configurar la plataforma Betaflight de Aerostack2:

Coefficiente	Valor
a	2891.764
b	115.683
c	-136.643
d	-2.794
e	-0.762
f	2.473

Tabla 6.4: Coeficientes de ajuste del mapa de 'thrust' para la interfaz de plataforma

Por otro lado, el sensor óptico permite el cálculo de la velocidad de giro del motor para cada comando de 'Throttle'. Las ecuaciones 3.12 y 3.13 que relaciona directamente 'Thrust' y 'Torque' y velocidad de giro de los motores permiten obtener los coeficientes aerodinámicos necesarios para completar la descripción de los motores en el simulador Gazebo. Estos valores de velocidad angular permiten también el cálculo de la constante de subida del motor que caracteriza su descripción como sistema de primer orden (ver 3.11). Los valores máximos de velocidad de giro y 'thrust' de los motores completan la lista de parámetros caracterizados. Todos los valores obtenidos quedan reflejados en la tabla 6.5 a continuación:

Parámetro	Valor
k_f	$1,51275e - 04$
k_t	$1,66935e - 05$
τ	0,1504
ω_{max} (rad/s)	549.779
T_{max} (N)	44.0

Tabla 6.5: Coeficientes aerodinámicos y parámetros obtenidos de los motores

6.3. Comparación de vuelo de dron real y en simulación

Los experimentos de comparación en los vuelos de dron real y simulado para idénticas referencias han resultado en las trayectorias que se muestran en la Figura 6.5.

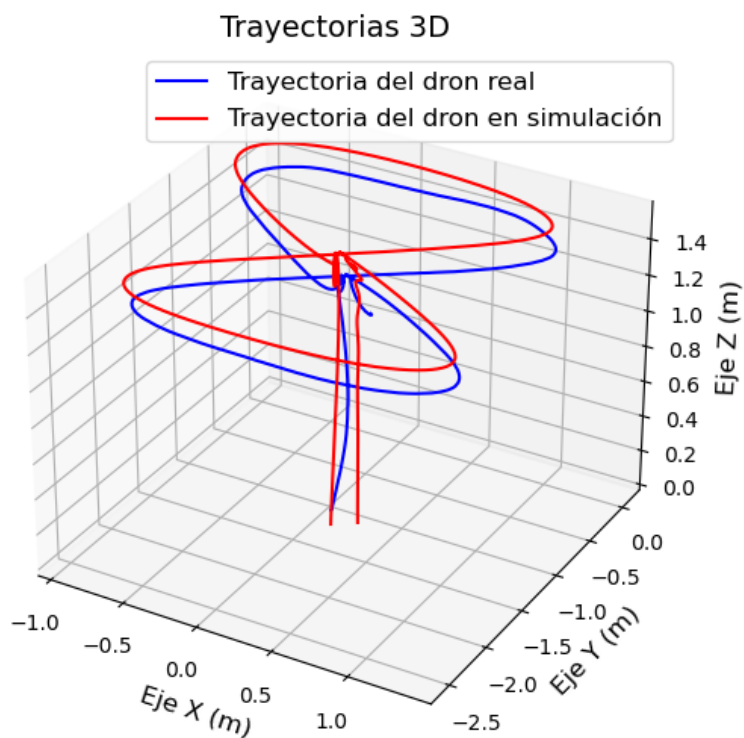


Figura 6.5: Trayectorias seguidas por el dron real (azul) y el dron simulado (rojo)

Las gráficas de las trayectorias seguidas en el plano XY exclusivamente y en altura de la Figura 6.6 permiten visualizar mejor el comportamiento de cada dron.

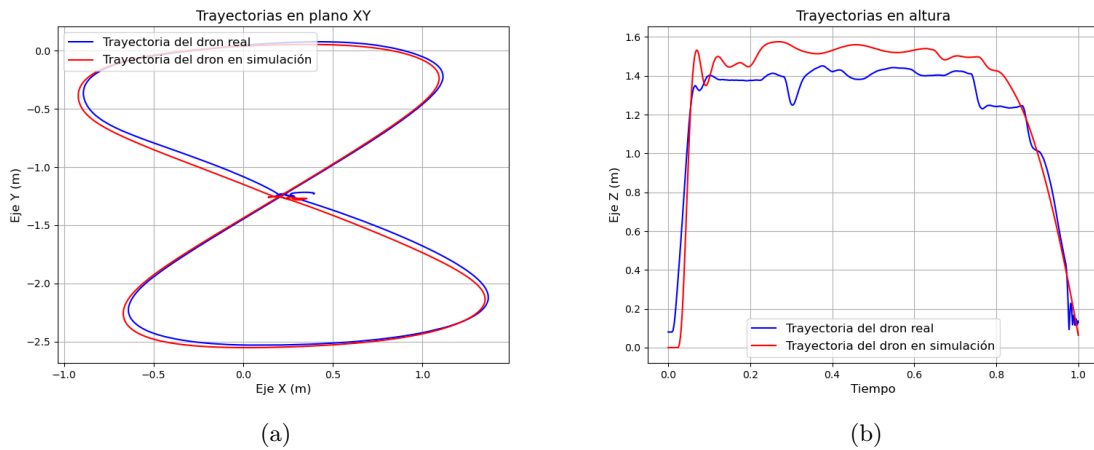


Figura 6.6: Comparación de las trayectorias seguidas por los drones real y simulado en a) plano XY, b) altura

Respecto a las trayectorias de la Figura 6.6 se ha calculado el error cuadrático medio MSE que mide la cercanía entre ambas curvas. Los resultados obtenidos de MSE para los tres experimentos se muestran en la Tabla ??

Experimento	MSE en plano XY	MSE en altura
Experimento 1	0,01499	0.08505
Experimento 2	0,01283	0.08472
Experimento 3	0,01423	0.09012

Tabla 6.6: Error cuadrático medio entre las trayectorias del dron real y del simulado en los experimentos de seguimiento y comparación

6.4. Resultados de aterrizaje en simulación

Se diferencian los experimentos dirigidos al refinamiento del algoritmo de aterrizaje en un entorno sencillo de los realizados en entorno marítimo para probar la robustez del sistema.

6.4.1. Ajuste del algoritmo en entorno simple

Los experimentos que se han realizado para afinar el algoritmo han resultado en las siguientes trayectorias del dron de Gazebo

Trayectoria de aterrizaje en 3D

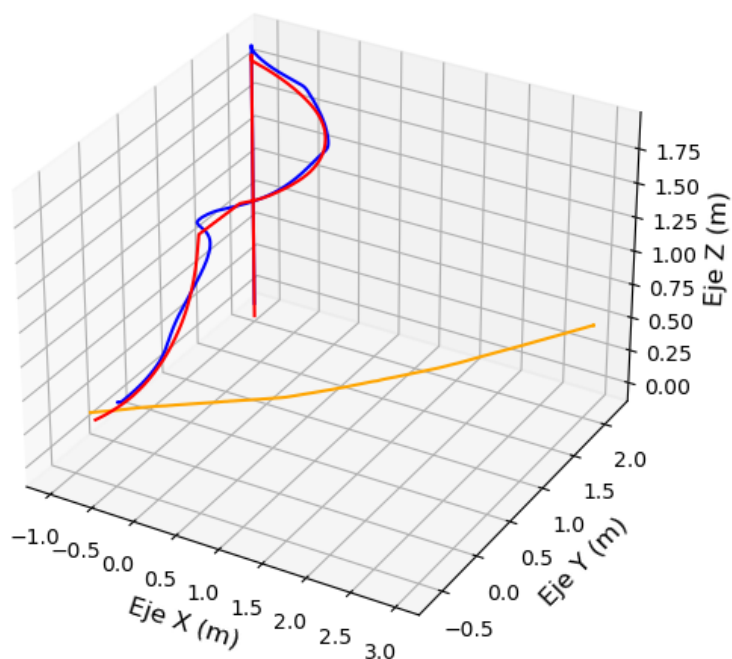


Figura 6.7: Trayectoria 3D seguida por el dron simulado (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en plataforma móvil simulada (naranja)

La gráfica de la Figura 6.8 muestra la distancia entre el dron y la plataforma móvil en todo momento, permitiendo un mejor análisis de la maniobra realizada por el dron.

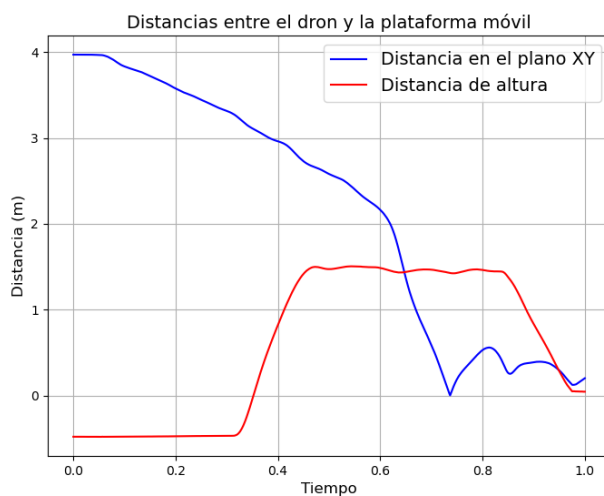


Figura 6.8: Evolución de la distancia entre el dron simulado y la plataforma móvil a lo largo de la simulación

6.4.2. Aterrizaje en entorno marítimo simulado

El algoritmo de aterrizaje diseñado y ajustado ha sido probado en un entorno con oleaje que dificulta el aterrizaje, obteniéndose las siguientes trayectorias

Trayectorias 3D en el aterrizaje en entorno marítimo

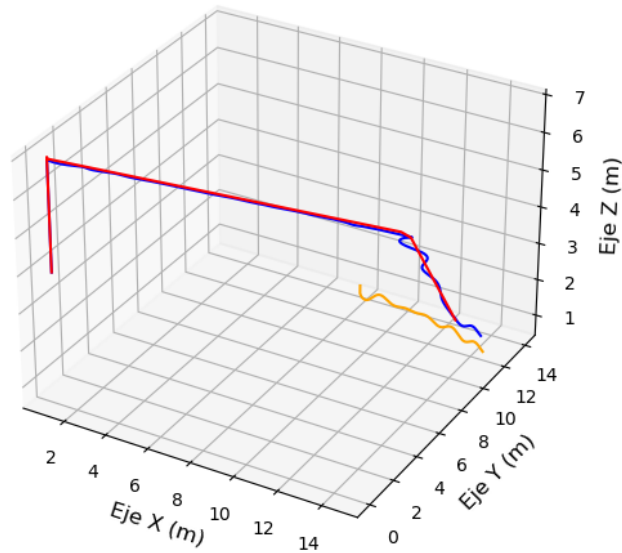


Figura 6.9: Trayectoria 3D seguida por el dron simulado (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en barco (naranja) en simulación

De nuevo, la evolución de las distancias entre el dron y el barco añade información acerca de la operación llevada a cabo y sus fases

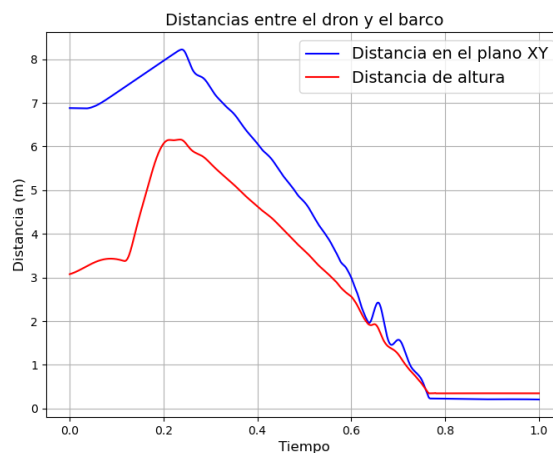


Figura 6.10: Evolución de las distancias entre el dron y el barco a lo largo de la simulación

6.5. Resultados de aterrizaje real

Finalmente, con el algoritmo probado en todos los entornos diseñados, se ha probado el conjunto del sistema en un experimento de aterrizaje real sobre plataforma móvil. La trayectoria seguida por el dron real en este último experimento se muestra en la Figura 6.11

Trayectorias 3D en el aterrizaje real

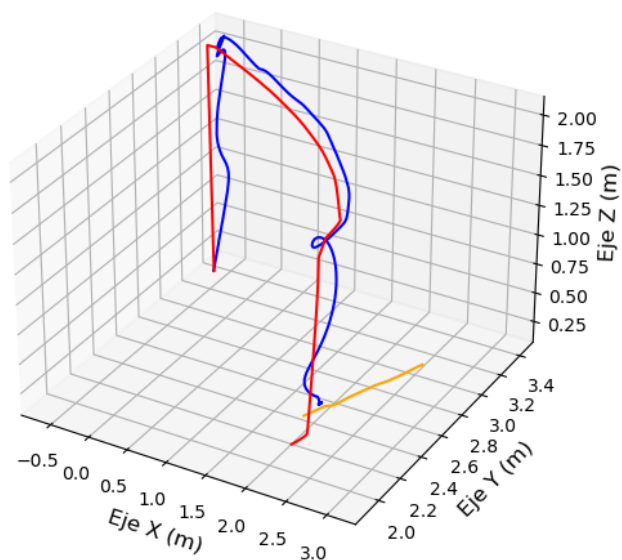


Figura 6.11: Trayectoria 3D seguida por el dron real (azul) y referencia de trayectoria (rojo) para lograr el aterrizaje en plataforma móvil real (naranja)

También en esta ocasión se usan las distancias entre dron y plataforma para validar el aterrizaje realizado

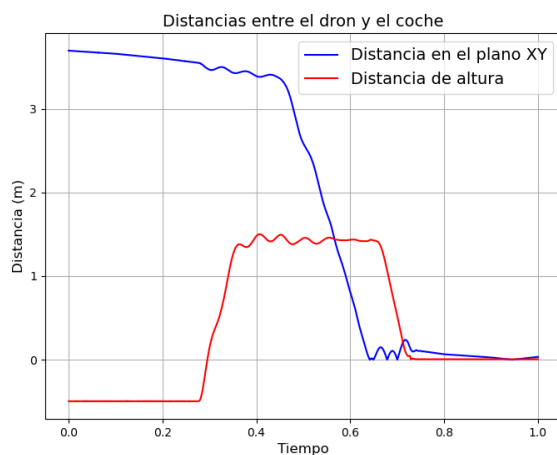


Figura 6.12: Evolución de las distancias entre el dron real y la plataforma móvil a lo largo del experimento

7. Discusión

En este capítulo se realiza un análisis de los resultados obtenidos y se describen los motivos que pueden haber dado lugar a ellos.

Sobre la caracterización de la inercia del dron, el péndulo bifilar ha demostrado ser un sistema sencillo y extremadamente económico para la estimación de este parámetro. En la mayoría de casos da medidas consistentes con escasa desviación, y el refuerzo del sistema Optitrack mejora un cálculo que, sin él, tendría que haberse basado en una única división del número total de oscilaciones en un periodo más extendido de tiempo.

Los experimentos con 'thrust stand' también han resultado muy efectivos. El dispositivo ha permitido el cálculo de todos los parámetros propuestos y su montaje y uso resultan accesibles. Especialmente en el vuelo del dron real, el mapa de thrust ha mejorado mucho la estabilidad del dron, que exhibe un vuelo sin apenas perturbaciones.

Estos resultados se ven reforzados por los experimentos de comparación de vuelo entre los drones real y simulados. El seguimiento que hace cada dron de la trayectoria es muy similar como se puede ver en la Figura 6.5. Especialmente si se atiende a la gráfica de trayectorias en XY, la diferencia es apenas perceptible. Otro resultado positivo que se puede apreciar en la misma figura es que el dron real no pierde altura a lo largo del experimento. Queda de manifiesto que el cálculo de thrust map con datos de batería resulta en un vuelo que no se ve afectado por las pérdidas de batería.

Al margen de los resultados anteriores, es notable la diferencia en altura en la Figura 6.5, donde el dron real parece tener problemas para alcanzar la referencia en el eje Z en régimen permanente.

Los experimentos de thrust stand se realizaron con un variador de menor potencia que el que tiene instalado el dron. Esto quiere decir que para los rangos más altos de potencia de los motores, los valores obtenidos para el empuje pueden ser menos cercanos a la fuerza real de los motores. Si bien el régimen de vuelo en el seguimiento de estas trayectorias no es muy exigente, el uso de un variador distinto puede afectar al vuelo del dron y al empuje que alcanza con las instrucciones del controlador.

El resto de imprecisiones pueden deberse a otros factores. Por parte de las medidas de inercia, como se puede ver en la tabla 6.2, la medida de la inercia respecto al eje X ha resultado ser ligeramente más irregular. El correcto funcionamiento del método depende en gran medida de lo bien equilibrado que esté el dron, es decir, si está totalmente paralelo al eje del que se quiere medir la inercia. El eje X es el eje respecto del cual ha sido más complicado encontrar ese equilibrio, no pudiendo lograrlo totalmente. Esto ha podido afectar a la precisión de las medidas. Otro factor que ha podido afectar al péndulo es la selección de un ángulo inicial de giro demasiado grande.

Hay que tener en cuenta que no se han caracterizado efectos como el drag de los motores o el drag aerodinámico del dron. Estos efectos requieren de técnicas o maquinaria más compleja como el CFD o túneles de viento, de los que no se ha dispuesto para este trabajo y que han sido despreciados. Además, sigue existiendo una gran diferencia entre los entornos simulado y real. En simulación no existen muchas de las fuerzas aerodinámicas que se oponen al movimiento del dron, ni existe el ruido, que es una de las principales fuentes de discrepancia entre el vuelo de los drones modelado y real. La inclusión de efectos aerodinámicos en simulación y su caracterización acercaría aún más ambos modelos de dron.

La caracterización del dron ha permitido también el ajuste de su control usando la simulación pa-

ra lograr una muy buena aproximación que ha podido ser refinada volando el dron real. El llevar ese ajuste definitivo de vuelta al dron de simulación, se ha comprobado que el comportamiento prácticamente coincide, como se puede comprobar una vez más en la Figura 6.5.

Un procedimiento muy similar se ha seguido para conseguir el aterrizaje del dron. La reproducción del movimiento de la plataforma real en el entorno de simulación ha permitido realizar múltiples pruebas de ajuste del algoritmo sin poner en riesgo el dron. Como resultado, se consigue el aterrizaje en entorno marítimo como se ve en la Figura 6.9. Resulta llamativo cómo el vuelo del dron en el entorno marítimo completamente simulado es más estable que el vuelo del dron simulado con la información de posición de la plataforma real de la Figura 6.7. El movimiento del barco es más irregular, pero la localización ruidosa de la plataforma real afecta más al vuelo del dron. Se aprecia también que el momento de cambio entre la fase de aproximación y aterrizaje en el aterrizaje de la Figura 6.7 tiene un efecto más negativo en el posicionamiento del dron, que tras alcanzar distancia 0 en el plano XY y situarse encima de la plataforma vuelve a alejarse de ella dos veces. En el aterrizaje sobre el barco solo vuelve a alejarse del punto de aterrizaje cuando el oleaje hace que el barco baje (ver Figura 6.10).

El intento de aterrizaje real en plataforma móvil ha resultado en éxito como muestra la figura 6.11. En la trayectoria seguida por el dron se aprecian las dificultades del vuelo real, que se aleja más de la referencia de trayectoria y muestra un mayor rizado por el ruido como se ve en la Figura 6.12. Aún así, el dron consigue el aterrizaje con una muy buena aproximación y sin apenas alejarse de la distancia 0 a la plataforma en la fase de aterrizaje.

8. Conclusión

Este capítulo expone las conclusiones extraídas del desarrollo y los resultados obtenidos. Estas conclusiones motivan la propuesta de líneas futuras de trabajo.

8.1. Conclusiones

En este trabajo se parte del deseo de lograr el aterrizaje autónomo de un dron cuadricóptero. En primer lugar, se realiza un estudio de las técnicas más usadas en la caracterización de drones, así como de los parámetros que se identifican con estas. De entre todas estas técnicas se seleccionan las que mejor se ajustan al alcance de este trabajo. Se concluye que la caracterización de la matriz de inercia del dron junto con los parámetros y coeficientes más importantes de los motores mejora el control de vuelo autónomo del dron.

Para la identificación se propone una metodología de caracterización basada en experimentos de péndulo bifilar y banco de pruebas 'thrust stand'. Se detallan los montajes y se exponen los experimentos con los que se calculan los valores deseados, cumpliendo los objetivos en este apartado.

Dentro de la metodología, se propone también la implementación de un modelo de dron en simulación usando Gazebo, que aprovecha los valores calculados con la caracterización. Se logra un vuelo realista en distintos entornos simulados como se ha podido comprobar en los resultados de los experimentos que comparan vuelos en simulación y reales, llegando a la conclusión que la metodología de caracterización propuesta es válida y efectiva. En los experimentos de vuelo reales se comprueba que el dron no pierde altura a lo largo de los vuelos. Esto refuerza la conclusión de que el thrust map calculado considerando datos de batería es bueno.

El control y la maniobra de aterrizaje se han llevado a otros entornos para probar la robustez del sistema y la misión, como el entorno marítimo de Gazebo y un entorno de vuelo real. En ambos casos los experimentos han resultado exitosos, validando la solución y concluyéndose que la mejora del control gracias a la caracterización es un método válido para la consecución del aterrizaje autónomo.

En definitiva, se han cumplido todos los objetivos parciales planteados, y su consecución ha permitido lograr el aterrizaje autónomo de un dron real en una plataforma móvil, completando así el objetivo principal de este trabajo. La metodología propuesta consigue mejoras en el control del vuelo de drones, consiguiendo maniobras más estables más allá del algoritmo utilizado.

8.2. Trabajo Futuro

Como trabajo futuro queda una verificación más exhaustiva de la metodología propuesta. Para ello se propone la realización de más pruebas y experimentos, tanto en el entorno controlado real como en la simulación marítimo, para finalmente llevar los experimentos reales a un entorno más cercano al marítimo simulado.

Por otro lado, se propone el refinamiento de la metodología de caracterización empleando métodos de CFD o túneles de viento. Estos métodos, además de afinar la caracterización con medidas precisas, permiten ampliarla incluyendo la identificación de otros efecto aerodinámicos que actúan sobre el dron.

Otra línea de trabajo en este sentido es la caracterización en tiempo de vuelo mediante métodos de identificación en línea, que permite realizar un control adaptativo que no se ciñe solo a la configuración inicial del dron.

Por último, se propone el trabajo en un algoritmo de aterrizaje más complejo y que se anticipe de mejor manera a la dinámica de la plataforma de aterrizaje, realizando predicciones sobre su movimiento y adaptándose mejor a él.

Referencias

- [1] “Aerostack2 documentation,” <https://aerostack2.github.io/index.html>, accessed: 24-01-2025.
- [2] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, “A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight,” *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.
- [3] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE ROBOTICS AND AUTOMATION LETTERS*, vol. 3, no. 2, pp. 620–626, APR 2018.
- [4] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart, “Model predictive control for micro aerial vehicles: A survey,” in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1556–1563.
- [5] A. Romero, R. Penicka, and D. Scaramuzza, “Time-optimal online replanning for agile quadrotor flight,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7730–7737, 2022.
- [6] A. K. Tripathi, V. V. Patel, and R. Padhi, “Autonomous landing design of uavs using feedback linearization controller with anti windup scheme,” *IFAC-PapersOnLine*, vol. 53, no. 1, pp. 81–86, 2020, 6th Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320300318>
- [7] Y. Zhou and K. Takaba, “Optimal landing control of an unmanned aerial vehicle via partial feedback linearization,” in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 2019, pp. 218–223.
- [8] J. W. Tweedale, “Fuzzy control loop in an autonomous landing system for unmanned air vehicles,” in *2012 IEEE International Conference on Fuzzy Systems*. IEEE, 2012, pp. 1–8.
- [9] I. Tsitses, P. Zacharia, E. Xidias, and M. Papoutsidakis, “A fuzzy-based system for autonomous unmanned aerial vehicle ship deck landing,” *Sensors*, vol. 24, no. 2, p. 680, 2024.
- [10] W. Li, Y. Ge, Z. Guan, H. Gao, and H. Feng, “Nmpc-based uav-usv cooperative tracking and landing,” *Journal of the Franklin Institute*, vol. 360, no. 11, pp. 7481–7500, 2023.
- [11] D. Chen, L. Xu, and C. Wang, “An advanced control method for aircraft carrier landing of uav based on capf-nmpc,” *Aerospace*, vol. 11, no. 8, p. 656, 2024.
- [12] P. M. Gupta, E. Pairet, T. Nascimento, and M. Saska, “Landing a uav in harsh winds and turbulent open waters,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 744–751, 2022.
- [13] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, “Reaching the limit in autonomous racing: Optimal control versus reinforcement learning,” *Science Robotics*, vol. 8, no. 82, p. eadg1462, 2023.
- [14] J. Amendola, L. R. Cenkeramaddi, and A. Jha, “Drone landing and reinforcement learning: State-of-art, challenges and opportunities,” *IEEE Open Journal of Intelligent Transportation Systems*, 2024.

-
- [15] P. Goldschmid and A. Ahmad, "Reinforcement learning based autonomous multi-rotor landing on moving platforms," *Autonomous Robots*, vol. 48, no. 4, pp. 1–21, 2024.
- [16] K. Backman, D. Kulić, and H. Chung, "Reinforcement learning for shared autonomy drone landings," *Autonomous Robots*, vol. 47, no. 8, pp. 1419–1438, 2023.
- [17] J. E. Kooi and R. Babuška, "Inclined quadrotor landing using deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2361–2368.
- [18] R. Munguía, S. Urzua, and A. Grau, "EKF-based parameter identification of multi-rotor unmanned aerial vehicles models," *Sensors*, vol. 19, no. 19, p. 4174, 2019.
- [19] Y. Chen, Y. Zeng, H. Li, J. Zhang, and L. Zhang, "Research on the measurement technology of rotational inertia of rigid body based on the principles of monocular vision and torsion pendulum," *Sensors*, vol. 23, no. 10, p. 4787, 2023.
- [20] S. Krishnaraj, R. S. Kumar, A. G. Krishnan, G. Ganeshkumar, M. Mohan, and M. Nirmal, "Aerodynamic analysis of hybrid drone," in *IOP Conference Series: Materials Science and Engineering*, vol. 1012, no. 1. IOP Publishing, 2021, p. 012023.
- [21] A. Mathur and E. Atkins, "Wind tunnel testing and aerodynamic characterization of a quadplane uncrewed aircraft system," *arXiv preprint arXiv:2301.12316*, 2023.
- [22] N. Elmeseiry, N. Alshaer, and T. Ismail, "A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications," *Aerospace*, vol. 8, no. 12, p. 363, 2021.
- [23] R. Noda, T. Ikeda, T. Nakata, and H. Liu, "Characterization of the low-noise drone propeller with serrated gurney flap," *Frontiers in Aerospace Engineering*, vol. 1, p. 1004828, 2022.
- [24] T. Setati, N. Botha, and J. M. Roux, "Experimental approach to calculate the moments of inertia of a hexacopter unmanned aerial vehicle," in *MATEC Web of Conferences*, vol. 370. EDP Sciences, 2022, p. 05001.
- [25] Y. M. Al-Rawashdeh, M. Elshafei, and H. Ouakad, "Mass and inertia estimation using all-accelerometer," in *2023 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2023, pp. 1–5.
- [26] K. Zhu, J. Yang, and X. Liu, "An online inertia identification based disturbance observer for attitude loop of quadrotor system," in *2022 IEEE 17th International Conference on Control & Automation (ICCA)*. IEEE, 2022, pp. 547–552.
- [27] A. Noormohammadi-Asl, O. Esrafilian, M. Ahangar Arzati, and H. D. Taghirad, "System identification and h-based control of quadrotor attitude," *Mechanical Systems and Signal Processing*, vol. 135, p. 106358, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327019305795>
- [28] Y. Sheng and G. Tao, "System characterization and adaptive tracking control of quadrotors under multiple operating conditions," *Guidance, Navigation and Control*, vol. 1, no. 02, p. 2150006, 2021.
- [29] P. Outeiro, C. Cardeira, and P. Oliveira, "Multiple-model control architecture for a quadrotor with constant unknown mass and inertia," *Mechatronics*, vol. 73, p. 102455, 2021.
- [30] F. Dong, B. Yuan, X. Zhao, Z. Ding, and S. Chen, "Adaptive robust constraint-following control for morphing quadrotor uav with uncertainty: A segmented modeling approach," *Journal of the Franklin Institute*, vol. 361, no. 5, p. 106678, 2024.

- [31] R. Gill and R. D’Andrea, “Propeller thrust and drag in forward flight,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 73–79.
- [32] M. N. Waqas Khan, “A propeller model for general forward flight conditions,” *International Journal of Intelligent Unmanned Systems*, vol. 3, no. 2,3, pp. 72–92, 2015.
- [33] Z. Czyż, P. Karpiński, K. Skiba, and M. Wendeker, “Wind tunnel performance tests of the propellers with different pitch for the electric propulsion system,” *Sensors*, vol. 22, no. 1, p. 2, 2021.
- [34] P. Polivanov and A. Sidorenko, “Aerodynamic characteristics of a quadcopter with propellers,” in *AIP Conference Proceedings*, vol. 2351, no. 1. AIP Publishing, 2021.
- [35] Z. Siddiqi and J. W. Lee, “A computational and experimental study on aerodynamics of motor-driven propellers using thrust stand and rotating cup anemometer,” *Progress in Computational Fluid Dynamics, an International Journal*, vol. 22, no. 1, pp. 23–36, 2022.
- [36] M. Kišev, L. Vacho, L. Tóth, M. Olejár, M. Harničárová, J. Valíček, H. Tozan *et al.*, “Experimental measurement of a uav propeller’s thrust,” *Tehnički vjesnik*, vol. 29, no. 1, pp. 73–80, 2022.
- [37] “Multicopter pid tuning guide (manual/advanced),” https://docs.px4.io/main/en/config-mc/pid_tuning_guide_multicopter.html, accessed: 19-01-2025.
- [38] T. Oktay and Y. Eraslan, “Numerical investigation of effects of airspeed and rotational speed on quadrotor uav propeller thrust coefficient,” *Journal of Aviation*, vol. 5, no. 1, pp. 9–15, 2021.
- [39] D. Santos, M. Saska, and T. Nascimento, “A generalized thrust estimation and control approach for multirotors micro aerial vehicles,” *IEEE Robotics and Automation Letters*, 2024.
- [40] C. A. Dimmig, G. Silano, K. McGuire, C. Gabellieri, W. Hšnig, J. Moore, and M. Kobilarov, “Survey of simulators for aerial robots: An overview and in-depth systematic comparisons,” *IEEE Robotics & Automation Magazine*, 2024.
- [41] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, “Flightmare: A flexible quadrotor simulator,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1147–1157.
- [42] L. Turco, J. Zhao, Y. Xu, and A. Tsourdos, “A study on co-simulation digital twin with matlab and airsim for future advanced air mobility,” in *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–18.
- [43] “Gazebo documentation,” <https://gazebosim.org/features>, accessed: 25-01-2025.
- [44] A. V. NC, A. R. Yadav, D. Mehta, J. Belani, and R. R. Chauhan, “A guide to novice for proper selection of the components of drone for specific applications,” *Materials Today: Proceedings*, vol. 65, pp. 3617–3622, 2022.
- [45] M. Fernandez-Cortizas, M. Molina, P. Arias-Perez, R. Perez-Segui, D. Perez-Saura, and P. Campoy, “Aerostack2: A software framework for developing multi-robot aerial systems,” *arXiv preprint arXiv:2303.18237*, 2023.
- [46] “Optitrack for robotics,” <https://www.optitrack.com/applications/robotics/>, accessed: 20-01-2025.

ANEXOS

A. Planificación y Presupuesto

Se realiza un desglose del tiempo y los recursos empleados en este trabajo para facilitar su reproducción posterior.

A.1. Planificación

Este trabajo se inició a finales de octubre de 2024 y ha finalizado en enero del 2025, extendiéndose por unos 3 meses aproximadamente. Durante este tiempo, la dedicación ha sido de 30 horas semanales, que multiplicadas por las 12 semanas de duración dan un total de unas 360 horas. Este tiempo se corresponde con el número de horas previstas para un Trabajo de Fin de Máster con valor de 12 créditos ECTS.

El desarrollo ha tenido lugar en el seno del grupo de investigación CVAR (Computer Vision and Aerial Robotics Group) del departamento de Electrónica, Automática e Informática Industrial de la Escuela Técnica Superior de Ingenieros industriales (ETSII) de la Universidad Politécnica de Madrid (UPM).

Las primeras semanas se centraron en el estudio previo de las necesidades y requisitos del trabajo y de los métodos actuales que permitirían su consecución. El mes de noviembre se centró en la realización de los experimentos que caracterizan al dron y la verificación de la metodología con vuelos comparativos. El mes de diciembre se centró en el diseño del aterrizaje y los experimentos que demuestran su funcionamiento. Finalmente, el mes de enero se dedicó a la redacción de la memoria del proyecto.

El diagrama EDP de la Figura A.1 muestra un desglose de la organización del proyecto. Los tiempos más detallados de trabajo en cada bloque y tarea quedan reflejados en el diagrama GANTT de la Figura A.2

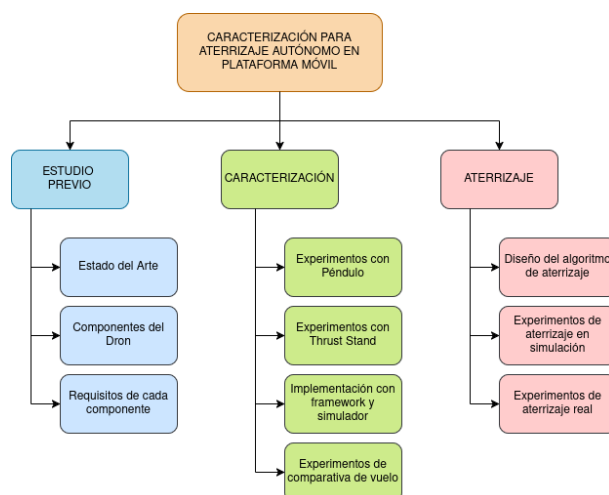


Figura A.1: Diagrama EDP de la organización de las distintas fases del trabajo

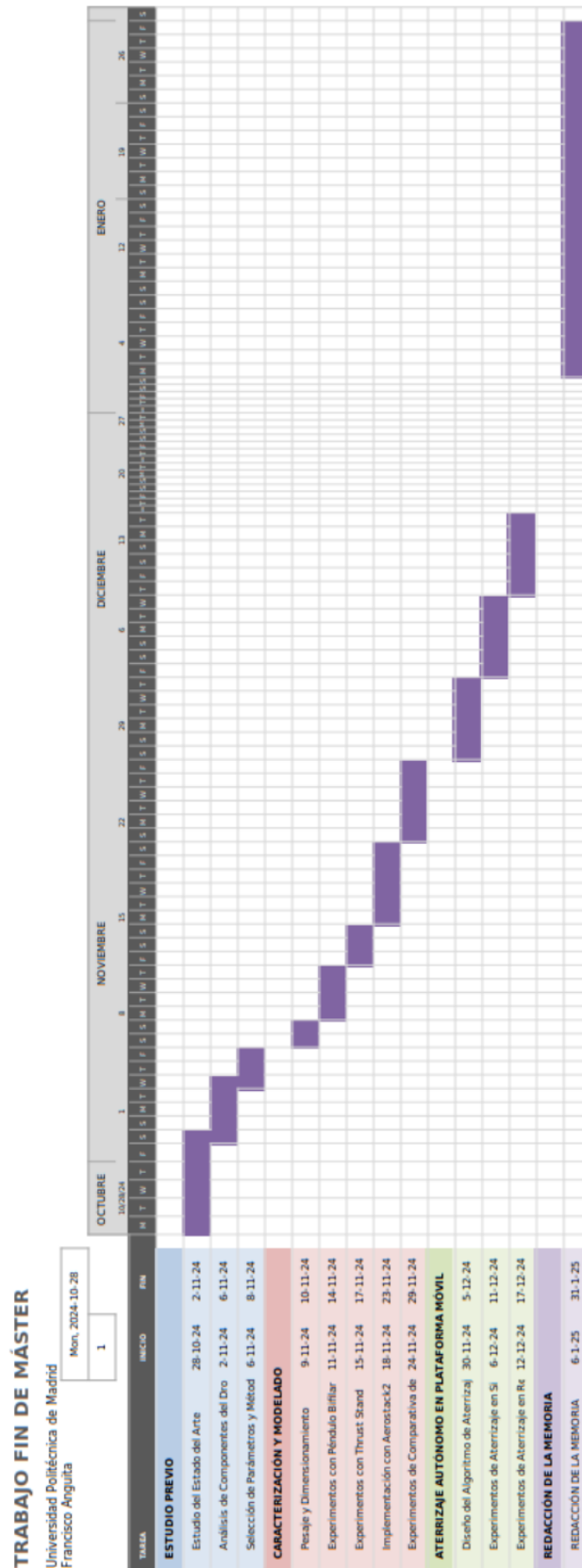


Figura A.2: Diagrama GANTT con las tareas realizadas, fechas de inicio y fin de cada una y tiempo empleado en ellas

A.2. Presupuesto

El presupuesto de este trabajo se divide en coste de recursos humanos y coste de los equipos utilizados.

Las 360 horas de trabajo dedicadas por el alumno se multiplican por el salario de un ayudante investigador a jornada completa en la universidad, que es de unos 8.33 euros la hora (1250 euros mensuales). Según el portal de transparencia de la UPM, el salario por horas del tutor es de 33.72 euros la hora. La Tabla A.1 recoge el coste total en recursos humanos

Elemento	Horas	Coste por horas (EUR)	Coste total (EUR)
Alumno	360	8.33	2998.8
Tutor	30	33.72	1011.6
Total			4010.4

Tabla A.1: Costes de los recursos humanos empleados en el proyecto

Entre los equipos necesarios para la realización del trabajo se cuentan el dron con todos sus componentes y el ordenador con el que se ha realizado el desarrollo. La tabla A.2 reúne el coste de todos estos elementos

Componente	Unidades	Coste unitario (EUR)	Coste total (EUR)
Ordenador portátil	1	1260	1260
Frame ZMR250	1	22.50	22.50
Motor XNOVA Black Thunder	4	29.90	119.60
ESC Foxeer Reaper F4	1	81.93	81.93
FCU Foxeer H7 MPU6000	1	101.23	101.23
Receptor IRC Ghost Atto	1	29.95	29.95
Ordenador Jetson Orin NX	1	674.60	674.60
Disipador de calor	1	19.21	19.21
BEC Matek BEC125-PRO	1	21.90	21.90
SSD Sabrent 1TB Rocket	1	119.99	119.99
Adaptador y Antena Wifi	1	29.75	29.75
Batería Lipo 6S Tattu R-Line	5	39.56	158.24
Cargador Lipo	1	109.90	109.90
Mando RC Radiomaster Zorro	1	115.80	115.80
Transmisor Ghost Module	1	69.95	69.95
Elementos de ensamblaje	-	-	30.00
Total			2965.11

Tabla A.2: Costes de los equipos empleados en el proyecto

El coste total del proyecto es

Concepto	Coste total (EUR)
Recursos Humanos	4010.40
Equipos y Componentes	2965.11
Total	6975.51

Tabla A.3: Costes total del desarrollo del proyecto

B. Impacto social, impacto medioambiental y Agenda 2030

Se realiza un análisis del impacto de este proyecto más allá de lo técnico para asegurar el cumplimiento de la normativa vigente, la no vulneración de ningún derecho y la contribución al desarrollo sostenible.

B.1. Impacto social, aspectos éticos y legales

La creciente popularidad de los pequeños y medianos UAV ha acercado estos dispositivos a la gente más que nunca. En estos casos es importante considerar cuáles pueden ser las consecuencias de la proliferación de estos sistemas robóticos aéreos.

La principal preocupación reside en la vulneración de la privacidad de los individuos. Los drones acostumbra a volar con cámaras equipadas que pueden obtener imágenes y grabar vídeos. Se debe asegurar en todo momento el cumplimiento de la Ley de Protección de datos y el Derecho al Honor e Intimidad de las personas.

También en materia legal, en España la Agencia Estatal de Seguridad Aérea (AESA) se encarga de regular el uso de drones. El vuelo de aeronaves no tripuladas está sujeto al Reglamento Europeo RE 2019/947 y al Real Decreto 2019/945.

El dron con el que se trabaja en este proyecto tiene una masa en el momento del despegue de 862 gramos, siendo clasificado como clase ANT C1. Esta clasificación limita los niveles máximos de potencia sonora, velocidad y altura máximas y determina los permisos necesarios para el vuelo de la aeronave.

El dron no podrá superar nunca los 120 metros de altura medidos desde el punto de despegue, ni tampoco los 19 metros por segundo de velocidad. Debe ser visible y controlable en todo momento por un piloto de seguridad a distancia. Dado el peligro que pueden suponer las aeronaves de más de 250 gramos, el dron debe ser visible y controlable en todo momento por un piloto de seguridad a distancia que, además, debe estar registrado. Asimismo, la aeronave debe realizar, en tiempo real y durante toda la duración del vuelo, una transmisión periódica su posición y altura, su número de serie, del número de registro del operador a distancia, y de la posición del piloto o del punto de despegue.

Al tratarse de un dron montado a partir de los componentes por separado y no uno comercializado con manual ni certificaciones, su vuelo en exteriores resulta complicado desde el punto de vista legal. Tampoco es recomendable desde el punto de vista técnico, ya que el sistema solo se ha probado en un entorno controlado de laboratorio y no en vuelos al aire libre.

Otra implicación importante tiene que ver con la aplicación de los UAV. Aunque los beneficios de su uso son innegables, pudiendo realizar operaciones que resultan peligrosas si las tiene que realizar un humano como rescates o inspecciones, su reciente aplicación militar puede resultar alarmante. Se debe hacer siempre un uso responsable de la tecnología que suponga siempre un avance y no un retroceso.

B.2. Impacto medioambiental

Los UAV en general y especialmente los de pequeño y mediano tamaño son sistemas en la mayoría de casos actuados por motores eléctricos. El empleo de estos motores no tiene solo una

motivación técnica sino que en algunos países, España ente ellos, su uso es obligatorio.

Los motores eléctricos no emiten gases a la atmósfera por lo que su impacto ambiental se ve reducido al desecho de las baterías que alimentan los motores. En el caso de este trabajo, las baterías Lipo pueden ser desechadas llevándolas a los distintos puntos limpios por todo el país, previamente descargadas mediante su uso normal.

B.3. Desarrollo sostenible y Agenda 2030

Se estudia cuáles son las contribuciones de este trabajo a los Objetivos de Desarrollo Sostenibles (ODS) propuestos por la Organización de las Naciones Unidas (ONU) en su Agenda 2030. Esta serie de objetivos pretenden influir en la dirección estratégica de los países para poder analizar su impacto en los objetivos globales de sostenibilidad.

La Agenda 2030 marca diecisiete objetivos, y este trabajo contribuye a cuatro de ellos. En primer lugar, al ODS número 9 de '**Industria, Innovación e Infraestructura**'. Este objetivo se centra en la innovación como medio para lograr infraestructuras más duraderas y una industria más eficaz mediante la tecnología. La caracterización de drones para conseguir maniobras más efectivas permite la aplicación de UAV a sectores más amplios de la industria y a nuevos servicios aéreos.

El ODS número 11 sobre '**Ciudades y Comunidades Sostenibles**' pretende lograr una vida mejor en las ciudades, haciendo que crezcan en inclusividad y que disminuyan su contaminación. El vuelo eficiente de drones logrado por medio de su caracterización y modelado permite la sustitución de sistemas de combustión más contaminantes por estos dispositivos de actuación eléctrica en aplicaciones como vigilancia, entrega de suministros o control ambiental.

En la misma línea se contribuye a los ODS número 12, '**Producción y consumo responsable**' y 13, '**Acción por el Clima**', que pretenden frenar el cambio climático mediante una mejor gestión de los recursos disponibles. A parte de la reducción en contaminación ya mencionada, el uso de drones en operaciones de inspección y mantenimiento de aerogeneradores o placas fotovoltaicas favorece la generación de energía por fuentes renovables. La mejora en el control de UAV que produce el trabajo de caracterización facilita su uso para estos fines.