



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Inspección de aerogeneradores con drones autónomos

Informe de Proyecto de Grado presentado por

Carolina Acosta, Bruno Centanaro y Martín Tapia

en cumplimiento parcial de los requerimientos para la graduación de la carrera de Ingeniería en
Computación de Facultad de Ingeniería de la Universidad de la República

Supervisor

Facundo Benavides

Montevideo, 19 de diciembre de 2024



Inspección de aerogeneradores con drones autónomos por Carolina Acosta, Bruno Centanaro y Martín Tapia tiene licencia CC Atribución 4.0.

Agradecimientos

Queremos expresar nuestro agradecimiento a todas las personas que contribuyeron de alguna forma al desarrollo de este proyecto.

En especial:

A nuestras familias y seres queridos, por su constante apoyo a lo largo de toda nuestra formación.

A nuestro tutor, Facundo Benavides, por siempre esforzarse en ayudarnos a alcanzar nuestras metas, como lograr la entrega de este proyecto en el plazo que nos habíamos propuesto. Su compromiso y orientación fueron fundamentales para el éxito de este trabajo.

A Analía Arimón, investigadora del InCO, por brindarnos herramientas para crear el entorno simulado en Gazebo.

A los funcionarios del parque eólico *Los Caracoles*, por brindarnos su tiempo y disposición para responder nuestras inquietudes.

Resumen

Uruguay avanza hacia una transición energética basada en fuentes renovables, cubriendo el 97 % de su demanda eléctrica con ellas en años normales de lluvia, siendo la energía eólica un 32 %. Actualmente el país cuenta con 41 parques eólicos y 1.514 MW instalados.

Las aspas de los aerogeneradores sufren desgaste por fatiga debido al viento y otros factores como erosión, lluvias, rayos, hielo, y colisiones con aves, afectando su rendimiento. Detectar defectos en las aspas a tiempo es clave para garantizar su eficiencia, prolongar su vida útil, reducir tiempos de inactividad y prevenir fallas estructurales graves.

El presente trabajo aborda el desarrollo de un sistema destinado a la inspección de las aspas de aerogeneradores mediante el uso de un dron autónomo. Este enfoque busca estandarizar el proceso de inspección, mejorar la seguridad en comparación con los métodos tradicionales y reducir significativamente los costos y tiempos asociados a dichos métodos.

La implementación del sistema se realizó utilizando el framework ROS 2 para el desarrollo de software para robots y PX4 para el control del vuelo del dron. Entre las principales tareas realizadas se destacan el posicionamiento del dron centrado con respecto al rotor y ortogonal al aerogenerador, y la planificación de la trayectoria a seguir para la inspección de las aspas. La primera tarea se resolvió mediante la aplicación de técnicas de visión por computadora, como la transformada probabilística de Hough y la integración de LiDAR (Light Detection and Ranging), para la detección y localización de los componentes del aerogenerador. Por su parte, la planificación de trayectorias se abordó como una instancia del Problema del Viajante Asimétrico (Asymmetric TSP), lo que permitió minimizar el tiempo de inspección y optimizar el uso de recursos. Además, se desarrolló un módulo de corrección que ajusta la posición del dron en tiempo real durante la inspección para asegurar que las aspas se mantuvieran dentro de la imagen capturada. Durante el proceso de inspección, el sistema captura fotografías en intervalos de distancia recorrida y en puntos clave, como las puntas de las aspas, para documentar el estado de las aspas.

La experimentación se realizó en un entorno simulado utilizando Gazebo, evaluando el sistema bajo diversas condiciones de viento y escenarios de posicionamiento inicial del dron, lo cual permitió validar su adaptabilidad. Los resultados obtenidos evidencian que un sistema autónomo de estas características puede reducir significativamente el tiempo de inspección, contribuyendo a la eficiencia en la gestión en parques eólicos.

Palabras clave: Aerogenerador, Inspección de aspas, Inspección automatizada, Planificación de trayectorias, dron, UAV, ROS 2, Gazebo, PX4

Índice general

1. Introducción	1
2. Marco de Trabajo	4
2.1. Revisión de Antecedentes	4
2.1.1. Preprocesamiento de las Imágenes Capturadas por el dron	5
2.1.2. Despegue y Posicionamiento Inicial del Dron	6
2.1.3. Alineación y Orientación del Dron Frente al Aerogenerador	9
2.1.4. Inspección de las Aspas del Aerogenerador	12
2.1.5. Fortalezas y Debilidades	14
2.2. Trabajo de Campo	16
2.3. Enfoque elegido	17
3. Descripción del Sistema	20
3.1. Solución propuesta	20
3.1.1. Hipótesis de trabajo	20
3.1.2. Esquema de la solución	21
3.2. Planificador de Caminos	21
3.2.1. Generación de la malla de las aspas	22
3.2.2. Planificador de Rutas de Inspección del Aerogenerador	23
3.2.3. Evolución del Planificador de Caminos	25
3.3. Control del dron	28
3.3.1. Modo de Vuelo Offboard	28
3.3.2. Sistemas de Referencia	29
3.3.3. Gestión de Setpoints	29
3.3.4. Control de Orientación (Yaw)	31
3.3.5. Publicación y Suscripción de Tópicos	31
3.3.6. Posición Local del Vehículo y EKF2	33
3.3.7. Gestión de Estados Críticos	33
3.3.8. Adaptabilidad y Correcciones en Tiempo Real	33
3.3.9. Recepción de una Posición GPS	34
3.3.10. Ajuste de la Velocidad	34
3.3.11. Rotación Centrada en el Aerogenerador	36
3.3.12. Vuelta a Tierra	37
3.4. Reconocimiento del aerogenerador	38
3.4.1. Detección del aerogenerador en etapa de posicionamiento ortogonal	38

3.4.2.	Detección de las aspas durante la inspección	42
3.4.3.	Efecto del viento en la cámara	44
3.5.	Máquina de Estados del Sistema de Control del Dron	49
3.5.1.	Transición entre Estados	49
3.5.2.	Estado <code>IdleState</code>	51
3.5.3.	Estado <code>TakeoffState</code>	52
3.5.4.	Estado <code>ApproachState</code>	52
3.5.5.	Estado <code>OrthogonalAlignmentState</code>	54
3.5.6.	Estado <code>RegistrationState</code>	56
3.5.7.	Estado <code>ReturnHomeState</code>	58
3.5.8.	Máquina de estados con una única ejecución	58
3.6.	Adquisición de imágenes	59
4.	Experimentación	61
4.1.	Simulaciones	62
4.1.1.	Entornos de hardware	62
4.1.2.	Entorno de software	62
4.2.	Mediciones realizadas	63
4.3.	Pruebas realizadas	63
4.3.1.	Alineación y Orientación del Dron Frente al Aerogenerador	64
4.3.2.	Inspección de las Aspas del Aerogenerador	66
4.4.	Análisis de las experimentaciones	70
4.4.1.	Capacidad de posicionamiento ortogonal	70
4.4.2.	Desempeño durante la inspección	70
4.4.3.	Impacto de las condiciones ambientales	71
4.4.4.	Eficiencia temporal y energética	71
4.4.5.	Eficiencia computacional	71
4.5.	Ejemplos de Ejecuciones del Sistema	72
5.	Conclusiones y Trabajo Futuro	73
5.1.	Conclusiones	73
5.2.	Trabajo futuro	74
5.2.1.	Uso de una flota de drones para la inspección	74
5.2.2.	Módulo para evitar obstáculos	74
5.2.3.	Reanudación de inspecciones interrumpidas	74
5.2.4.	Planes de contingencia ante errores	74
5.2.5.	Mejora de la integración del Planificador de Caminos	75
5.2.6.	Mejoras en la adquisición de imágenes	76
5.2.7.	Pruebas Unitarias	76
5.2.8.	Implementación en el Mundo Real	76
5.2.9.	Seguridad Informática	77
Anexo A		80
5.3.	Glosario	80

Anexo B	82
5.4. Herramientas y Tecnologías	82
5.4.1. ROS 2	82
5.4.2. Gazebo	82
5.4.3. QGroundControl	83
5.4.4. MAVLink	83
5.4.5. PX4	83
5.4.6. EKF	83
5.4.7. Google OR-Tools	84
5.5. Conceptos	84
5.5.1. Problema del Viajante Asimétrico (asymmetric TSP)	84
5.5.2. K-Means Clustering	84
5.5.3. K-Means Constrained	85

Capítulo 1

Introducción

A partir del año 2010, Uruguay adoptó como política de Estado la transición energética hacia fuentes autóctonas y renovables. En un año de lluvias normales, el 97% de la demanda eléctrica del país es cubierta por energías renovables, mediante una combinación de energía eólica (32%), biomasa (17%), solar (3%) e hidroeléctrica (45%). Según el informe de 2023 de la Red de Política de Energía Renovable para el Siglo XXI (REN21), Uruguay se encuentra en segundo lugar a nivel mundial en cuanto a la participación de energías renovables variables, como la solar y la eólica, en su matriz de generación eléctrica (Uruguay XXI 2023).

Previo a 2008, en Uruguay no existían parques eólicos a gran escala. Actualmente, hay 41 parques eólicos en funcionamiento en el país, con una potencia instalada de 1.514 MW (Uruguay XXI 2023). Este crecimiento en la capacidad de generación conlleva la necesidad de dedicar más tiempo y recursos al mantenimiento, la detección de fallas y las reparaciones de los aerogeneradores.

Las aspas de los aerogeneradores son susceptibles a la degradación por fatiga debido a las cargas de viento. Diversos factores, como la erosión, lluvias intensas, rayos, acumulación de hielo, vientos fuertes y colisiones con aves, pueden provocar daños significativos. En particular, la presencia de arena o gotas de agua en el viento durante la operación de los aerogeneradores puede ocasionar una considerable erosión (Memari et al. 2024).

Castorrini et al. 2020 logró modelar la erosión en el borde de ataque de las aspas de los aerogeneradores, simulando hoyos y surcos mediante Dinámica de Fluidos Computacional (CFD). Su estudio, realizado en un aerogenerador de 5 MW, evidenció una pérdida anual de producción de energía del 1-2% debido a los daños en el borde de ataque.

Dicho estudio demuestra que la detección temprana de defectos en las superficies de las aspas es fundamental para mantener la eficiencia operativa. Identificar estos defectos de manera anticipada facilita la reparación oportuna de las aspas, prolongando su vida útil, reduciendo el tiempo de inactividad de los aerogeneradores y mitigando el riesgo de fallas estructurales más severas (Tanriverdi et al. 2023).

Los métodos tradicionales para la inspección de las aspas de los aerogeneradores, como el acceso mediante cuerdas, grúas o plataformas móviles, resultan ser soluciones lentas y costosas. Según Tanriverdi et al. 2023, estas técnicas requieren detener el aerogenerador por un período de 12 a 18 horas, lo que no solo representa una pérdida considerable en la producción de electricidad, sino que también implica elevados costos operativos y riesgos para la seguridad laboral. Por otro lado, métodos como la fotografía desde el suelo son seguros para el operario, más rápidos y económicos.

Sin embargo, presentan limitaciones en la precisión de los ángulos de captura, lo que dificulta determinar con exactitud la ubicación y el tamaño de los daños en las aspas (Tanriverdi et al. 2023).

En este contexto, el uso de drones equipados con cámaras y sensores especializados ha surgido como una alternativa más eficiente para la inspección de aerogeneradores, y es una técnica ya adoptada en gran cantidad de parques eólicos. Esta técnica se destaca por su simplicidad, la mejora en la seguridad de las operaciones y la significativa reducción de tiempo y costos. En el caso de drones equipados únicamente con cámara y LiDAR, la detección se limita a identificar fallas superficiales en las aspas. No obstante, la incorporación de otros sensores puede mejorar significativamente estos resultados. Por ejemplo, el estudio de Hwang et al. 2024 realizó un escaneo mecánico con un láser lineal combinado con la captura simultánea de ondas térmicas, logrando detectar de forma exitosa una delaminación interna —es decir, una separación de las capas internas del material— de 10 mm de diámetro, ubicada a 1 mm de profundidad bajo la superficie de un aspa, desde una distancia de 10 m.

Actualmente, las inspecciones con drones se realizan de manera manual, lo que requiere la intervención de operadores especializados. Esto puede dar lugar a inconsistencias en los resultados y depender en gran medida de la habilidad del operador, lo que introduce la posibilidad de errores humanos. En su estudio, Grindley et al. 2024 analizaron 77 informes de accidentes de drones entre 2012 y 2024, encontrando que el 55 % de ellos identificaron una contribución humana. Esta tasa es más alta que en estudios previos (32 % en Manning et al., 2004; 42 % en Giese et al., 2013; 20 % en Wild et al., 2017), excepto en el de Tvaryanas et al. (2006), que reportó un 60 %. La diferencia podría deberse a cómo se define “contribución humana”, ya que si solo se considera el error humano directo, se indica que la tasa baja a un 40 %, similar a estudios previos. Ante estas limitaciones, se ha planteado la evolución hacia sistemas de inspección con drones autónomos.

La automatización de los drones en las inspecciones de aerogeneradores ofrece avances notables en eficiencia. Al implementar sistemas autónomos, es posible realizar inspecciones estandarizadas y consistentes, sin importar el modelo del aerogenerador. Esta capacidad de estandarización es fundamental, ya que permite recopilar datos con mejor precisión y facilita el postprocesamiento de los resultados.

Además, la automatización reduce significativamente el tiempo de la operación en comparación con las inspecciones tradicionales¹. Esto implica que los aerogeneradores pasan menos tiempo fuera de servicio, lo que minimiza la pérdida de producción de energía y maximiza la eficiencia operativa. Reducir el tiempo de inspección también permite realizar más inspecciones en menos tiempo, mejorando el mantenimiento preventivo al detectar problemas de manera temprana y evitando fallos. Aunque el proyecto actual no contempla su uso, la incorporación de una flota de drones podría optimizar aún más los tiempos de inspección. Por ejemplo, al emplear tres drones, sería posible inspeccionar simultáneamente las tres aspas del aerogenerador.

Este proyecto tiene como objetivo la implementación de un sistema capaz de realizar la inspección de las aspas de un aerogenerador mediante el uso de un dron autónomo, el cual recolectará imágenes para su posterior procesamiento y detección de fallas. El procesamiento de las imágenes no está contemplado dentro del alcance del presente trabajo. Para el control y la operación del dron, se utilizarán PX4 y el framework ROS 2, mientras que la realización de pruebas del sistema se llevará a cabo en el entorno de simulación Gazebo.

El resto del documento se organiza de la siguiente manera:

¹<https://enterprise-insights.dji.com/user-stories/wind-turbine-inspection-ids>

En el **Capítulo 2** se destacan los aspectos más relevantes de la literatura consultada, el trabajo de campo realizado y las hipótesis de trabajo.

En el **Capítulo 3** se proporciona una visión general de la solución propuesta, las decisiones de diseño tomadas y el proceso de desarrollo.

En el **Capítulo 4** se presentan las pruebas y mediciones realizadas para evaluar el desempeño del sistema, así como el análisis de los resultados obtenidos.

Finalmente, en el **Capítulo 5** se concluye el trabajo destacando los principales aportes y mencionando las direcciones de trabajo futuro más importantes.

Capítulo 2

Marco de Trabajo

2.1. Revisión de Antecedentes

La revisión bibliográfica se basó en el método propuesto por Okoli 2015. En el contexto del presente proyecto, la búsqueda de fuentes se realizó principalmente en el portal Timbó, utilizando términos clave relacionados con el problema, estos fueron: *wind turbine*, *drone*, *inspection*, *path-planning*, *image processing*. A su vez, dado que el uso de *ROS* y *Gazebo* ya estaba definido para este proyecto, estos términos se incluyeron en las búsquedas para identificar estudios relevantes que utilizaran dichas tecnologías.

Los documentos seleccionados para incluir en la revisión se centraron en aquellos de publicación reciente y que aborden alguna de las problemáticas principales de este estudio, tales como: posicionamiento, planificación de trayectorias (path-planning), y procesamiento de imágenes para la detección de fallas.

Un criterio de exclusión utilizado fue el idioma, aceptándose solo documentos en español o inglés. También se estableció como criterio que la fecha de publicación estuviera dentro de los últimos cinco años; sin embargo, se mantuvo cierta flexibilidad en este aspecto, ya que se encontró una publicación del año 2015, ampliamente citada por otras investigaciones. Debido a su relevancia en el ámbito, se consideró valioso incluirla como referencia.

Los documentos utilizados se encuentran en la Tabla 2.1.

Tras revisarlos, se identificaron etapas comunes que se repetían en los diferentes enfoques tomados en cada uno de estos. Estas etapas son las siguientes:

1. Preprocesamiento de las Imágenes Capturadas por el Dron
2. Despegue y Posicionamiento Inicial del Dron
3. Alineación y Orientación del Dron Frente al Aerogenerador
4. Inspección de las Aspas del Aerogenerador

Aunque estas etapas no siempre se presentan ni se llevan a cabo de forma idéntica, comparten similitudes claras. En las secciones a continuación se describen estas etapas basadas en la información recopilada de las fuentes consultadas.

Título	N° de referencias	N° de citas	Año
Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection (Stokkeland et al. 2015)	22	68	2015
Wind Turbine Blade Inspection Based on Unmanned Aerial Vehicle (UAV) Visual Systems (Rao et al. 2019)	4	12	2019
Autonomous Wind Turbine Inspection using a Quadrotor (Gu et al. 2020)	12	12	2020
Autonomous Wind-Turbine Blade Inspection Using LiDAR-Equipped Unmanned Aerial Vehicle (Car et al. 2020)	34	19	2020
Visual Tracking Nonlinear Model Predictive Control Method for Autonomous Wind Turbine Inspection (Amer et al. 2023)	23	0	2023
LiDAR-based automated UAV inspection of wind turbine rotor blades (Castelar Wembers et al. 2024)	33	0	2024

Tabla 2.1: Documentos considerados en el análisis de antecedentes, junto con su fecha de publicación, número de citas y referencias.

2.1.1. Preprocesamiento de las Imágenes Capturadas por el dron

Los documentos analizados aplican, en las etapas analizadas posteriormente, técnicas avanzadas de análisis de imágenes, como la transformada de Hough y modelos de detección de objetos como YOLO, para extraer información destinada a la navegación del dron. Estas técnicas dependen de imágenes de alta calidad, lo que hace esencial el preprocesamiento de las imágenes capturadas por el dron. Este preprocesamiento mejora la calidad de las imágenes, maximizando la precisión en la detección de características clave como bordes, líneas y formas, optimizando así los métodos de análisis.

Rao et al. 2019 primero emplea un filtrado por wavelets para eliminar el ruido introducido por el entorno electromagnético y las limitaciones de la cámara del dron. Este método se prefiere sobre el filtrado de ruido tradicional debido a su simplicidad y buen rendimiento en tiempo real.

Luego, se estima la función de dispersión de puntos (PSF) de la imagen y se utiliza el filtrado Wiener para corregir el desenfoque por movimiento causado por el viento. Además, se aplica el modelo SSR de Retinex para preservar los detalles de la imagen y mejorar el balance de color, lo cual es especialmente útil en condiciones de visibilidad reducida, como en ambientes neblinosos o arenosos.

Posteriormente, el sistema procede a la binarización y extracción de bordes. Aquí, se separa el objeto del fondo utilizando el espacio de color HSV y se detectan los bordes en la imagen mediante el operador Sobel.

Finalmente, se utiliza la transformada de Hough para detectar líneas en la imagen. Estas líneas son cruciales ya que proporcionan parámetros de navegación para el dron y pueden indicar la presencia de grietas en las aspas del aerogenerador.

En Stokkeland et al. 2015 también se menciona el preprocesamiento de la imagen. Se utiliza la detección de bordes de Canny, que comienza con el cálculo de derivadas direccionales mediante el operador Sobel en imágenes en escala de grises. Luego, los bordes encontrados en la imagen se afinan conservando sólo los más fuertes y finalmente se convierte el mapa de bordes en una imagen binaria, eliminando los más débiles que probablemente no sean significativos.

Una vez binarizada de esta forma la imagen, al igual que en Rao et al. 2019 se utiliza la transformada de Hough para detectar líneas. En particular, se menciona como eficiente la transformada probabilística progresiva de Hough proporcionada por OpenCV (librería para visión por computadora y aprendizaje automático).

2.1.2. Despegue y Posicionamiento Inicial del Dron

En esta sección se detalla como los distintos artículos abordan la primera tarea del problema a resolver. Esta consiste en lograr que el dron despegue y aproximarlo al aerogenerador hasta ubicarse a una distancia específica del mismo. Este posicionamiento es crucial para asegurar que el aerogenerador se mantenga dentro del campo de visión del dron durante la siguiente fase. Diversos estudios establecen que el dron debe alcanzar una altura predeterminada, generalmente coincidente con la altura del rotor. A su vez, se detallan las precondiciones consideradas por los diferentes estudios recopilados.

Para comprender mejor el entorno y los componentes claves de los aerogeneradores involucrados en esta y en las siguientes etapas, la Figura 2.1 presenta una vista detallada de los elementos principales de un aerogenerador: la base, la torre, el rotor, la góndola y las aspas. Además, se ilustra un aspa en posición vertical y una sección transversal de la misma, destacando características esenciales como la raíz, la punta y los bordes de ataque y salida. Se describe también la posición Mercedes o Y invertida, que, según Castelar Wembers et al. 2024, debe ser la posición mantenida por el aerogenerador durante toda la inspección.

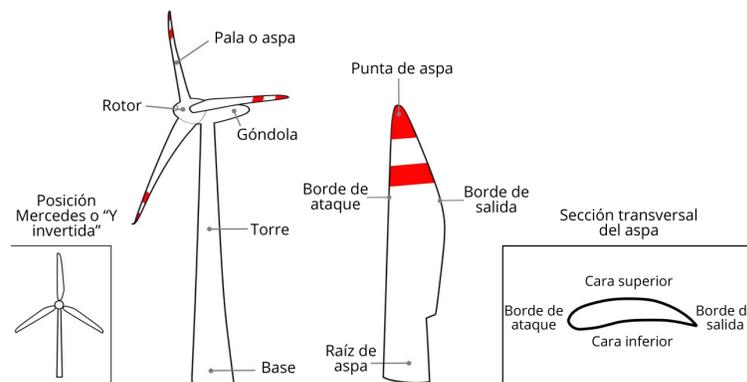


Figura 2.1: Principales componentes de un aerogenerador: base, torre, rotor, góndola y aspas. Además, se detalla un aspa del rotor posicionada verticalmente en vista frontal y en sección transversal, mostrando la raíz del aspa, la punta del aspa, los bordes de ataque y salida, y las superficies superior e inferior. También se representa la posición Mercedes o Y invertida. (Castelar Wembers et al. 2024)

Con el mismo propósito, la Figura 2.2 ilustra el ángulo de guiñada (yaw), el ángulo de cabeceo

(pitch) y el ángulo de alabeo (roll) de un dron. Estos tres ángulos de orientación definen la actitud de una aeronave. El ángulo de guiñada se refiere a la rotación del dron alrededor de su eje vertical, determinando la dirección en la que apunta la parte frontal del dron. El ángulo de cabeceo describe la inclinación hacia adelante o hacia atrás, controlando el movimiento sobre el eje lateral, mientras que el alabeo se refiere a la inclinación lateral sobre el eje longitudinal.

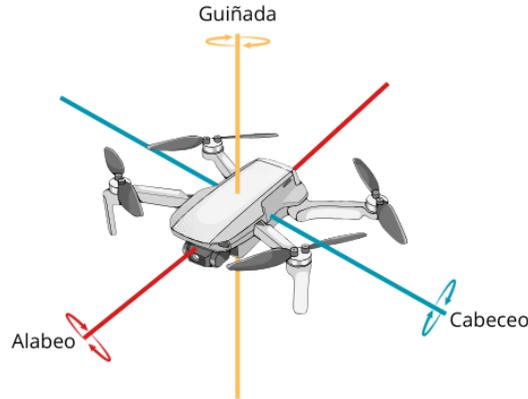


Figura 2.2: Se ilustran los tres ángulos de orientación que definen la actitud de un dron: ángulo de guiñada (yaw), ángulo de cabeceo (pitch) y ángulo de alabeo (roll). (Imagen generada por los autores).

Una vez definidos los componentes clave del aerogenerador y los ángulos de orientación del dron, se puede comprender mejor el contexto en el que se desarrolla la inspección.

Lo primero a tener en cuenta en la preparación para la inspección es el posicionamiento inicial del dron en el suelo. Stokkeland et al. 2015 detalla que el dron debe situarse en la base del aerogenerador, a una distancia mayor al largo de las aspas, para minimizar el riesgo de colisiones. A su vez, este menciona que la cámara debe orientarse hacia la torre para garantizar un despegue seguro desde esa ubicación. De manera similar, Rao et al. 2019 destaca la importancia de que la cámara esté enfocada hacia la torre en esta etapa, asegurando una correcta orientación para las fases posteriores de la inspección.

Una vez iniciado el sistema, Gu et al. 2020 plantea que lo primero que se debe hacer es ascender verticalmente hasta la altura del rotor y luego dirigirse horizontalmente hacia el aerogenerador hasta detenerse a la mínima distancia segura (d_{seg}) del rotor. Esta distancia (d_{seg}) se determina según la Ec. (2.1), en función de la longitud del aspa (l) y el ángulo de visión (AOV) de la cámara, como se ilustra en la Figura 2.3.

$$\tan\left(\frac{1}{2}AOV\right) = \frac{l}{d_{seg}} \quad (2.1)$$

Por su parte, Castelar Wembers et al. 2024 propone tres formas distintas de llevar a cabo esta etapa.

El primer método propuesto, requiere que el dron vuele hacia adelante hasta que el LiDAR detecte al aerogenerador, y luego se posicione centralmente con respecto a la torre, ascendiendo a

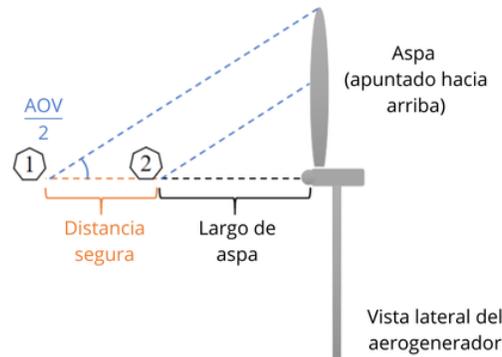


Figura 2.3: Ilustración de la Ec. (2.1). La distancia de seguridad está diseñada para garantizar la vista completa de la aspa en el campo de visión. Por ejemplo, la cámara puede ver el aspa completa cuando el dron está en la posición 1. Mientras que, en la posición 2, solo se muestra una parte del aspa en la imagen, lo que podría afectar el rendimiento del sistema de detección de aspas basado en visión más adelante (Gu et al. 2020).

una velocidad preestablecida y evitando obstáculos (por ejemplo, un aspa del rotor) mediante giros de 90° .

Sin embargo, esta primera solución es descartada por Castelar Wembers et al. 2024 ya que su LiDAR tiene un alcance de 20 metros y la distancia al aerogenerador probablemente comience siendo mayor a esta. Además, si bien este enfoque requiere que el dron apunte hacia el aerogenerador, no se puede asumir que el dron seguirá estando orientado en dicha dirección después del despegue, por lo que simplemente avanzar hasta que la distancia sea menor a 20 metros no es una buena idea.

El segundo método propuesto es especificar las coordenadas GPS del aerogenerador, y de esa forma el dron sabrá dónde se encuentra y podrá ascender verticalmente hasta el objetivo.

Si bien este enfoque funcionaría, sería mejor poder detectar el aerogenerador de forma más autónoma. Por este motivo, surge la tercera solución.

La tercera solución consiste en detectar la torre del aerogenerador automáticamente. El dron utiliza su cámara FPV (First Person View) para detectar los bordes de la torre del aerogenerador, girar y moverse hacia él hasta estar dentro del alcance del LiDAR. Para lograr esto, se utiliza un algoritmo (capaz de ejecutarse en tiempo real en la computadora a bordo) que incluye un filtro de desenfoque gaussiano, reducción de ruido y la transformada de Hough para detectar líneas verticales. Una vez detectada la torre, el dron deberá ascender hasta quedar a la altura del borde superior de la góndola.

Según Rao et al. 2019, como la torre es vertical, para identificarla se deben reconocer dos líneas verticales paralelas entre sí que corresponderían a los bordes de esta. Se debe tener en cuenta que el largo y la distancia entre estas deben ser adecuados respecto al aerogenerador.

Una vez que se han identificado las líneas correspondientes a la torre, se utiliza la línea central entre las dos líneas paralelas para corregir tanto el ángulo de guiñada como la translación lateral del dron, logrando así que la torre quede centrada en la imagen. Posteriormente, el dron asciende de manera vertical hasta alcanzar la altura preestablecida del centro del rotor.

2.1.3. Alineación y Orientación del Dron Frente al Aerogenerador

El punto de partida para la inspección, según los estudios analizados, es la posición en la que el rotor del aerogenerador se encuentra centrado en la imagen, con el dron orientado de manera ortogonal y a una distancia segura definida. Esta configuración es esencial para asegurar una correcta alineación y visualización de las aspas del aerogenerador desde el inicio de la inspección. En particular, el rotor se selecciona como punto de referencia debido a que es uno de los componentes más fácilmente identificables del aerogenerador.

Para lograr esta alineación, Rao et al. 2019 primero detecta las aspas del aerogenerador guiándose por el hecho de que estas forman un ángulo de 120° con sus aspas adyacentes, por lo que se busca en la imagen binarizada tres líneas que tengan otra línea adyacente a sí misma a dicho ángulo.

Una vez encontradas, se utiliza la detección de círculos de Hough para encontrar el centro del rotor. Este debe estar en el cruce de las líneas de las aspas, por lo que el círculo cuyo centro está más cercano al cruce de las aspas será el correspondiente al centro del rotor y el dron procede a posicionarse frente a este.

Gu et al. 2020 incorpora una etapa que consiste en encontrar el lado frontal del aerogenerador. Para esto, el dron realiza un movimiento circular horizontal con respecto al rotor y compara la posición relativa del rotor y la góndola detectados por el sistema de detección basado en visión.

Para lograr esta detección en tiempo real se utiliza el modelo YOLOv3¹. Si el rotor detectado está a la izquierda de la góndola detectada, el camino más corto para que el dron llegue al lado frontal del aerogenerador es en dirección de las agujas del reloj, por lo que se trasladará en esa dirección. Análogamente, si el rotor está a la derecha de la góndola, el dron se trasladará en dirección antihoraria. En el momento en que se detecta el rotor y no la góndola, el dron detendrá el movimiento circular y habrá encontrado el lado frontal.

Luego para estimar la posición del rotor Gu et al. 2020 lleva a cabo un enfoque similar a Rao et al. 2019, en el que un sistema de detección basado en visión recupera el ángulo de rotación de las aspas y estima las coordenadas de la posición del rotor.

Para la detección de las aspas, se utiliza una variante del algoritmo de la transformada de Hough, específicamente diseñado para la inspección de aerogeneradores. A partir de los segmentos de línea de las aspas detectadas, se obtienen los ángulos de rotación y la coordenada del rotor en la imagen se calcula como el punto de intersección de estas.

Teniendo en cuenta estos cálculos, el dron se acerca al aerogenerador mientras mantiene el rotor en el centro de la imagen.

A diferencia de Rao et al. 2019, en Gu et al. 2020 se menciona que una vez que el dron comienza a acercarse al aerogenerador, la performance de la detección de las aspas por el algoritmo de Hough comenzará a degradarse, por lo que se utiliza también el YOLOv3 para posicionarse frente al rotor.

Cuando la coordenada del rotor es devuelta por Hough, la red YOLOv3 la compara con el centro del rotor detectado por ella. Si la distancia entre ellas supera determinado umbral, la red YOLOv3 se encarga del posicionamiento del rotor y esta coordenada se registra como información a priori para la siguiente detección durante el acercamiento del dron al rotor. Finalmente, se detendrá a una distancia segura del rotor medida por el LiDAR 2D.

Por su parte, Stokkeland et al. 2015 plantea que esta fase contiene tres tareas ($T1$, $T2$ y $T3$) con prioridad decreciente. Cada tarea tiene condiciones específicas que determinan su finalización. Solo es posible avanzar a la siguiente tarea una vez que todas las condiciones de la tarea actual hayan sido satisfechas. Si durante la ejecución de una tarea se invalidan las condiciones de una tarea

¹<https://docs.ultralytics.com/models/yolov3/>

de mayor prioridad, el sistema interrumpe la tarea en curso y pasa a ejecutar la tarea de mayor prioridad que no haya cumplido sus condiciones de finalización.

$T1$ asegura que el dron esté orientado hacia el rotor, tanto vertical como horizontalmente. Si este está dentro de cierto umbral, se pasa a $T2$. De lo contrario, se ajusta la orientación del mismo.

En la segunda tarea, se ajusta el ángulo de guiñada relativo volando lateralmente hasta alcanzar una tolerancia específica. En caso de que el dron deje de estar orientado hacia el rotor, se vuelve a $T1$. De lo contrario, se sigue con $T3$.

Por último, en $T3$ el dron avanza hacia el objetivo hasta alcanzar la distancia deseada, siempre con la posibilidad de volver a $T1$ o $T2$ para corregir el rumbo. En la Figura 2.4 puede verse un ejemplo de cómo se vería la ejecución de este algoritmo visto desde arriba.

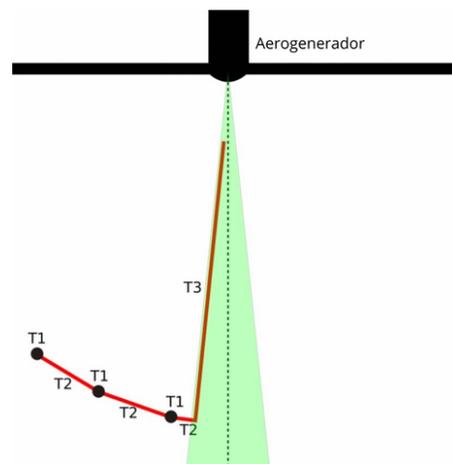


Figura 2.4: Perspectiva desde arriba que muestra un escenario de ejemplo del camino de navegación (línea roja) del artículo Stokkeland et al. 2015. Los puntos negros marcados como $T1$ representan las ubicaciones donde se lleva a cabo la tarea $T1$, mientras que las líneas rojas etiquetadas como $T2$ y $T3$ indican las trayectorias recorridas durante la ejecución de dichas tareas. El triángulo verde ilustra el rango tolerado para el ángulo de guiñada relativo.

En la solución planteada en este artículo, con la llegada de cada fotograma se ejecuta el algoritmo de navegación en bucle, ver Figura 2.5.

Una vez preprocesada la imagen, pasa a detectar la torre, el rotor y las aspas.

Para detectar la torre, al igual que Rao et al. 2019, opta por buscar líneas verticales en la imagen. A su vez, agrega que al volar a una altura similar a la del rotor, la base de la torre queda fuera del marco de la imagen. Por lo tanto, para identificar a la torre se busca que dichas líneas verticales terminen cerca del borde inferior de la imagen.

Si al menos una línea es identificada como parte de la torre, el algoritmo pasa a identificar las aspas. Para esto, primero se busca el punto más alto entre las líneas de la torre y así se obtiene una primera posición estimada para el rotor, y por lo tanto todas las líneas que terminen en un área circular alrededor de esta son candidatas de pertenecer a las aspas.

Cabe destacar que si la línea formada por el horizonte se encuentra en la imagen en una posición vertical similar a la del rotor, se puede confundir a esta con el borde de un aspa. Una posible solución sería ignorar las líneas horizontales detectadas; sin embargo, esta opción se descarta en

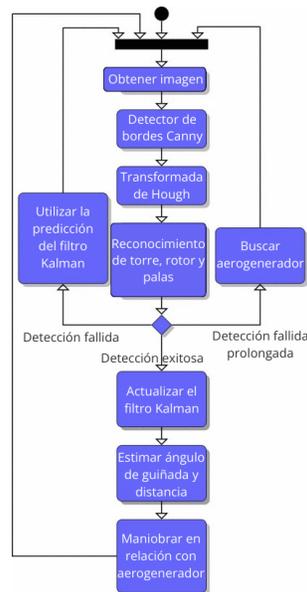


Figura 2.5: Diagrama que describe el flujo del sistema utilizado para la navegación automática del artículo Stokkeland et al. 2015.

este artículo, ya que una de las aspas podría estar en posición horizontal en la imagen al no asumirse la precondition de que las aspas estén en Y invertida.

Las líneas que se determinan como pertenecientes a las aspas se ordenan por ángulo y se agrupan con otras líneas de ángulos adyacentes, formando un “objeto de aspa”. Luego se utiliza un sistema de votación para detectar aspas falsas, donde las aspas que no tienen un ángulo cercano a 120 grados (teniendo en cuenta una tolerancia de 20 grados en este caso) reciben votos. Esto se realiza comparando cada aspa detectada entre sí, y si una aspa recibe tres votos o más, se considera una detección falsa y se elimina.

Una vez localizadas las aspas, se pasa a identificar el centro del rotor. Para esto se calculan las intersecciones entre cada par de líneas correspondientes a las aspas, excepto las de la misma aspa. Luego, de todos los puntos obtenidos se eliminan aquellos que se encuentren lejos horizontalmente de la torre y se calcula el promedio de los restantes. Este promedio proporciona la posición estimada del centro del rotor.

Este algoritmo realiza un seguimiento del centro del rotor en la imagen para permitir que el dron maniobre según lo descrito en las tareas. Por lo tanto debe continuar rastreando cómo se mueven las distintas partes del aerogenerador detectadas en la secuencia de imágenes que va tomando, y actualizar la información sobre la posición de estas entre fotogramas. Para esto se utiliza un filtro de Kalman para manejar el ruido entre fotogramas y la falta de detección del rotor. Si el rotor no se detecta, se estima su posición utilizando una predicción del filtro, y si después de cierto tiempo este sigue sin ser localizado, cambia al estado de búsqueda, en el que deja de moverse y rota sobre su eje de guiñada hasta que el aerogenerador vuelva a ser identificado.

Por su parte, Castelar Wembers et al. 2024 menciona dos opciones posibles para posicionar el dron con respecto al aerogenerador y las aspas.

La primera consiste en detectar la góndola y determinar su orientación, la cual puede usarse para inferir la posición y orientación de las aspas. La segunda se basa en una descomposición en valores singulares (SVD) con respecto al vertical del aerogenerador.

Ambos métodos resultan en que el dron se posicione a la distancia establecida frente al borde de salida o la superficie superior del aspa. Sin embargo, al realizar las pruebas de campo, el primer método trajo múltiples problemas y resultó en un peor rendimiento que el conseguido por el segundo. Por lo tanto, en esta fase el dron asciende 10 metros por encima de la góndola y realiza una maniobra de rotación de 360° alrededor del aspa, para así poder realizar la determinación SVD de la orientación de esta.

2.1.4. Inspección de las Aspas del Aerogenerador

Una vez posicionados frente al centro del rotor a la distancia definida, se comienza con la inspección de las aspas.

En Rao et al. 2019, el procedimiento consiste en elegir una de las aspas detectadas y utilizar su ángulo y longitud para inspeccionarla. El dron navega por esta hasta llegar a la distancia especificada y vuelve al centro del rotor, marcando dicha aspa como inspeccionada. Este procedimiento se repite hasta completar la inspección de las tres aspas.

En Gu et al. 2020, el dron navega a lo largo del aspa dos veces, inspeccionando tanto el borde de ataque como el borde de fuga. Durante esta navegación, la cámara del dron se orienta hacia el aspa con un ángulo de 45° , lo que permite inspeccionar simultáneamente tanto el borde como la cara correspondiente del aspa. Además, utilizando un LiDAR 2D montado en el gimbal, el dron mantiene una distancia constante respecto al aspa durante toda la inspección. Después de inspeccionar un lado del aspa, el dron vuela alrededor de la punta para inspeccionar el otro lado, asegurando así una cobertura completa. Cuando un aspa ha sido completamente inspeccionada, el dron transita a la siguiente siguiendo una secuencia predefinida determinada utilizando una variante de la transformada de Hough que minimiza el riesgo de colisiones, especialmente con la torre.

Por su parte, Car et al. 2020 se centra en automatizar el recorrido de un aspa desde la base hasta la punta y de regreso a la base (una sola cara del aspa) manteniendo distancia relativa constante y dirección perpendicular al aspa, como se muestra en la Figura 2.6.

Para esto se utiliza un sensor LiDAR para obtener una nube de puntos, a partir de los cuales se obtiene el plano del aspa utilizando el algoritmo RANSAC.

En Castelar Wembers et al. 2024, se realiza la secuencia de movimientos descrita en la Figura 2.7. Una vez determinada la orientación del aspa mediante el SVD, el dron vuela hasta encontrarse en posición frontal al borde de ataque o de salida a una distancia preestablecida a la estructura (entre 7 y 12 metros). Desde esta posición, el dron asciende a lo largo de este borde registrando imágenes hasta que el aspa sea demasiado estrecha para ser detectada consistentemente por el escaneo LiDAR. Desde esa altura hasta el final del aspa, el dron realiza una maniobra de rotación de 360° alrededor del aspa (mientras captura imágenes) cada 4 metros. Cuando el dron llega a la punta del aspa, desciende a lo largo de la cara inferior del aspa hasta la altura de la góndola mientras registra imágenes. Luego, rota hacia el lado de la cara superior y asciende nuevamente a la altitud preestablecida por encima de la góndola. Finalmente, rota hacia el último borde y nuevamente desciende a la altura de la góndola.

Por último, Amer et al. 2023 utiliza el alto de la torre, y el largo y ancho de las aspas para generar una malla simplificada del aerogenerador, a partir de la cual se calculará el path-planning. El proceso para calcular la ruta se divide en tres pasos: agrupación, ordenamiento e interpolación.

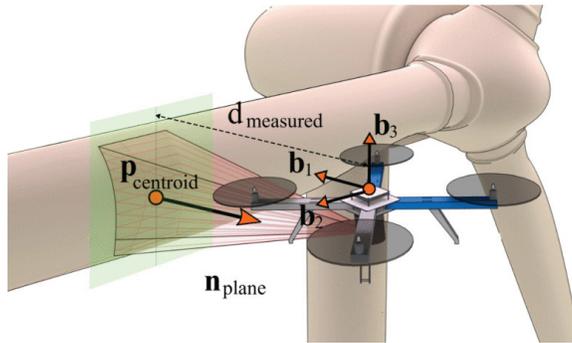


Figura 2.6: Figura del artículo Car et al. 2020 que muestra un dron equipado con LiDAR realizando una inspección autónoma un aspa de un aerogenerador. A través del análisis de la nube de puntos obtenida, el dron es capaz de mantener una distancia relativa constante $d_{measured}$, mientras mantiene su rumbo paralelo al vector normal del plano n_{plane} . Por último, el desplazamiento vertical del vector $p_{centroid}$ se utiliza para mantener la altitud del dron en el centro del aspa en todo momento.

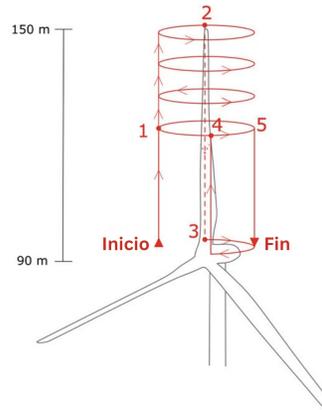


Figura 2.7: Figura correspondiente al artículo Castelar Wembers et al. 2024 mostrando la trayectoria de vuelo para la inspección.

Primero, los nodos que forman los triángulos del modelo del aerogenerador se agrupan en 12 superficies, que corresponden a las dos caras (superior e inferior) y a los bordes de ataque y salida de cada una de las tres aspas. La agrupación se realiza según la ubicación de los centroides y el valor de su vector normal.

Posteriormente, se construye un grafo utilizando los nodos situados en los extremos de cada superficie, como se ilustra en la Figura 2.8, y se resuelve como si fuera una instancia del problema del vendedor viajero (TSP) para definir el orden óptimo de recorrido. Finalmente, se determina una lista de puntos intermedios mediante interpolación.

A su vez se mantiene la posición óptima del dron mediante un controlador VT-NMPC (Visual Tracking Nonlinear Model Predictive Control, en español: Modelo de Control Predictivo No Lineal para Seguimiento Visual) que ajusta constantemente las tasas de actitud y empuje a partir de

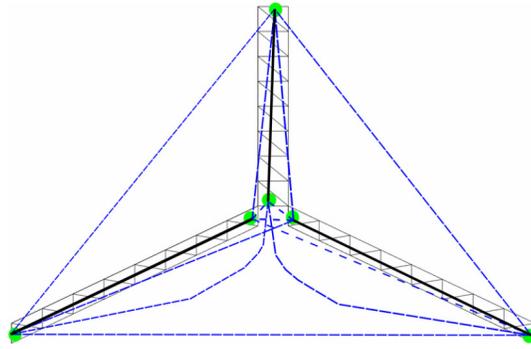


Figura 2.8: Representación del grafo obtenido en el artículo Amer et al. 2023. Un nodo (verde) se coloca al final de la superficie de cada aspa, los cuales están conectados por aristas (azules) para formar un grafo. El objetivo es encontrar la secuencia de nodos a visitar que minimice la distancia total recorrida.

la información visual del dron para seguir la superficie de inspección, asegurando que el dron se mantenga a una distancia adecuada de esta.

Esto se ilustra en la Figura 2.9, donde se representa gráficamente la posición óptima del dron durante la inspección. La imagen muestra la intersección entre la línea $r = r^{ref}$, el círculo de referencia $d = d^{ref}$, y el vector normal n , indicando el punto donde el dron debe situarse para mantener una distancia y ángulo adecuados respecto a la superficie inspeccionada.

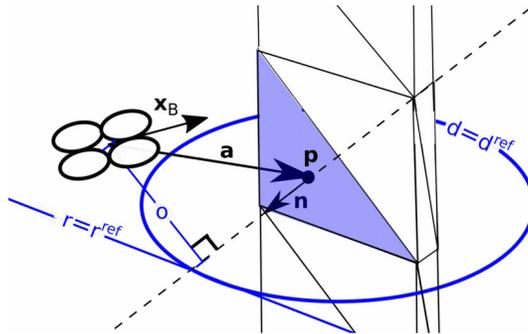


Figura 2.9: Representación gráfica de los costos visuales para los cálculos utilizados en el artículo Amer et al. 2023 al inspeccionar una superficie con normal n . La intersección entre la línea $r = r^{ref}$, el círculo $d = d^{ref}$ y el vector normal n representa la posición óptima que el dron debe tener durante la inspección con respecto a la superficie.

Una vez terminada la inspección, el dron regresa al punto de salida.

2.1.5. Fortalezas y Debilidades

A continuación se presentan las fortalezas y debilidades de los estudios revisados en este trabajo.

El estudio realizado por Stokkeland et al. 2015 destaca por haber llevado a cabo experimentos de campo en el parque eólico de Bessakerfjellet, proporcionando datos experimentales válidos.

Como debilidad se resalta que la detección es susceptible a cambios en las condiciones de iluminación, lo que puede impactar negativamente en la precisión de la detección de líneas.

El enfoque sugerido por Rao et al. 2019 tiene algoritmos de mejora de imágenes como Retinex que permiten una mejor visualización en entornos con poca visibilidad, pero a su vez esto es una gran limitante ya que el preprocesamiento de la imagen debe hacerse en tiempo real. Por este motivo, los algoritmos utilizados para esta tarea deben ser altamente eficientes y el dron debe ser capaz de procesar las imágenes rápidamente, utilizando de manera óptima sus recursos limitados.

Una fortaleza clave del sistema de Gu et al. 2020 es la integración de la red YOLOv3 con la transformada de Hough, lo que mejora su robustez. Aunque no se realizaron pruebas de campo, las simulaciones y pruebas HIL (Hardware-in-the-loop) sugieren un buen desempeño en escenarios reales. No obstante, el sistema presenta limitaciones, como la baja tasa de procesamiento de la red YOLOv3 (2 FPS), lo que podría dificultar su uso en aplicaciones en tiempo real.

Además, requiere configuraciones específicas: el aerogenerador debe estar en reposo con un ángulo fijo de guiñada y de rotación de las aspas, y un ángulo de ataque del aspa de 90 grados (es decir, las aspas son perpendiculares al plano de rotación). Además, se debe conocer tanto la longitud del aspa y la altura de la torre, como la información aproximada de GPS sobre la posición del rotor del aerogenerador.

En el estudio presentado en Car et al. 2020, se lleva a cabo una inspección autónoma del aspa de un aerogenerador de raíz a punta y de regreso, pero con la limitación de que un operador debe posicionar el dron frente al aspa antes de comenzar el proceso, lo que convierte la inspección en semi-autónoma.

Una ventaja clave del sistema es el uso de sensores LiDAR en combinación con drones, lo que permite obtener mediciones precisas a largas distancias. Otra fortaleza importante del sistema es que ha sido validado tanto en simulaciones como en condiciones experimentales reales. Se presentan resultados experimentales para un escenario de inspección de un aspa al aire libre, tanto en una maqueta como en un aspa de un aerogenerador de tamaño real.

El estudio de Amer et al. 2023 presenta una metodología robusta ante cambios en las condiciones del viento, demostrada en simulaciones y pruebas reales. Sin embargo, utiliza un modelo simplificado de los aerogeneradores (malla triangular básica) que no considera el perfil completo de las aspas, especialmente en sus extremos. Además, aunque el VT-NMPC ofrece mejor rendimiento en la inspección visual, requiere más poder computacional que métodos tradicionales, lo que puede ser un desafío para drones con recursos limitados.

Los algoritmos propuestos por Castelar Wembers et al. 2024 para el control y la planificación de rutas fueron probados en el entorno de simulación AirSim, así como en campos de aviación locales y en aerogeneradores reales, tanto en tierra como en el mar. Gracias a estos algoritmos, el tiempo total de la rutina de inspección se reduce a aproximadamente 14 minutos, lo que representa la mitad del tiempo que un piloto experto necesitaría para realizarla. Sin embargo, este estudio tiene la limitante de que sólo se inspecciona el aspa vertical del aerogenerador.

En conjunto, estas observaciones evidencian si bien se lograron avances significativos en la inspección con drones autónomos de aerogeneradores, aún existen desafíos clave que deben resolverse para alcanzar sistemas de inspección robustos.

2.2. Trabajo de Campo

El viernes 10 de mayo de 2024, se realizó una visita al parque eólico *Los Caracoles*, el primer parque de energía eólica de propiedad exclusiva de UTE. Ubicado en el departamento de Maldonado e inaugurado en 2009, este parque representa un importante avance en energías renovables en la región.

La visita formó parte de la fase inicial del proyecto, con el objetivo de entender en profundidad el funcionamiento del negocio, aprender de los expertos en el sector y observar de cerca los procesos operativos. La fecha fue elegida para coincidir con una inspección programada de uno de los aerogeneradores, lo que permitió una experiencia directa y práctica sobre su mantenimiento y funcionamiento.

Durante la visita al parque eólico, se tuvo la oportunidad de observar de primera mano cómo se realiza la inspección de los aerogeneradores utilizando drones. Este trabajo de campo permitió al equipo comprender en detalle el proceso de inspección, las acciones que se llevan a cabo y las posibilidades que tiene el operador para modificar la orientación de la góndola y las aspas.

Se observó que las aspas de un aerogenerador no solo giran alrededor del rotor, giro de fase, sino que también pueden rotar sobre sí mismas, lo que se conoce como ángulo de paso. Este mecanismo permite ajustar el ángulo de las aspas en función de la dirección del viento, optimizando así la eficiencia del aerogenerador bajo diferentes condiciones. El ángulo de paso está limitado a un rango máximo de 90 grados, lo que permite que las aspas cambien su orientación en respuesta a las variaciones del viento, pero siempre dentro de este límite.

El ángulo de guiñada del aerogenerador puede ajustarse libremente. Esta capacidad se utiliza para orientar el aerogenerador en función del viento, facilitando que se alinee hacia la dirección más favorable. A diferencia del ángulo de paso, la rotación del ángulo de guiñada no tiene restricciones.

Por lo tanto, aunque es posible mover el aerogenerador, los movimientos no pueden realizarse de cualquier forma, ya que deben ajustarse a los límites definidos por los ángulos de paso y guiñada.

En la Figura 2.10 se muestra un diagrama que representa tanto el giro de orientación o ángulo de guiñada, como el giro de pala o ángulo de paso, así como también el giro de fase de un aerogenerador.

También se detectó que es posible dejar fijo el ángulo de paso y de guiñada en un valor específico, así como controlar el ángulo de las aspas alrededor del rotor. Esto permite replicar de manera consistente el escenario de inspección.

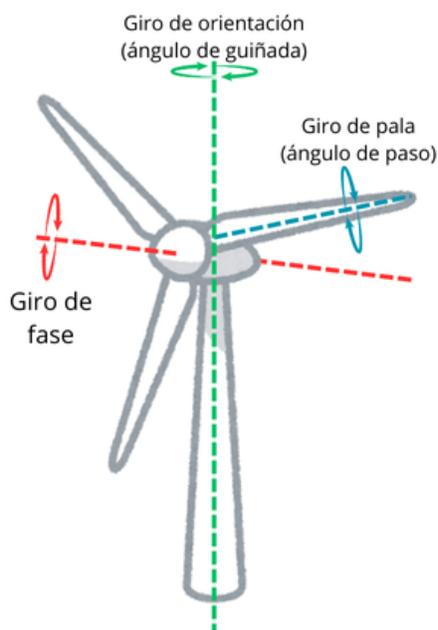


Figura 2.10: En verde se muestra el giro de orientación, correspondiente al ángulo de guiñada del aerogenerador; en azul, el giro de pala, que representa el ángulo de paso; y en rojo, el giro de fase, que indica la rotación de las aspas alrededor del rotor. (Imagen generada por los autores).

En la Figura 2.11, se observa al equipo en camino hacia el parque eólico. Una vez en el lugar, se tomó una fotografía del equipo en el parque, como se muestra en la Figura 2.12. Además, el dron inspeccionando una de las aspas de un aerogenerador está representado en la Figura 2.13. Finalmente, la Figura 2.14 muestra una vista capturada desde la cámara del dron durante la inspección.

2.3. Enfoque elegido

En esta sección se identifican las áreas clave de trabajo y las distintas formas de abordar el problema.

En función de la literatura relevada y las enseñanzas de la salida de campo se define la estrategia de la solución. El primer acercamiento al aerogenerador se realiza utilizando las coordenadas GPS del mismo. Para la identificación de las partes del aerogenerador, se opta por utilizar la transformada de Hough, descartando un enfoque basado en aprendizaje automático debido a su mayor complejidad y la necesidad de contar con un conjunto de imágenes etiquetadas. Aunque se encontraron conjuntos de imágenes de aerogeneradores, no se disponía de datos previamente etiquetados, lo que habría requerido un esfuerzo manual de etiquetado para su uso. Además, el proceso habría implicado un considerable tiempo y esfuerzo adicional para el entrenamiento de un modelo.

La alineación y posicionamiento ortogonal del dron respecto al aerogenerador se lleva a cabo mediante un bucle, inspirado en la metodología propuesta en Stokkeland et al. 2015, en el que primero el dron se alinea vertical y horizontalmente en el espacio. Luego, se traslada en sentido horario o antihorario hasta posicionarse ortogonalmente al aerogenerador. Una vez alcanzada esta



Figura 2.11: Equipo en ruta hacia el parque eólico.



Figura 2.12: El equipo en el parque *Los Caracoles*.



Figura 2.13: Dron inspeccionando una de las aspas del aerogenerador.



Figura 2.14: Vista de la cámara del dron durante la inspección.

orientación, el dron avanza hacia el aerogenerador y se repiten los pasos de alineación ortogonal, continuando el proceso hasta alcanzar la distancia establecida.

Por otro lado, para la planificación de la ruta se utiliza un enfoque similar al mencionado en Amer et al. 2023. Se utilizan las medidas del aerogenerador como parámetro, a partir del cual se genera un grafo cuyos nodos representan los extremos de las superficies frontales al dron de las

aspas, y se resuelve como un problema del vendedor viajero asimétrico (TSP).

La inspección seguirá el enfoque observado en la visita descrito en la Sección 2.2. Primero, el aerogenerador se orienta de frente al sol mediante un giro de orientación. A continuación, se inspecciona desplazándose a la altura de las aspas y recorriéndolas desde la raíz hasta la punta. En esta posición frontal, se inspecciona el borde de ataque o la cara superior, dependiendo de la orientación específica de las aspas en ese momento. Una vez completada esta inspección, se realiza un giro de pala (ángulo de paso) para rotar las aspas sobre su eje y así inspeccionar la superficie que falta.

Posteriormente, el aerogenerador se rota 180° mediante otro giro de orientación, de modo que la parte trasera quede orientada hacia el sol. En esta nueva posición, se inspecciona primero el borde de salida o la cara inferior, siguiendo un procedimiento similar. Finalmente, tras un nuevo giro de pala, se inspecciona la superficie que falta de esta manera, completando cuatro pasadas por aspa.

Capítulo 3

Descripción del Sistema

3.1. Solución propuesta

En esta sección se presenta un esquema general que describe el enfoque adoptado para abordar el problema.

3.1.1. Hipótesis de trabajo

Las hipótesis de trabajo que sustentan la solución propuesta:

- La inspección se realiza a una distancia aproximada de 10 metros de las aspas del aerogenerador. Esta distancia permite obtener imágenes de alta resolución que facilitan la detección de posibles daños o anomalías, manteniendo al mismo tiempo una operación segura del dron.
- Para obtener mejores resultados en las imágenes, es recomendable que el aerogenerador esté orientado lo más de frente posible al sol. Una buena iluminación mejora significativamente la calidad de las imágenes, reduciendo sombras y contrastes que podrían ocultar defectos en las aspas.
- La velocidad máxima del viento para realizar las inspecciones depende de las especificaciones técnicas y operativas del dron en condiciones de viento. Aunque el dron observado durante la salida de campo, de la serie DJI M200, tiene un límite técnico máximo de 12 m/s ¹, los operadores prefieren adoptar un enfoque conservador y establecer el límite en aproximadamente 9 m/s para asegurar una operación estable y segura.
- Se deben realizar cuatro pasadas por cada aspa: borde de ataque, borde de fuga, superficie superior e inferior. Dado que el aerogenerador debe estar orientado hacia el sol, la inspección sigue el mismo recorrido para cada una de estas superficies. En el caso de las inspecciones del borde de ataque y la superficie superior, así como del borde de fuga y la superficie inferior, es posible mantener fija la posición de la góndola y solo rotar las aspas, lo que implica que la rotación de la góndola solo se realiza una vez.

¹<https://www.dji.com/global/products/compare-m200-series>

- El aerogenerador se encuentra en posición de Y invertida, también llamada posición Mercedes en la bibliografía. Esta configuración presenta la ventaja de que ninguna de las aspas se superpone con la torre, lo cual facilita el uso de la transformada de Hough y simplifica el procesamiento posterior de imágenes para la detección de fallas.
- Es posible para el operador colocar el dron frente al aerogenerador.
- No existen obstáculos en el trayecto entre el punto de despegue del dron y el aerogenerador.

3.1.2. Esquema de la solución

La implementación del sistema se realizó utilizando ROS 2 y Python. El código fuente desarrollado está disponible en un repositorio público en GitLab² bajo una licencia MIT, lo que permite su uso libre.

Se utiliza una máquina de estados para dividir la misión en las principales etapas, las cuales se ilustran en la Figura 3.1 y se enumeran a continuación:

1. **Despegue:** Inicio de la misión con el despegue del dron.
2. **Aproximación:** El dron se acerca al aerogenerador deseado hasta posicionarse a una distancia definida.
3. **Posicionamiento ortogonal:** Ajuste de la posición del dron para posicionarse ortogonal al aerogenerador y con el rotor centrado en la imagen.
4. **Primera inspección de las aspas:** Se realiza la inspección de las superficies de las aspas que están orientadas hacia el frente del dron.
5. **Rotación de las aspas por parte del operador:** El operador realiza un giro de pala, exponiendo una nueva superficie para la inspección.
6. **Segunda inspección de las aspas:** Se lleva a cabo la inspección de las nuevas superficies que quedaron al frente tras la rotación.
7. **Retorno a tierra:** El dron retorna al punto de partida.

Estas etapas se repiten para la inspección trasera. En la Sección 3.5 se proporciona un mayor detalle sobre el funcionamiento de la máquina de estados, así como los ajustes necesarios para ejecutar estas etapas en ambas inspecciones.

A continuación, se describen las soluciones implementadas para abordar los desafíos específicos encontrados durante el desarrollo, y cómo estas se integraron para conformar la solución final del sistema.

3.2. Planificador de Caminos

El desarrollo del planificador de rutas se fundamenta en gran medida en el trabajo presentado por Amer et al. 2023. Este módulo es utilizado en la etapa de la inspección de las aspas del aerogenerador, detallada en la Subsección 3.5.6, y se ejecuta con el objetivo de determinar el orden y la dirección en que el dron debe inspeccionar las aspas del aerogenerador.

Es importante destacar que este planificador de caminos considera únicamente las superficies de las aspas que están orientadas hacia el frente del dron. Cada aspa tiene cuatro superficies que deben

²<https://gitlab.fing.edu.uy/bruno.centanaro/tesis-aerogeneradores-ws>

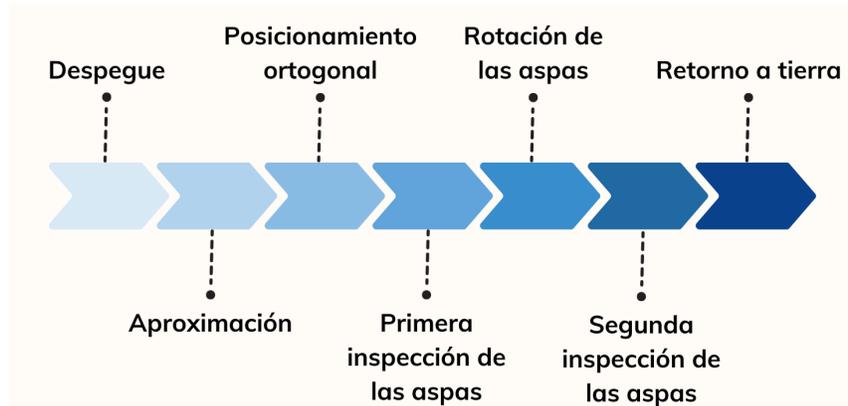


Figura 3.1: Etapas principales de la misión realizadas por el dron durante la inspección de aerogeneradores: despegue, aproximación al aerogenerador, posicionamiento ortogonal, primera inspección de las aspas, rotación de las aspas por parte del operador, segunda inspección de las aspas y retorno a tierra.

inspeccionarse de manera independiente (borde de ataque y de fuga, superficie superior e inferior). Por este motivo, en la solución final, se prevé que el planificador se ejecute un total de cuatro veces a lo largo de la misión, asegurando que en cada pasada se inspeccione una de las superficies de cada aspa. A continuación, se describe la estructura y funcionamiento del planificador de rutas.

3.2.1. Generación de la malla de las aspas

En primer lugar, se elabora una representación simplificada de las aspas del aerogenerador a través de una malla de triángulos. Una malla de triángulos es una estructura compuesta por múltiples triángulos interconectados que permite modelar objetos en 3D de manera eficiente. Para esto, se utilizan como entradas el largo del aspa (l_A) y el diámetro del rotor (d_R). A partir de estas dimensiones, se generan tres paralelogramos de dimensiones $l_A \times d_R$, dispuestos en una configuración de Y invertida que representa la superficie frontal de las aspas hacia el dron. Como puede verse en la Figura 3.2, la generación de la malla da como resultado un archivo STL (formato ampliamente utilizado para la representación de modelos y geometría tridimensional) que será utilizado en la siguiente etapa del planificador.

Es importante señalar que el ancho del aspa no es un factor relevante en este contexto ya que, a la hora de calcular el trayecto que realizará el dron, se toman en cuenta únicamente los extremos de la mediana paralela al lado de largo l_A del paralelogramo. Esto se debe a que el aspa mantiene constante la ubicación de su mediana independientemente de su ancho. En consecuencia, los puntos de referencia utilizados para la planificación de la trayectoria no varían según el ancho de la misma.

Además, cabe mencionar que únicamente se consideran los extremos del aspa al planificar la trayectoria ya que se asume que las aspas tienen un ancho razonable y que el dron se posicionará a una distancia segura del aerogenerador. Este enfoque permite recorrer el aspa de un extremo al otro en línea recta, lo que debería ser suficiente para abarcar toda la superficie y que el dron pueda captar la totalidad de la misma en la imagen.

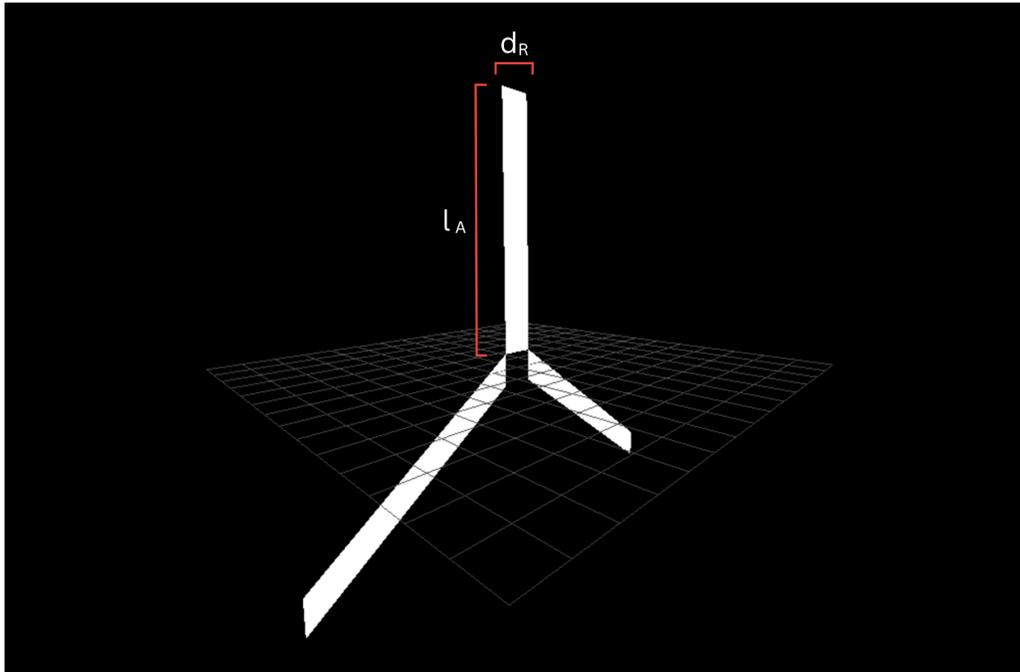


Figura 3.2: Captura de pantalla de un visualizador de STL, mostrando la malla de triángulos generada para las aspas del aerogenerador dispuestas en Y invertida, utilizada para la planificación del trayecto del dron. Se indican las dimensiones l_A y d_R , correspondientes al largo del aspa y al diámetro del rotor respectivamente

De no ser así, sería necesario considerar un mayor número de puntos. En tal caso, se podría planificar un recorrido más complejo, como un patrón en zigzag, por ejemplo, con el fin de inspeccionar exhaustivamente la totalidad de la superficie del aspa. Sin embargo, al no ser un escenario realista, se optó por simplificar el modelo, como se mencionó anteriormente.

Por otro lado, vale la pena mencionar que el proceso de generación de la malla es altamente personalizable, lo que permite adaptarlo a cualquier configuración inicial deseada para el aerogenerador. Si bien en este caso se ha optado por la disposición en Y invertida, debido a que constituye una precondition esencial para llevar a cabo satisfactoriamente la etapa de posicionamiento ortogonal de la misión, se podría optar por otra.

3.2.2. Planificador de Rutas de Inspección del Aerogenerador

A partir de la malla de las aspas generada, el objetivo de este planificador es generar una secuencia óptima de puntos de inspección, minimizando la distancia total recorrida. Para ello, el algoritmo de planificación de rutas consta de tres pasos: agrupamiento, generación del grafo y ordenamiento.

En el proceso de agrupamiento, primero se extraen las posiciones de los vértices que forman cada triángulo de la malla a partir del archivo STL, y se calcula el centroide de cada triángulo a partir de estos. Con base en la posición de cada centroide, se asigna cada vértice de la malla a uno

de tres grupos, donde cada uno representa una de las aspas del aerogenerador. Para esto se utiliza el algoritmo K-Means Constrained (ver Anexo 5.5.2 y 5.5.3), con las restricciones de que el número de clusters debe ser igual a la cantidad de aspas, y que cada cluster debe contener una cantidad de puntos igual a $\frac{\text{totalidad de los puntos}}{\text{cantidad de aspas}}$.

En el segundo paso, por cada uno de los grupos de puntos definidos, se extraen los nodos correspondientes a la raíz y la punta de las aspas, lo que resulta en un total de seis nodos.

El objetivo es planificar un recorrido que, partiendo desde una posición inicial (y sin volver a ella), pase por los seis nodos definidos una sola vez, minimizando la distancia total recorrida. Además, en caso de ser necesario, como lo es en la última fase de la inspección, el recorrido debe finalizar en una posición final distinta de los nodos previamente mencionados.

Para lograr esto, se construye un grafo en el cual los nodos son los puntos obtenidos de las superficies de las aspas, junto con la posición inicial del dron, y, si es requerido, la posición final. Cada arista entre los nodos tiene un costo asociado que depende de la distancia entre ellos.

A su vez, el planificador de caminos acepta dos ángulos como parámetros (ver Figura 2.10 para identificarlos):

1. **El giro de fase actual de las aspas del aerogenerador:** este permite corregir el error que se produciría si las aspas no estuvieran en posición de Y invertida.
2. **El giro de orientación del aerogenerador respecto al dron:** este ajusta el error que surgiría si el dron no estuviera alineado ortogonalmente al plano de las aspas.

Con esta información y mediante cálculos basados en matrices de transformación de puntos, se recalculan las posiciones de los seis nodos para representar con precisión la ubicación de las aspas en relación con el dron. Si estos ángulos no se incorporan y el aerogenerador no está exactamente en posición Mercedes ni ortogonal al dron, los puntos generados podrían no representar correctamente los extremos de las aspas.

En función de lo mencionado, el problema se formula como un Problema del Viajante Asimétrico (asymmetric TSP o ATSP). En el contexto del ATSP, un grafo $G(V, E)$, está compuesto por un conjunto de nodos V y un conjunto de aristas E , donde cada arista conecta dos nodos. En el caso del Problema del Viajante Asimétrico, los costos de las aristas no son simétricos, es decir, el costo de ir de un nodo v_i a v_j puede diferir del costo de ir de v_j a v_i . El objetivo del ATSP es encontrar el camino más corto que recorra todos los nodos exactamente una vez, considerando los costos asimétricos entre los nodos. Ver Anexo 5.5.1 para más información sobre ATSP.

Para definir el costo de las aristas entre los nodos pertenecientes al aerogenerador, se utiliza el siguiente criterio:

- Si pertenecen a grupos diferentes, el costo es la distancia euclidiana entre ellos.
- Si los nodos pertenecen al mismo grupo, el costo es 0. Esto asegura que el camino incluya las tres aristas necesarias para recorrer cada aspa, evitando que el dron salte entre superficies sin inspeccionarlas completamente.

Además de los nodos correspondientes al aerogenerador y los costos asociados entre ellos, es fundamental considerar tanto la posición de partida, que será el punto inicial del algoritmo, como la posición final, que se utilizará en la última fase de la inspección. Esto es importante porque las distancias entre la posición inicial y el primer nodo del recorrido del aerogenerador, así como la distancia entre el último nodo de este recorrido y la posición final, también forman parte del

trayecto total y pueden influir en la solución óptima. Los costos de las aristas que involucran estos dos nodos específicos se calculan de la siguiente manera:

- El costo desde la posición inicial hacia cualquier otro nodo consiste en la distancia euclidiana. En cambio, el costo desde cualquier nodo hacia la posición inicial es 0.
- Si se especifica una posición final, el costo para ir desde esta hacia cualquier otro nodo es infinito, mientras que el costo desde cualquier nodo hacia la posición final es su distancia euclidiana.

En el TSP clásico, se considera el costo de regresar al nodo inicial. Sin embargo, al asignar un costo de 0 para este regreso, se mantiene constante el costo del recorrido de vuelta a la posición inicial, independientemente del nodo final seleccionado. Esto resulta beneficioso para el caso en cuestión, ya que, al no requerirse que el dron regrese al punto de partida, se evita penalizar aquellas soluciones cuyo nodo final se encuentre más alejado de este.

Por otro lado, al asignar un costo infinito desde el nodo final hacia cualquier otro, garantizamos que este sea el último en el recorrido del algoritmo. Si el algoritmo intentara pasar por el nodo final antes de terminar, sumaría un costo infinito, lo que asegura que ninguna solución válida lo ubique en otra posición que no sea la última.

Finalmente, el problema se resolvió utilizando el algoritmo TSP implementado en la librería Google OR-Tools, una librería de código abierto diseñada para resolver problemas complejos en áreas como el enrutamiento de vehículos, la programación lineal, entre otras (ver Anexo 5.4.7 para más información). Se logró obtener una solución adecuada configurando el parámetro *first_solution_strategy* como *PATH_CHEAPEST_ARC*, sin necesidad de un refinamiento adicional.

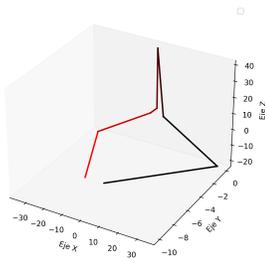
Para resolver el TSP, se proporcionó como entrada el grafo generado junto con el índice correspondiente a la posición inicial. El resultado fue un recorrido óptimo por los nodos, excluyendo el último nodo que representa el retorno al punto de partida, ya que, como fue mencionado, este no era relevante para el caso en cuestión.

Las imágenes en la Figura 3.3 muestran distintos trayectos generados por el planificador de caminos, donde se utiliza una escala de colores rojos para indicar el orden en que el dron pasa por cada arista entre los puntos, comenzando desde el tono más claro. Además, presentan el resultado de utilizar los distintos parámetros soportados por el algoritmo creado: punto de inicio, punto de fin, giro de fase y orientación del aerogenerador con respecto al dron.

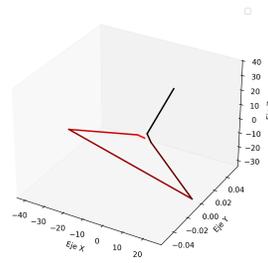
En la Figura 3.3a, se visualiza el trayecto con puntos de inicio y fin definidos. La Figura 3.3b muestra el trayecto generado al especificar el giro de fase, mientras que la Figura 3.3c presenta el trayecto considerando el giro de orientación. La Figura 3.3d muestra la primera fase, donde el dron parte desde el centro del aerogenerador sin un destino final definido. Por último, la Figura 3.3e presenta la segunda fase, donde el dron inicia desde la punta del aspa vertical y sigue sin un punto final definido. En esta última, el dron comienza desde la punta del aspa vertical, ya que este es el punto en el que queda tras completar la primera fase, como se muestra en la Figura 3.3d.

3.2.3. Evolución del Planificador de Caminos

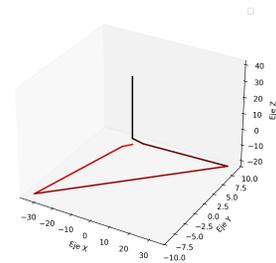
Inicialmente, se había considerado realizar la inspección en una única fase, en la cual el dron recorrería todas las superficies de las tres aspas en una sola pasada, sin la intervención del operario. El diseño del planificador de caminos se inició con este enfoque, pero se simplificó para ajustarse al resultado final. Si la inspección se hubiera hecho en una sola pasada, el procedimiento del planificador de rutas habría sido similar al utilizado, pero con algunas diferencias clave.



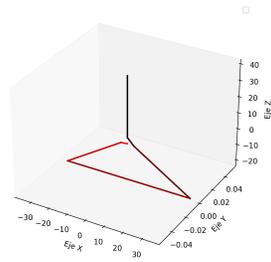
(a) Camino generado para un ejemplo donde se especifican tanto un punto de partida como un punto final.



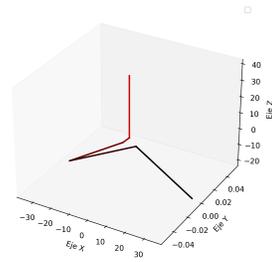
(b) Camino generado para un ejemplo donde se especifica un giro de fase de 20 grados.



(c) Camino generado para un ejemplo donde se especifica un giro de orientación de 20 grados.



(d) Camino generado para la primera fase de la solución, con el centro del aerogenerador como punto de partida y sin punto final definido.



(e) Camino generado para la segunda fase de la inspección sin un punto final definido.

Figura 3.3: Las imágenes presentan cinco trayectos distintos generados a partir del planificador de caminos. En cada una de ellas, se utiliza una escala de color rojo para indicar el orden en que el dron pasa por cada arista entre los pares de puntos, comenzando desde el tono más claro y finalizando en el más oscuro.

Primero, la generación de la malla del aerogenerador habría incluido todas las superficies de las tres aspas (doce en total), no solo las frontales. Además, el ancho y grosor de las aspas habrían sido relevantes para la generación de la malla, ya que habría sido necesario pasar de una cara a otra.

Durante la etapa de agrupamiento, habría sido de interés registrar la normal de los puntos para cada grupo detectado (uno por aspa). Al igual que los puntos, las normales se habrían extraído directamente del archivo STL mediante una librería basada en NumPy. Estas, al construir el grafo de costos, hubiesen servido para penalizar transiciones que implicasen un cambio de normal y prohibir transiciones entre superficies con normales opuestas.

El TSP se habría resuelto de manera similar, utilizando la misma librería de OR-Tools, con un total de veinticuatro nodos en el modelo (uno en cada extremo de cada lado de cada aspa). El TSP es un problema NP-completo, lo que implica que no se conoce un algoritmo eficiente que lo resuelva en tiempo polinomial respecto al tamaño de la entrada definida por $G(V, E)$. Existen diversas estrategias para abordar este problema (Fuentes 2024):

1. Fuerza bruta: Se enumeran todas las soluciones posibles, calculando el costo de cada una y

seleccionando la mejor. Esta solución tiene una complejidad de $O(n!)$, lo que la hace inviable para grandes cantidades de nodos. Esta complejidad corresponde al peor caso, que ocurre cuando el grafo G es completo, es decir, cada nodo está conectado con todos los demás. En este caso, el número de soluciones posibles es igual al número de permutaciones de n nodos, es decir, $n!$.

2. Métodos exactos mejorados: Algoritmos como *Branch and Bound* (B&B), *Branch and Cut* (B&C), o el algoritmo de *Held-Karp*, que aunque sigue siendo exponencial $O(n^2 \cdot 2^n)$, reduce la complejidad en comparación con la fuerza bruta, proporcionando una solución exacta.
3. Heurísticas: Métodos que ofrecen buenas soluciones en tiempos de cómputo reducidos, sin garantizar que sean las óptimas. Ejemplos incluyen los algoritmos *Nearest Neighbor* $O(n^2)$ y *Ant Colony Optimization* $O(m \cdot n^2 \cdot t)$.

En la solución en la que toda la inspección se realizaba en una sola fase, los mejores resultados, que a su vez minimizaban el tiempo de espera del dron en el aire, se obtuvieron al configurar la librería la Google OR-Tools con la *first_solution_strategy* como *PATH_CHEAPEST_ARC* y la *local_search_metaheuristic* con el valor *GUIDED_LOCAL_SEARCH*, con un tiempo límite de cinco segundos debido a que el dron estaría esperando en el aire.

La espera del dron en el aire se debía a la ejecución “online” del algoritmo, es decir, a la ejecución del planificador de caminos mientras el dron estaba en vuelo. Esta modalidad permite que el algoritmo inicie desde la posición real en la que se encuentra el dron tras completar la fase de alineación ortogonal al rotor. Este enfoque puede afectar positivamente los resultados del cálculo y es especialmente ventajoso si la alineación inicial no deja al dron perfectamente posicionado frente al rotor, ya que de no hacerse “online”, se debería suponer que el mismo quedará bien posicionado.

En contraste, la ejecución “offline” se realiza antes del despegue, lo que elimina la necesidad de cálculos en pleno vuelo. En esta, se deberá asumir que el dron se posicionará adecuadamente frente al rotor. Las principales ventajas de esta modalidad son la conservación de la batería, ya que el dron no consume energía durante la espera de la resolución, y permitir un mayor tiempo límite de cálculo, lo cual generaría soluciones potencialmente más óptimas.

Sin embargo, en nuestra solución integral, la elección entre ejecución “online” u “offline” resultó ser prácticamente indiferente. Esto se debe a que, por un lado, las ventajas de la ejecución “online” no siempre se materializan, ya que el proceso de alineación del dron suele dar buenos resultados. Por otro lado, la ventaja de la ejecución “offline” tampoco era significativa, ya que el límite de cinco segundos no impactaba negativamente en el rendimiento del proceso ni en los recursos del dron. A su vez, el tiempo límite de cinco segundos demostró dar buenos resultados, incluso en comparación con límites mayores.

En consecuencia, se optó por la ejecución “online”, ya que simplificaba la implementación y mantenía un equilibrio adecuado entre eficiencia y simplicidad.

De todas formas, para la solución final, al reducir el problema para que el dron solo inspeccionara tres superficies por pasada, la cantidad de nodos se redujo considerablemente y el tiempo de resolución dejó de ser un inconveniente, pasando a resolverse de manera instantánea. Esto eliminó las diferencias visibles entre la ejecución “online” y “offline”, y por simplicidad se optó por la primera. En este escenario, incluso un enfoque de fuerza bruta habría producido resultados satisfactorios debido al menor tamaño del problema. No obstante, se decidió mantener el uso de OR-Tools, ya que esta solución permite una mayor adaptabilidad para futuras expansiones o reutilización del código en otros contextos, asegurando buenos resultados independientemente de la cantidad de nodos.

Se destaca que esta simplificación del planificador de caminos surgió a partir de la observación del trabajo de los operarios durante la investigación de campo, que, como se mencionó anteriormente, se rotaban las aspas para que las superficies a inspeccionar estuviesen siempre de frente al sol.

3.3. Control del dron

Para controlar al dron, se investigaron las plataformas de control de vuelo de código abierto PX4³ y ArduPilot⁴, ambas reconocidas por su amplio uso en sistemas autónomos. Tras realizar pruebas con ambas, PX4 brindó un control preciso del dron y sin mayores complicaciones. En contraste, ArduPilot presentó varios desafíos que dificultaron alcanzar el nivel de control requerido. Debido a estas razones, se optó por PX4 como el firmware a utilizar para cumplir con los objetivos planteados. Para más información sobre PX4, ver Anexo 5.4.5.

Para gestionar el control del dron, se desarrolló un paquete dedicado exclusivamente a esta función en ROS 2, un framework de middleware ampliamente utilizado para aplicaciones robóticas (ver Anexo 5.4.1). La comunicación entre el nodo de ROS 2 y PX4 se lleva a cabo a través de tópicos de ROS 2, lo que facilita la integración y el control del dron desde el sistema. Este paquete define diversas funciones responsables de ejecutar acciones sobre el dron, como despegar, navegar a puntos específicos, rotar manteniendo el centro, entre otras, que se comunican con el dron mediante mensajes predefinidos de PX4 para ROS 2⁵.

Internamente, PX4 emplea un sistema de mensajería denominado *uORB*, que gestiona la comunicación entre los distintos módulos del firmware. Para habilitar la interoperabilidad con ROS 2, PX4 incorpora un puente llamado *Micro RTPS Bridge*, que traduce los mensajes entre *uORB* y ROS 2. De este modo, los mensajes publicados en los tópicos de ROS 2 son recibidos por PX4, y viceversa, lo que garantiza una comunicación bidireccional eficiente.

3.3.1. Modo de Vuelo Offboard

El modo de vuelo utilizado es el denominado *offboard*⁶. Este modo permite al vehículo seguir puntos de referencia de posición, velocidad o actitud que se le proporcionan a través de MAVLink o ROS 2. MAVLink (Micro Air Vehicle Link) es un protocolo de comunicación utilizado para intercambiar datos entre drones y estaciones terrestres o sistemas de control. Para más detalles sobre MAVLink, consultar el Anexo 5.4.4. Durante la operación, el nodo de control del dron emite continuamente el setpoint correspondiente a la posición deseada del dron y la velocidad con la que debe alcanzarla. En robótica, un setpoint es un punto de referencia dinámico que indica al sistema de control la posición, velocidad u orientación deseada en tiempo real, permitiendo un control preciso. Es esencial mantener una frecuencia mínima de mensajes de 2 Hz, ya que PX4 lo requiere para mantenerse en el modo mencionado. Si la frecuencia de los mensajes cae por debajo de este umbral, el dron cambiará automáticamente al modo *Return Home*.

³<https://github.com/PX4/PX4-Autopilot>

⁴<https://github.com/ArduPilot/ardupilot>

⁵https://docs.px4.io/main/en/msg_docs/

⁶https://docs.px4.io/main/en/flight_modes/offboard

3.3.2. Sistemas de Referencia

En PX4, el sistema de referencia global es el marco de coordenadas NED (North, East, Down), que se utiliza para ubicar al dron en el mundo. Este sistema tiene su origen $(0, 0, 0)$ en la posición inicial del dron establecida en el momento del arranque.

El sistema de referencia relativo, en cambio, se desplaza junto con el dron. En este sistema, los ejes X e Y están alineados con el cuerpo del dron, siendo X hacia adelante y Y hacia la derecha. Estos ejes se ajustan cuando el dron cambia su ángulo de guiñada. El eje Z, por su parte, coincide con el eje Z global, que apunta hacia la dirección de la gravedad. Esta elección de mantener el eje Z fijo y no relativo al dron se debe a que coincide también con la dirección vertical del aerogenerador. Esto facilita algunos cálculos, por ejemplo, durante la inspección de las aspas, permite mover el dron hacia arriba o hacia abajo (en el eje Z) manteniendo constante la distancia al aerogenerador. La utilización de este sistema relativo simplifica los cálculos y el control del dron, ya que los movimientos y acciones se calculan en función de su orientación actual. Ambos sistemas de referencia están ilustrados en la Figura 3.4.

Cuando el dron se mueve, es necesario transformar los comandos entre estos dos sistemas. El controlador recibe los setpoints en el sistema relativo y los convierte al sistema global NED que usa PX4, teniendo en cuenta la orientación del dron, para que los movimientos se procesen correctamente en el mundo.

Un ejemplo de este proceso son los setpoints obtenidos del planificador (Sección 3.2), estos se calculan en un sistema de referencia relativo al dron. El controlador, al momento de procesarlos, los transforma al sistema NED global, teniendo en cuenta la orientación actual del dron para luego ponerlos en el stack a procesar.

Para realizar esta transformación, se lleva a cabo una rotación de las coordenadas del sistema de referencia del dron, para pasarlas al sistema NED global. Esto se logra mediante una matriz de rotación que depende del ángulo de orientación del dron con respecto al norte verdadero. Dicha matriz se aplica a los vectores de posición y velocidad. Matemáticamente, esta transformación se expresa como sigue:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Donde $R(\theta)$ representa la matriz de rotación en función del ángulo θ de orientación del dron.

3.3.3. Gestión de Setpoints

Para gestionar los movimientos del dron, se decidió mantener un **setpoint actual**, tratando todas las acciones subsiguientes como cambios relativos a dicha posición. Se implementó un **stack de setpoints relativos agrupados**, en el cual se agrupan las acciones o movimientos basados en un mismo sistema de referencia relativo al dron. En el ejemplo de la Figura 3.5, se requiere que el dron visite los setpoints $\{p_1, p_2, p_3, p_4\}$ en orden, estos setpoints se calculan en función del sistema de referencia relativo a la posición del dron (x_1, y_1, z_1) al momento de calcular los puntos, sin considerar las rotaciones intermedias.

Asimismo, se implementó un **stack setpoints globales para el controlador**, que contiene los setpoints transformados al sistema de referencia NED global, listas para ser enviadas a PX4. Siguiendo con el ejemplo de la Figura 3.5, en este stack se guardan p_1, p_2, p_3 y p_4 una vez que son convertidos al sistema global (x_G, y_G, z_G) .

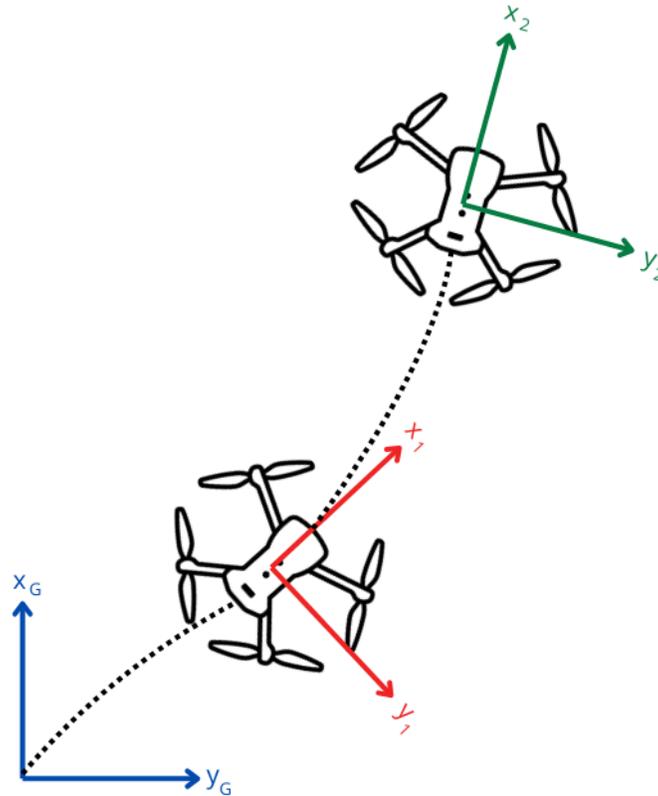


Figura 3.4: La imagen muestra el sistema de referencia global NED y dos sistemas de referencia relativos al dron en dos instantes de tiempo distintos, donde el dron tenía diferentes ángulos de guiñada y posición. En los tres sistemas, el eje z apunta hacia abajo, siendo perpendicular al plano de la hoja y extendiéndose hacia adentro, a lo largo de la tercera dimensión. Los ejes (x_G, y_G, z_G) corresponden al sistema de referencia global, mientras que los ejes (x_1, y_1, z_1) y (x_2, y_2, z_2) representan los sistemas de referencia relativos al dron.

Este segundo stack se consume durante la operación. Una vez que este stack de setpoints globales queda vacío, se toma el siguiente grupo de setpoints relativos del stack agrupado y se transforman al sistema global conforme a la orientación actual del dron. De esta manera, se garantiza que los movimientos se realicen de forma correcta, sin importar los cambios de orientación que el dron haya experimentado.

Es fundamental que la transformación al sistema de referencia global de nuevos setpoints se realice al finalizar de procesar y visitar cada grupo de setpoints, ya que esto asegura que el dron se encuentra en la posición y orientación adecuadas para iniciar el siguiente conjunto de movimientos. Esto resulta esencial para definir correctamente la dirección frontal del dron en el sistema de referencia global y para que los comandos enviados a PX4 se interpreten correctamente.

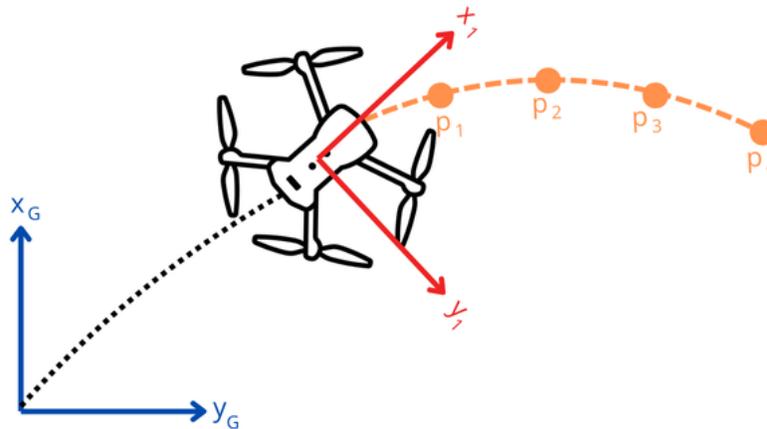


Figura 3.5: La imagen muestra el sistema de referencia global (x_G, y_G, z_G) y un sistema de referencia relativo al dron (x_1, y_1, z_1) . En ambos sistemas, el eje z apunta hacia abajo, siendo perpendicular al plano de la hoja y extendiéndose hacia adentro, a lo largo de la tercera dimensión. Se denotan los puntos $\{p_1, p_2, p_3, p_4\}$ a ser visitados en orden por el dron. Estos puntos se calculan en el sistema de referencia relativo, pero luego deben ser transformados al sistema de referencia global.

3.3.4. Control de Orientación (Yaw)

El control del *yaw* es un aspecto clave en la operación del dron. Al incluir la orientación deseada en los *setpoints* enviados, se garantiza que el dron mantenga la dirección adecuada en todo momento. Este control es especialmente importante para tareas donde los sensores o cámaras deben enfocarse continuamente en un punto de interés, como ocurre durante la inspección de las aspas.

3.3.5. Publicación y Suscripción de Tópicos

El nodo de control del dron publica mensajes en tópicos específicos, como por ejemplo `/fmu/in/-trajectory_setpoint`, utilizado para enviar los *setpoints* de posición, velocidad y orientación. También se suscribe a determinados tópicos de PX4 que proporcionan información clave para el sistema desarrollado. Algunos de estos son:

- `/fmu/out/vehicle_local_position`: Proporciona la posición y velocidad locales del dron.
- `/fmu/out/vehicle_global_position`: Ofrece la posición global (GPS) y la altitud.

La información que se obtiene de ambos es vital para ajustar los cálculos realizados y asegurar que los comandos enviados son adecuados para el estado actual del dron.

La Figura 3.6 muestra un diagrama de los nodos de ROS 2 que se utilizaron junto a las relaciones entre los mismos a través de tópicos.

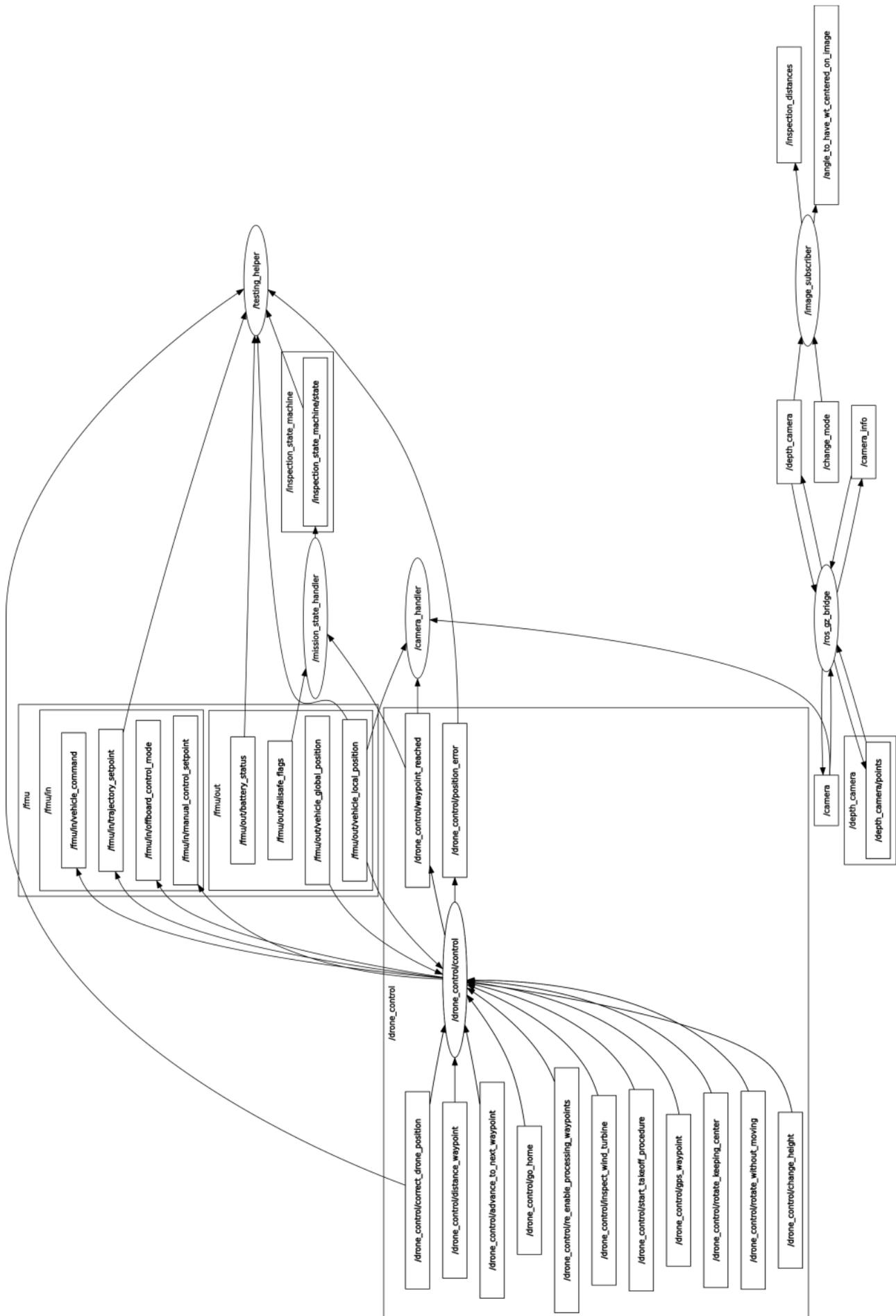


Figura 3.6: Diagrama que muestra los nodos de ROS 2 utilizados para la solución.

3.3.6. Posición Local del Vehículo y EKF2

La **Posición Local del Vehículo** representa una estimación de la posición y velocidad del dron dentro de un marco de referencia local, relativo al punto de despegue. Esta información es esencial para la navegación y el control del dron, ya que proporciona una ubicación confiable del dron durante toda la operación.

PX4 emplea el algoritmo de fusión de sensores conocido como **Filtro de Kalman Extendido 2 (EKF2)** para calcular tanto la posición como la orientación del dron (Ver Anexo 5.4.6). EKF2 integra datos de varios sensores y, mediante esta fusión, genera una estimación altamente acertada del estado del dron, que abarca los siguientes aspectos:

- **Posición y velocidad en el sistema NED:** Coordenadas relativas al origen local.
- **Orientación (roll, pitch, yaw):** Ángulos que definen la actitud del dron en el espacio.
- **Desvíos de sensores:** Correcciones aplicadas a errores sistémicos en los sensores.

El nodo de ROS 2 se suscribe al tópico `/fmu/out/vehicle_local_position`, publicado por PX4, para obtener la estimación de la posición y velocidad local generada por EKF2. Esta información resulta clave para las siguientes tareas:

- **Transformación de coordenadas:** Facilita la conversión entre sistemas de referencia locales y globales.
- **Correcciones en tiempo real:** Ayuda a la detección de desviaciones y la aplicación de ajustes inmediatos en la trayectoria.

Gracias a la integración de EKF2, se mejora la precisión en la determinación de la posición del dron.

3.3.7. Gestión de Estados Críticos

La gestión de estados críticos, como el armado (*arm*), desarmado (*disarm*), despegue y retorno al punto de lanzamiento, se realiza mediante la publicación de mensajes en el tópico `/fmu/in/vehicle_command`. Estos comandos son interpretados por PX4 y ejecutados en el dron.

3.3.8. Adaptabilidad y Correcciones en Tiempo Real

En caso de ser necesario ajustar la trayectoria, el sistema provee una manera de hacerlo para luego poder continuar con normalidad la operación. Se envía un setpoint de corrección que tiene prioridad sobre los setpoints en curso. Este setpoint permite realizar ajustes inmediatos en la trayectoria del dron si se detecta que la operación no está progresando según lo esperado. De esta forma, el sistema se adapta a variaciones sin interrumpir el flujo de control, lo cual es crucial en situaciones que demandan una respuesta inmediata, como evitar obstáculos imprevistos o modificar la ruta ante cambios en el entorno.

En aplicaciones específicas, como las inspecciones de las aspas del aerogenerador, se utiliza para ajustar la trayectoria buscando mantener el aspa completamente en la imagen y a una distancia adecuada. Al igual que el resto de setpoints, se emplean coordenadas NED y, al aplicar un setpoint

de corrección, se evalúa la posición y orientación actuales del dron, recalculando y ajustando su trayectoria en el sistema de referencia global.

Al activar el modo de corrección, el dron detiene el procesamiento de nuevos cambios de posición hasta recibir una confirmación externa que verifique si la corrección fue exitosa o si persisten errores en la posición. Esta medida fue implementada para evitar que el dron procese comandos pendientes sin asegurarse de que su estado es correcto antes de continuar. En caso de persistir errores, el sistema puede aceptar un nuevo setpoint de corrección para ajustar la trayectoria nuevamente. Si no se recibe la confirmación en un plazo de 20 segundos desde la interrupción, el procesamiento de nuevas posiciones se reanuda automáticamente.

3.3.9. Recepción de una Posición GPS

Otra de las funcionalidades clave desarrolladas en el nodo de control del dron es la capacidad de proporcionarle una posición GPS, permitiendo que se desplace a dicho punto o a una distancia determinada del mismo. Aunque PX4 ofrece soporte nativo para navegar a posiciones GPS, utilizar esta funcionalidad implicaría alternar entre diferentes modos de operación, como el modo Offboard y el modo Mission, lo cual puede complicar el control y aumentar el riesgo de errores durante la transición entre modos. Para evitar estos inconvenientes, se diseñó una solución que mantiene al dron en un punto relativo a su posición inicial (0, 0, 0), incluso al desplazarse a coordenadas GPS. Esto permite un control más consistente y simplifica la operación al eliminar la necesidad de cambiar de modo, reduciendo así la complejidad en el sistema.

Para llevar a cabo esta funcionalidad, al recibir una posición GPS, se procede a calcular tanto la distancia en metros desde la posición actual del dron hasta la posición GPS objetivo, así como el rumbo, que es el ángulo de dirección que el dron debe seguir para dirigirse hacia la posición objetivo.

Para calcular la distancia, se utiliza la librería `geopy`, específicamente la función `geodesic`, que permite obtener distancias precisas entre dos puntos geográficos.

Por otro lado, la determinación del ángulo α que el dron debe rotar se realiza mediante la fórmula del azimut. El azimut se define como el ángulo entre el norte geográfico y una línea de referencia, medido en sentido horario. Para calcular α , primero se obtiene el ángulo entre la dirección actual del dron y el norte, y luego el ángulo entre el norte y la línea que une la posición actual del dron con la posición objetivo (posición del aerogenerador). La resta de estos dos ángulos resulta en la rotación necesaria para que el dron apunte hacia la posición objetivo. En la Figura 3.7 se observan ambos ángulos y el ángulo α que el dron debe rotar.

Tras ajustar su orientación en base al rumbo, el dron avanzará en esa dirección con el fin de alcanzar el destino. Este proceso incluye la adición de un *setpoint* a la cola de comandos para la rotación necesaria, seguido de otro *setpoint* que indica el movimiento hacia adelante a lo largo de la distancia previamente calculada.

3.3.10. Ajuste de la Velocidad

PX4 únicamente permite establecer la velocidad con la que el dron debe alcanzar un setpoint, sin ofrecer una opción directa para controlar la velocidad a lo largo de toda una trayectoria. Sin embargo, este comportamiento puede lograrse al dividir la trayectoria en múltiples puntos intermedios⁷. De este modo, al configurar que el dron alcance cada uno de estos puntos con la misma

⁷<https://discuss.px4.io/t/offboard-mode-trajectory-setpoint/32850/2>

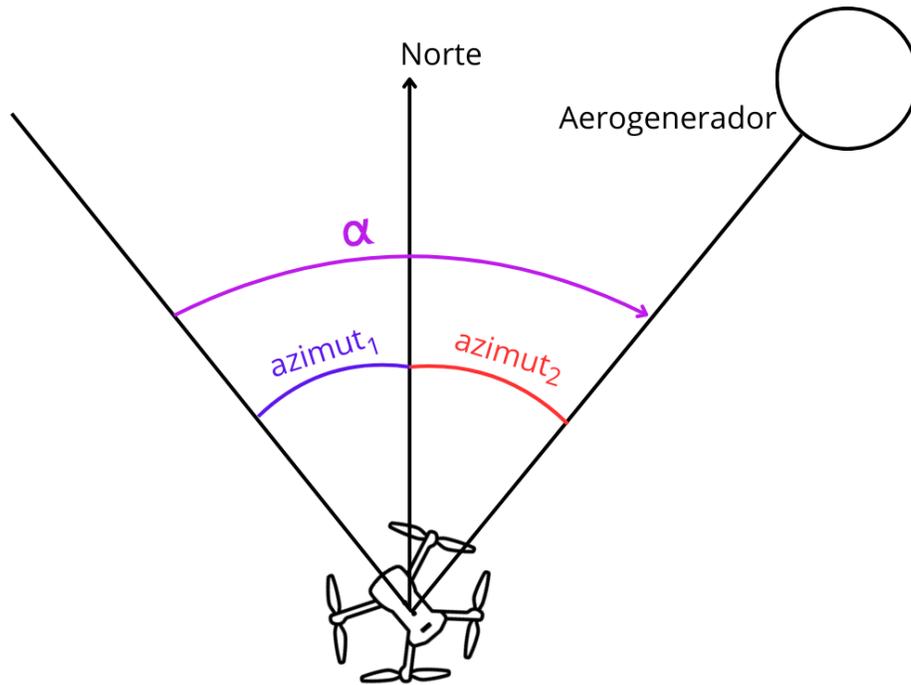


Figura 3.7: Diagrama ilustrativo del ángulo α que el dron debe rotar para quedar apuntando hacia el aerogenerador, resaltado en color violeta. Se muestra en azul el ángulo $azimut_1$, que representa la dirección actual del dron con respecto al norte, y el ángulo $azimut_2$, en rojo, que corresponde al ángulo que forma el norte con la línea que conecta la posición actual del dron con el aerogenerador.

velocidad, se consigue mantener una velocidad uniforme a lo largo de todo el trayecto, siempre y cuando se definan suficientes puntos intermedios.

Además, este enfoque presenta una ventaja adicional: ayuda a que el dron siga una trayectoria en línea recta. Esto contrasta con la ruta que podría generarse en ausencia de puntos intermedios, ya que PX4 se encarga de dirigir el dron al punto final con la velocidad establecida, pero no garantiza que esto se haga en un trayecto recto. La desviación puede ocurrir por diversos factores, como, por ejemplo, el viento, que empuja al dron fuera de la trayectoria ideal, especialmente en entornos al aire libre donde las condiciones climáticas pueden cambiar repentinamente. Otra posible causa es que PX4, al calcular la trayectoria hacia el setpoint final, determine que una ruta curva es más eficiente o rápida que una línea recta.

En nuestro caso, poder realizar un trayecto recto resulta crucial en tareas como la inspección de un aspa de aerogenerador. Por ejemplo, si se desea recorrer el aspa izquierda de un aerogenerador en configuración de Y invertida y el dron se encuentra en la raíz del aspa, al indicarle que debe llegar al otro extremo sin puntos intermedios que guíen su desplazamiento, no se garantiza que siga el recorrido deseado a lo largo del aspa. En cambio, podría desviarse de su trayectoria, por ejemplo, moviéndose hacia la izquierda y luego hacia abajo, en lugar de seguir una línea recta desde la raíz hasta la punta del aspa. La incorporación de puntos intermedios asegura un control más preciso del

desplazamiento y una mayor exactitud en tareas de inspección.

Para garantizar una trayectoria suave y evitar frenadas innecesarias, mientras el dron procesa un setpoint al que se debe llegar con una cierta velocidad, se analiza la dirección requerida para dirigirse hacia el siguiente punto. Para lograrlo, al procesar un nuevo setpoint, se compara la velocidad actual con la variación necesaria hacia el siguiente setpoint. Si todas las componentes de la variación tienen el mismo signo, se ajusta la velocidad utilizando una regla de tres, limitándola según la velocidad máxima previamente establecida. De este modo, el dron llega al setpoint con la velocidad adecuada para continuar hacia el siguiente setpoint sin detenerse. Esta velocidad máxima se estableció en 0.2 m/s debido a que mantiene un tiempo de inspección similar al trabajo de campo donde demoraba alrededor de 10 minutos por cara y, además, experimentalmente no tuvo problemas en las simulaciones. Cabe resaltar que esta velocidad máxima solo tiene efecto cuando se utilizan puntos intermedios y no cuando, por ejemplo, se va de un aspa a la otra.

En la Figura 3.8 se puede observar cómo el siguiente setpoint (s_{i+1}) tiene una velocidad v_{i+1} que permite seguir la trayectoria hacia s_{i+2} sin frenarse. Sin embargo, al llegar a s_{i+2} , el dron se frena porque mantener la velocidad previa lo alejaría de s_{i+3} y una velocidad que lo aproxime a este último podría implicar que no pase por s_{i+2} .

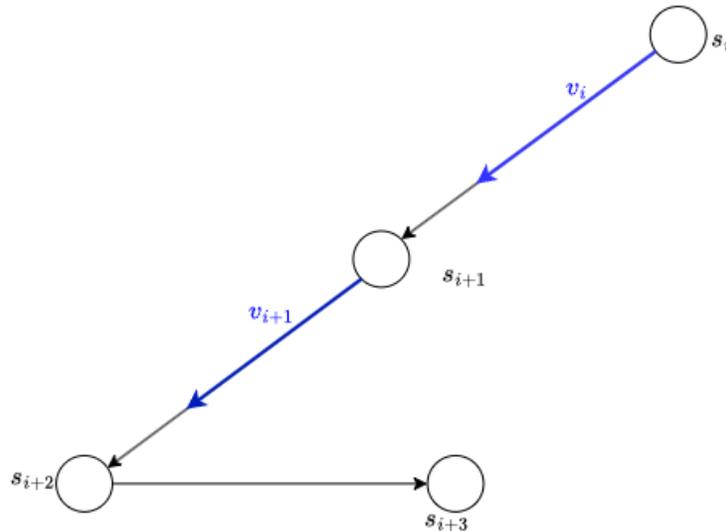


Figura 3.8: Se observa cómo se divide la trayectoria en puntos para controlar la velocidad, buscando que si los puntos están alineados, el dron no necesite frenarse.

3.3.11. Rotación Centrada en el Aerogenerador

Para lograr una rotación centrada en torno al aerogenerador, manteniendo una distancia constante al mismo, es necesario definir setpoints que interpolen la circunferencia. Si el dron se desplazara en línea recta en lugar de seguir una trayectoria curva, podrían surgir diversos problemas, dentro de los cuales se encuentra el riesgo de colisión.

Para realizar la interpolación de la trayectoria, se itera por la cantidad de grados a rotar y

se calculan puntos de referencia intermedios en los que el dron ajusta tanto su posición como su orientación. Este ajuste continuo garantiza que el aerogenerador se mantenga siempre centrado en la trayectoria del dron, evitando desviaciones y asegurando una rotación precisa alrededor del objeto. Un diagrama que ilustra este proceso se muestra en la Figura 3.9.

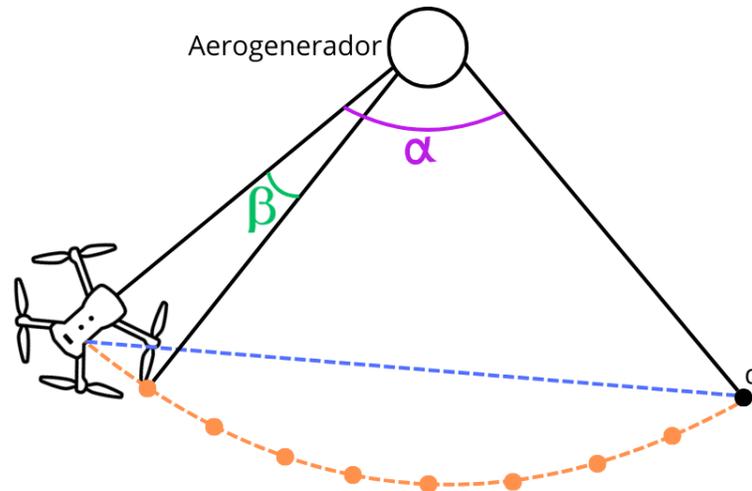


Figura 3.9: Diagrama que ilustra el proceso de rotación centrada en el aerogenerador a un ángulo α . En azul se muestra la trayectoria que el dron seguiría si fuera en línea recta al destino d , mientras que en naranja se representa la trayectoria curva con los puntos intermedios que el dron describe para mantener la rotación. Estos puntos intermedios son equidistantes y tienen un ángulo β entre sí.

3.3.12. Vuelta a Tierra

Al finalizar la inspección del aerogenerador, es necesario que el dron regrese de manera segura al punto de lanzamiento. Para llevar a cabo este proceso, se utiliza el modo de vuelo correspondiente proporcionado por PX4⁸. Para esto, se envía el comando `VEHICLE_CMD_NAV_RETURN_TO_LAUNCH` al dron a través del nodo de control en ROS 2, el cual instruye al dron a retornar al punto de lanzamiento de forma automática y controlada.

La activación de este comando se realiza mediante la publicación de un mensaje en el tópic `/fmu/in/vehicle_command`, configurado con los parámetros adecuados. Una vez recibido el comando, el dron inicia el procedimiento de retorno siguiendo la siguiente secuencia predefinida:

1. **Ascenso a Altura Segura:** Si el dron se encuentra volando por debajo de la altura segura preestablecida, asciende hasta alcanzar dicha altura, lo que busca garantizar la evitación de obstáculos durante el trayecto.
2. **Vuelo Horizontal al Punto de Lanzamiento:** El dron continúa su trayectoria horizontal en dirección al punto de lanzamiento.

⁸https://docs.px4.io/main/en/flight_modes_mc/return.html

3. **Descenso y Aterrizaje:** Al llegar horizontalmente al punto de origen, el dron desciende de manera controlada y aterriza automáticamente en el mismo.

Este procedimiento automatizado asegura una operación eficiente y segura durante la fase de retorno.

3.4. Reconocimiento del aerogenerador

3.4.1. Detección del aerogenerador en etapa de posicionamiento ortogonal

Para poder detectar el aerogenerador se sigue un enfoque similar a los mencionados en el estado del arte, se hace uso de la transformada de Hough que permite detectar líneas rectas en la imagen.

Antes de aplicar la transformada de Hough, se convierte la imagen a escala de grises y se aplica un filtro gaussiano para reducir el ruido y suavizar la imagen. Luego, se emplea el algoritmo de detección de bordes de Canny, que identifica los bordes al detectar cambios bruscos de intensidad en la imagen.

En particular, se emplea la transformada probabilística de Hough de la biblioteca OpenCV, ya que, aunque se probó también con la versión no probabilística, la primera ofreció mejores resultados.

Identificación de aspas y rotor

Se utiliza un algoritmo propio para identificar las aspas y calcular la posición aproximada del rotor.

El algoritmo utilizado busca una terna de líneas que tengan alrededor de 120° entre ellas, con cierto threshold, y obtiene las intersecciones de estas. El punto intermedio de las intersecciones es la posición estimada del rotor.

Incluso en el caso de que el aerogenerador no esté perfectamente en posición de Y invertida la detección de las aspas y el rotor siguen funcionando bien. Se realizaron pruebas rotando todo el aerogenerador, ya que no se puede girar solo las aspas del modelo de aerogenerador utilizado, y el resultado se muestra en las imágenes 3.10 y 3.11.

Se consideró la posibilidad de implementar un enfoque similar al descrito en Rao et al. 2019, el cual consiste en detectar primero la torre filtrando las líneas verticales de la imagen y luego buscar líneas cercanas al límite superior de la torre. Sin embargo, este método presentó una limitación significativa debido a que, al encontrarse el aerogenerador en una posición de Y invertida, no fue posible distinguir de manera precisa entre la torre y el aspa superior, ya que ambas están representadas por líneas verticales en la imagen.

Algunos descubrimientos interesantes respecto al desempeño de Hough para identificar el aerogenerador fueron los siguientes:

1. La luz tiene un impacto importante en la detección de líneas. Ver Sección 3.4.1: **Stream RGB vs imagen LIDAR.**
2. La efectividad del método depende del modelo del aerogenerador, cuanto más “curvas” sean las aspas mas difícil va a ser encontrar rectas. Como se observa en la Figura 3.12.

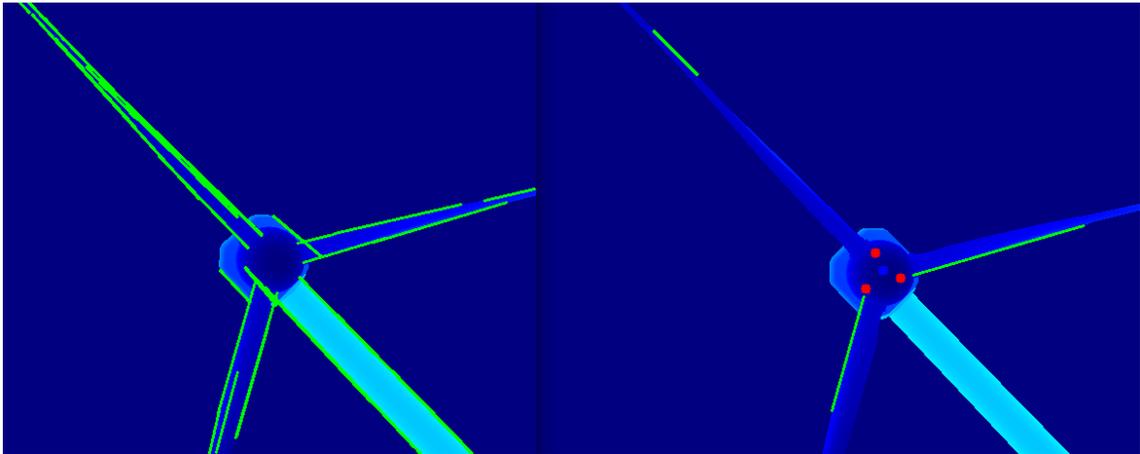


Figura 3.10: Detección de líneas con la transformada de Hough e identificación de las aspas y el rotor. Aerogenerador girado en sentido antihorario.

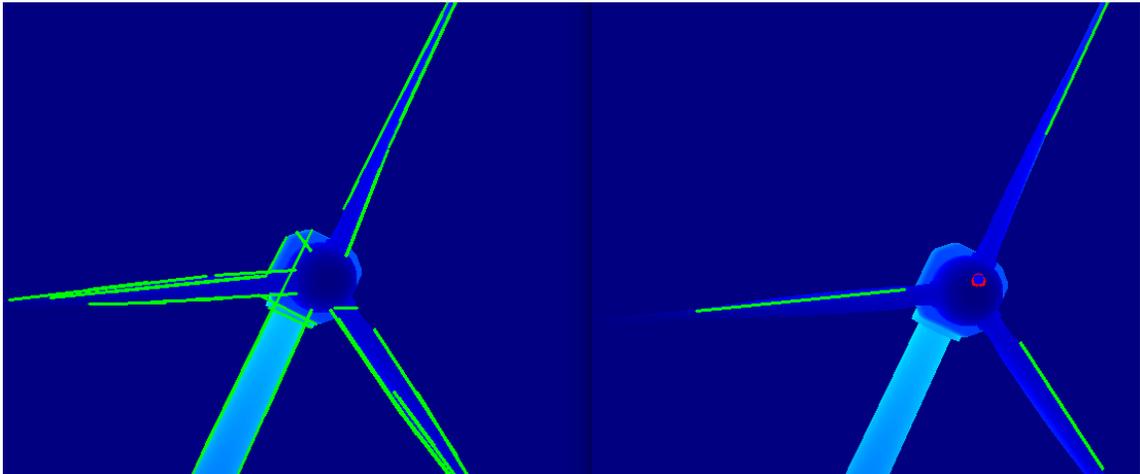


Figura 3.11: Detección de líneas con la transformada de Hough e identificación de las aspas y el rotor. Aerogenerador girado en sentido horario.

Stream RGB vs imagen LIDAR

El primer enfoque empleado, y el más directo, fue aplicar la transformada de Hough para detectar líneas en un stream RGB directo de la cámara, siendo este el método que primero viene a la mente. Sin embargo, este método presenta varios inconvenientes:

- La detección depende de los colores, lo que puede generar errores, por ejemplo cuando el aerogenerador se mezcla visualmente con las nubes o el horizonte.
- La efectividad también depende de las condiciones de iluminación. Para lograr un buen con-



Figura 3.12: Detección de líneas utilizando la transformada de Hough en dos modelos distintos de aerogeneradores.

traste con el fondo y detectar los bordes de la torre y las aspas, es necesario que la luz incida frontalmente sobre el aerogenerador.

Se exploraron alternativas para mitigar estos problemas, optando por un enfoque similar, pero utilizando las imágenes generadas por el LiDAR que se reciben a través de un tópicos de ROS 2 con el mensaje de tipo *Image*. Estas imágenes, planas y basadas en distancias, se normalizan y se convierten a escala de grises para obtener las líneas de Hough, mientras que se normalizan a color para la visualización y el cálculo del centroide (utilizado en el módulo de corrección).

Una de las principales ventajas de utilizar imágenes generadas mediante LiDAR es que permite segmentar la imagen en función a un rango de distancias deseadas, permitiendo centrar el foco en la región de interés y eliminar el ruido de fondo presente en la imagen RGB.

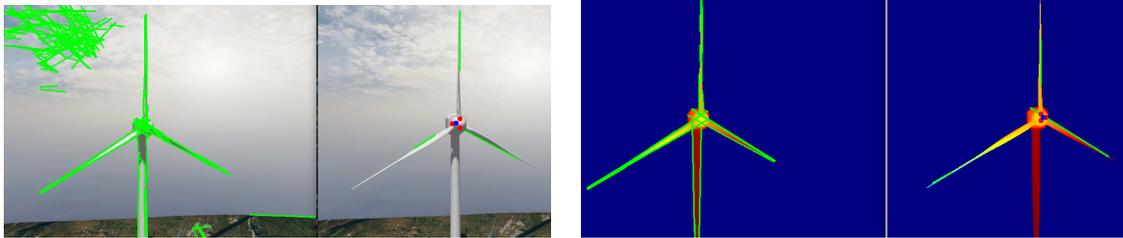
A continuación se muestran imágenes comparativas de las líneas detectadas con la transformada de Hough y de la identificación de aspas y rotor, usando stream RGB y LiDAR, con distintas condiciones lumínicas:

Las figuras 3.13 y 3.14 presentan imágenes comparativas que muestran la detección de líneas usando la transformada de Hough y la identificación de las aspas y el rotor en un flujo de datos RGB y LiDAR bajo diferentes condiciones de iluminación.

En la Figura 3.13, se observa la detección de líneas e identificación de las aspas y el rotor con iluminación frontal al aerogenerador.

En la Figura 3.14, se muestra la misma detección e identificación, pero en condiciones sin iluminación.

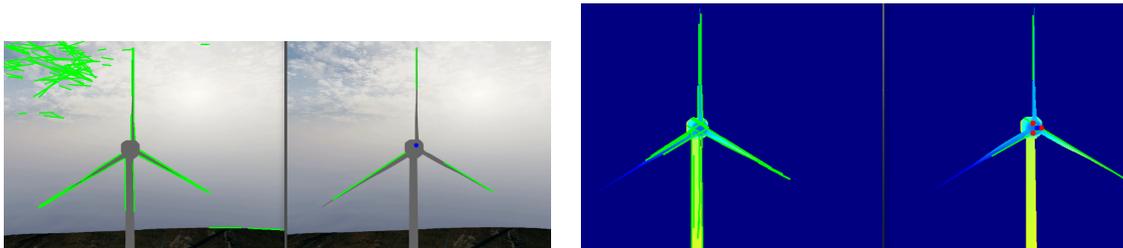
Finalmente, en la Figura 3.15, se exhibe la detección de líneas y la identificación con la iluminación que incide desde arriba al aerogenerador.



(a) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos RGB.

(b) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos LiDAR.

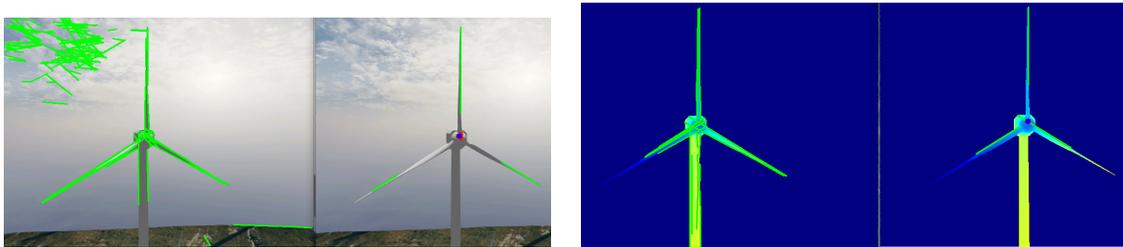
Figura 3.13: Detección de líneas con la transformada de Hough e identificación de las aspas y el rotor para distintos flujos de datos. La iluminación apunta de frente al aerogenerador.



(a) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos RGB.

(b) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos LiDAR.

Figura 3.14: Detección de líneas con la transformada de Hough e identificación de las aspas y el rotor para distintos flujos de datos. Sin iluminación.



(a) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos RGB.

(b) Detección de líneas e identificación de las aspas y el rotor mediante un flujo de datos LiDAR.

Figura 3.15: Detección de líneas con la transformada de Hough e identificación de las aspas y el rotor para distintos flujos de datos. La iluminación apunta desde arriba al aerogenerador.

Obtención de ángulo a rotar

Otra de las tareas a resolver es colocar el dron de manera ortogonal al plano de las aspas del aerogenerador y frente al rotor. Para esto se necesita corregir el ángulo del dron respecto al aerogenerador. Se necesita poder extraer el sentido en el que se debe rotar (horario o antihorario) y una medida del error, en función de los cuales se puede obtener el ángulo a rotar con el signo correcto.

Se intentó decidir el sentido de rotación en base a los ángulos entre las aspas, pero los resultados obtenidos no eran consistentes, cuando se estaba cerca de lograr ortogonalidad empezaba a indicar sentidos contrarios alternadamente lo que no permitía tomar una decisión.

Otro enfoque implementado con el que se obtuvo mejores resultados fue el siguiente: una vez encontrada la posición aproximada del rotor se pudo diferenciar las líneas verticales del aspa vertical de las líneas verticales de la torre usando el rotor como separación. Comparando las líneas del aspa vertical con las de la torre y su distancia se pudo obtener la información necesaria. Si bien los resultados no son tan buenos para casos cercanos a la ortogonalidad tiene la ventaja de que se puede aplicar sobre un flujo de datos RGB, es decir, podría utilizarse si no se tuviera LiDAR.

La solución implementada en la solución aprovecha la información de profundidad disponible. En primer lugar, se identifican las dos aspas inferiores del aerogenerador. A partir de esta identificación, se determina el punto en la imagen donde ambas aspas coinciden en su valor de y más bajo, lo cual corresponde al mayor valor de y en la imagen, ya que este aumenta hacia abajo. En ese valor de y , se mide la distancia del dron a cada aspa para su comparación.

En la Figura 3.16, se observan los puntos p_l (aspa izquierda) y p_r (aspa derecha). Si la distancia desde el dron a p_l es menor que la distancia a p_r , como es el caso en la figura, se debe girar en sentido antihorario; si ocurre lo contrario, el giro debe ser en sentido horario. La medida de error de orientación se obtiene como la diferencia en valor absoluto entre ambas distancias, ya que, cuando las distancias son iguales, se considera que el dron está ortogonal al aerogenerador.

Para abordar el caso en el que el aerogenerador no se encuentra perfectamente en una configuración de Y invertida, como se observa en la Figura 3.17, se define como línea vertical aquella que presenta la mayor pendiente en valor absoluto. Una vez identificada la línea vertical, las otras dos se asignan como el aspa izquierda y el aspa derecha, de acuerdo con el signo de sus pendientes. Esta metodología asegura que, incluso si el operario no coloca el aerogenerador exactamente en la posición de Y invertida, el algoritmo sigue funcionando correctamente, siempre que el ángulo de desviación α respecto a la configuración ideal no supere los 45 grados.

3.4.2. Detección de las aspas durante la inspección

De manera similar a la parte anterior, al principio se pensó en utilizar la imagen RGB y aplicando líneas de Hough buscar rectas paralelas y de esta manera identificar la posición de las aspas y así poder ir corrigiendo la posición.

A este método se le encontraron varios problemas:

- hay ruido, por ejemplo elementos en el fondo, ya que la imagen no está segmentada.
- es posible que no estén ambos bordes del aspa en la imagen, por lo que no se encuentran dos líneas paralelas con las cuales identificar el aspa.

Para solucionar estos inconvenientes, se optó por utilizar el sensor LiDAR para detectar puntos dentro de un rango específico. Durante la inspección frontal, se identifica la posición más cercana

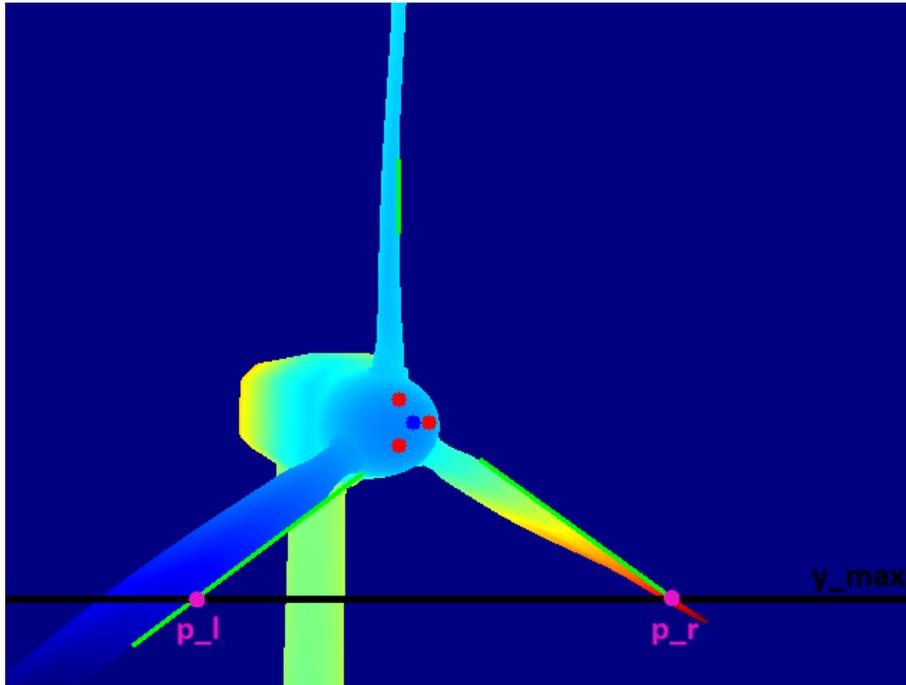


Figura 3.16: Detección de las aspas y el rotor en un aerogenerador inclinado sobre su eje. Se indican los puntos p_l y p_r , donde el algoritmo empleará los datos del LiDAR para comparar las distancias.

(correspondiente al aspa) y se filtran los puntos que se encuentran hasta 3 metros de distancia a partir de esta posición (ver Figura 3.18). Esto permite enfocarse en la parte de interés y descartar elementos como la base o la góndola del aerogenerador, que pueden aparecer al inicio de la inspección. En la inspección trasera, se toma el punto más lejano y se filtran los puntos dentro de los 6 metros a partir de este punto (ver Figura 3.19). Esta diferencia en los valores de umbral se debe a que, al tomar como referencia el punto más lejano en la inspección trasera, si se utilizara un valor similar al de la inspección frontal, solo se reconocerían las puntas de las aspas y no la zona de interés que se desea inspeccionar.

Una vez segmentada la imagen, se busca obtener un punto representativo de la posición del aspa para detectar su proximidad y determinar si está ubicada en algún borde de la imagen, con el objetivo de corregir la posición del dron y lograr una mejor inspección. Véase ejemplos de estos centroides en la Figura 3.20.

La primera aproximación para obtener este punto fue calcular el promedio de las coordenadas de los píxeles con valores de profundidad válidos (no nulos). Sin embargo, se encontró que este método presentaba un defecto, especialmente al inspeccionar la punta del aspa: el punto promedio podía indicar que el aspa estaba en el borde de la imagen, lo cual era cierto, pero esto era lo planificado y no requería una corrección.

Para resolver este problema, se decidió asignar una preferencia hacia los puntos cercanos al centro de la imagen, ya que es hacia donde el dron está orientado. Se calculan las distancias euclidianas de los píxeles con valores de profundidad válidos al centro de la imagen y se utiliza el inverso de

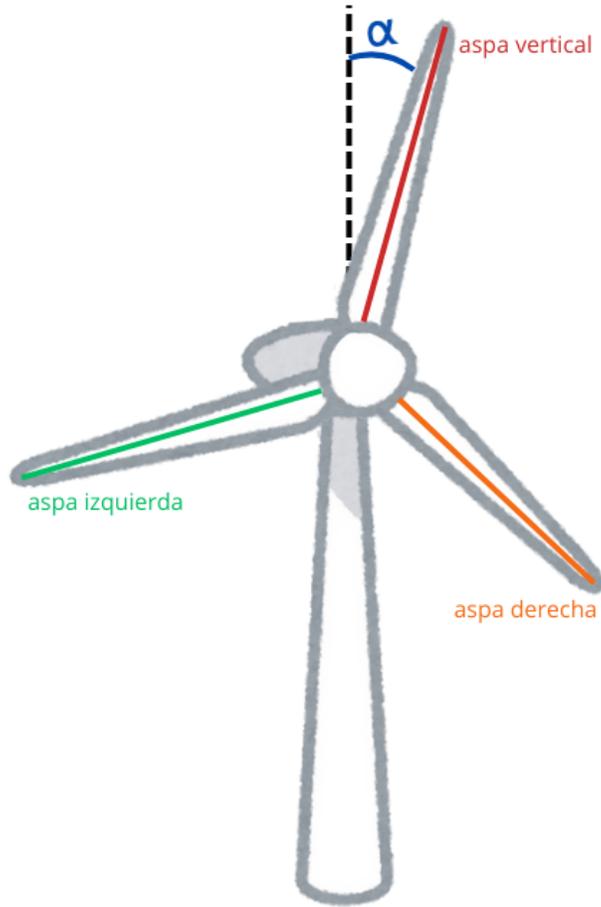
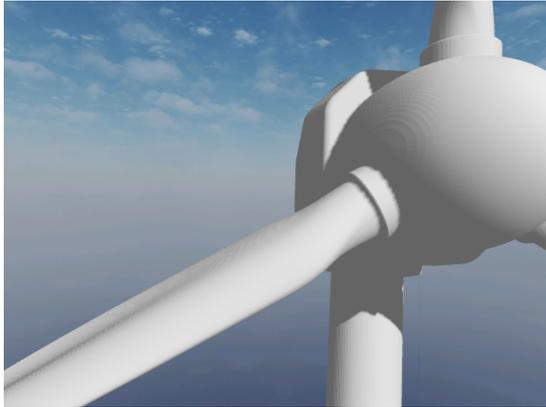


Figura 3.17: Aerogenerador casi en posición Y invertida, una de las aspas está desviada un ángulo α de la vertical. El aspa con la línea roja se reconocerá como aspa “vertical” por tener la mayor pendiente en valor absoluto. El aspa izquierda (en verde) tiene pendiente positiva, mientras que la derecha (en naranja) presenta pendiente negativa.

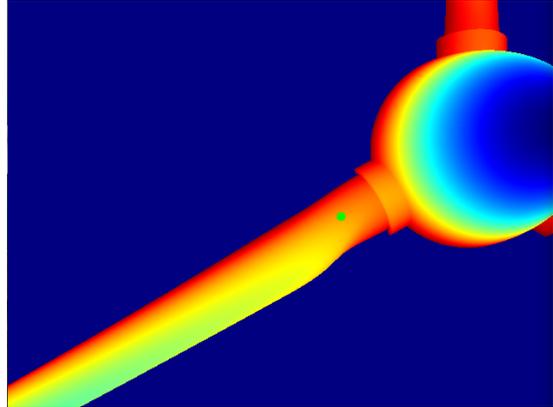
estas distancias como pesos para calcular un centroide ponderado. De esta manera, se priorizan los puntos más cercanos al centro, obteniendo un punto representativo más preciso de la posición del aspa en relación con el campo de visión del dron. La Figura 3.21 muestra la comparación entre el centroide utilizando este sesgo central y el obtenido sin sesgo, ilustrando claramente cómo esta modificación mejora la precisión, especialmente en el caso problemático de la punta del aspa.

3.4.3. Efecto del viento en la cámara

Durante las primeras inspecciones, se encontró un problema que no se había anticipado. El sistema funcionaba bien en condiciones sin viento, pero al introducir viento, aunque el dron mantenía su posición, la cámara/LiDAR, al estar fija, apuntaba en una dirección incorrecta. Esto sucedía tanto

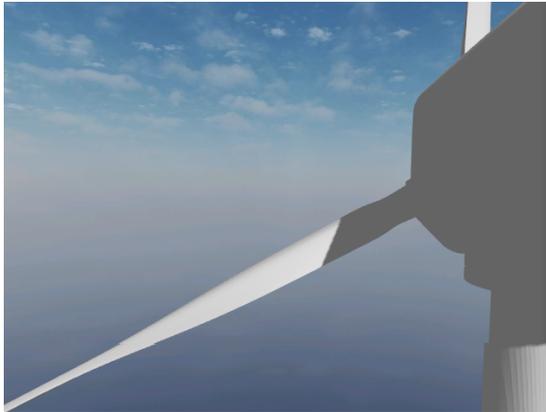


(a) Imagen de la cámara RGB.

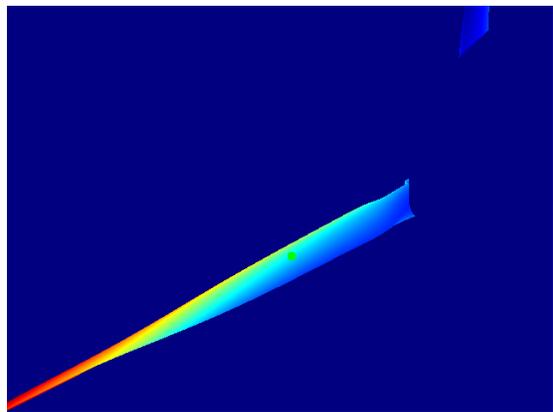


(b) Se muestra el centroide ponderado que prioriza los puntos cercanos al centro de la imagen para una detección más precisa de la posición del aspa.

Figura 3.18: Segmentación de las aspas en la inspección frontal utilizando un umbral de 3 metros desde el punto más cercano. Se puede observar en 3.18b como la torre del aerogenerador se filtra por el LiDAR.



(a) Imagen de la cámara RGB.

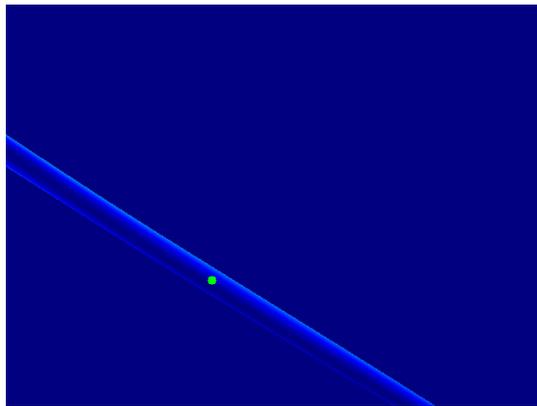


(b) Se muestra el centroide ponderado que prioriza los puntos cercanos al centro de la imagen para una detección más precisa de la posición del aspa.

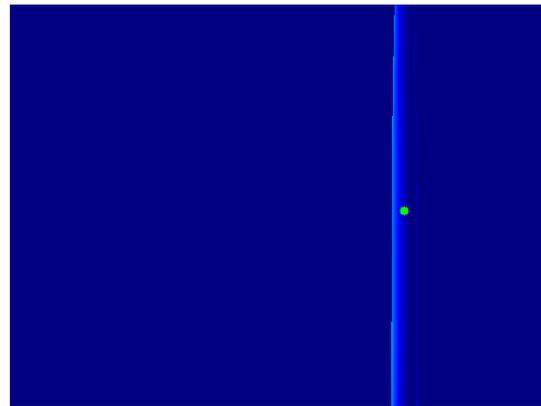
Figura 3.19: Segmentación de las aspas en la inspección trasera utilizando un umbral de 6 metros desde el punto más lejano. Se puede observar en 3.19b como la torre y la góndola del aerogenerador se filtran por el LiDAR.

en el eje vertical, cuando el viento venía de frente o detrás, como en el eje de rotación, cuando el viento era lateral.

Después de realizar pruebas en entornos ventosos, se identificaron varios problemas al utilizar

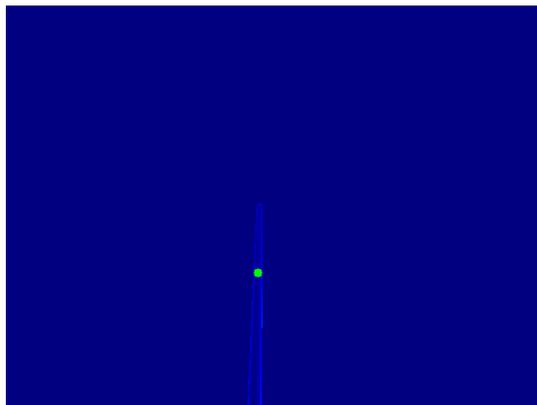


(a) Vista en inspección de aspas inferiores

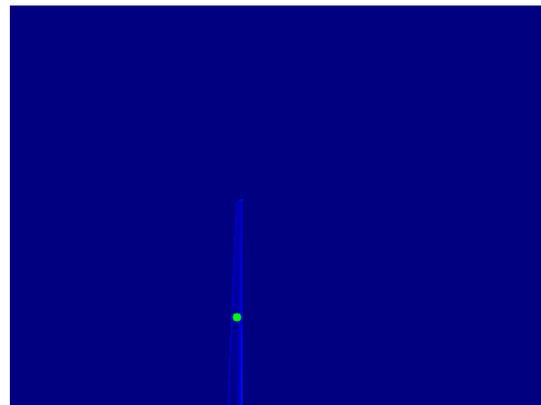


(b) Vista en inspección del aspa superior

Figura 3.20: Centroide en la inspección desde diferentes ángulos



(a) Centroide con sesgo central



(b) Centroide sin sesgo central

Figura 3.21: Comparación de métodos para obtener el centroide

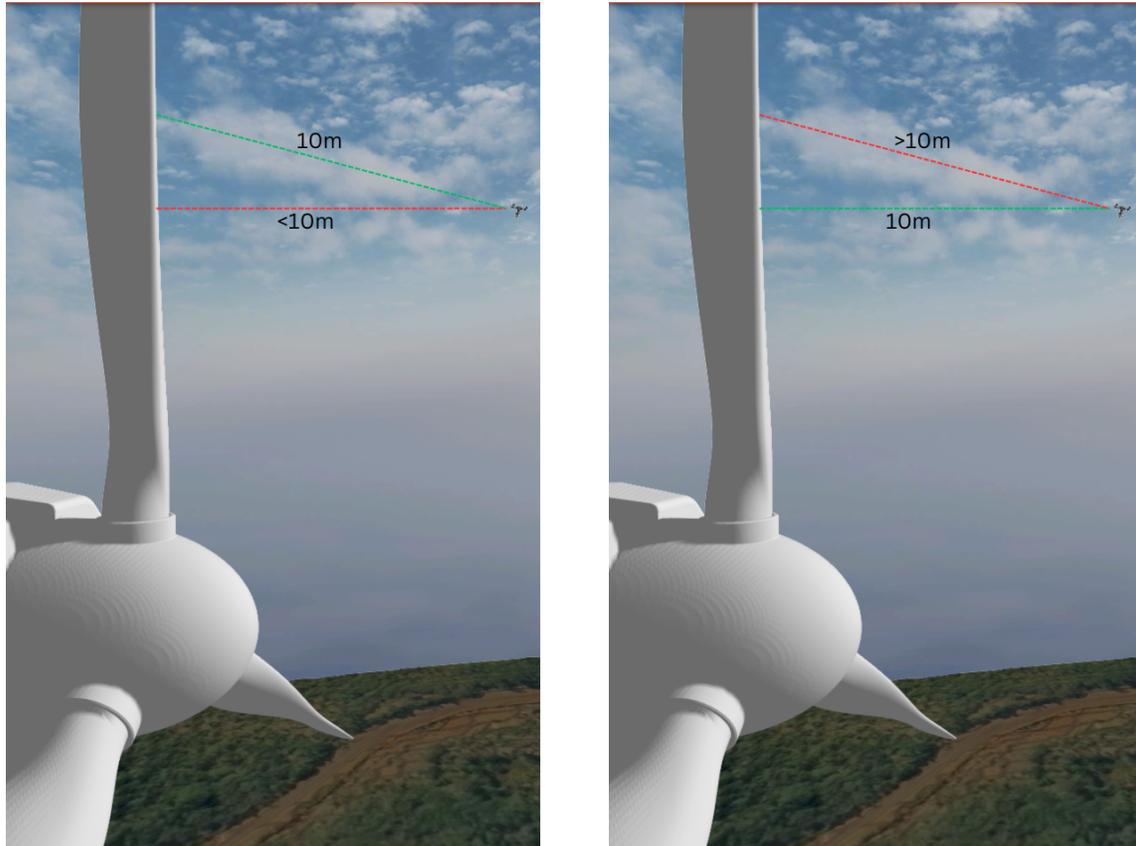
una cámara/LiDAR fija:

Distancia al aerogenerador:

Uno de los requisitos necesarios para realizar la inspección es mantener una distancia óptima del aerogenerador, pero surgió la duda de cómo debía medirse esa distancia: ¿respecto de las posiciones GPS, por las medidas recabadas por el LiDAR o de otra manera?

Debido a la inclinación de la cámara/LiDAR por el viento, puede que haya discrepancia entre ambas distancias, y se debe decidir en cuál basarse. Si la distancia se midiera desde el LiDAR, sería necesario ajustar constantemente la posición del dron en dirección norte-sur (sin modificar el eje este-oeste) para los casos de viento de frente o de atrás, compensando el efecto del viento y los cambios en la inclinación del dron.

En los ejemplos ilustrados en 3.22, se puede observar que en escenarios con viento, el cumplimiento de la distancia deseada en una de las componentes provoca que la otra no la alcance.



(a) La distancia al aerogenerador en la diagonal, que sería la medida por el LiDAR, es 10m.

(b) La distancia horizontal al aerogenerador, que sería la medida según el GPS, es de 10 metros.

Figura 3.22: Impacto del viento en la inclinación de la cámara cuando el viento sopla desde detrás del dron.

Corrección de la Rotación de la Imagen:

Cuando el viento sopla de forma lateral, sería necesario aplicar constantemente correcciones a la rotación de la imagen, sincronizándola con el ángulo de inclinación del dron, para poder realizar los cálculos adecuados durante la inspección.

Punto de interés:

Si la cámara no está posicionada perpendicularmente al aerogenerador, es posible que el punto de interés no se capture adecuadamente. Por ejemplo, el rotor del aerogenerador podría bloquear

partes del aspa que están cerca del centro del rotor, impidiendo una inspección precisa. Este caso se podría dar en un entorno similar al de Figura 3.22 en caso de que el dron tenga que inspeccionar el principio del aspa vertical y el rotor bloquearía esa primera parte.

Impacto del Movimiento del Dron:

Los problemas anteriormente descritos implican que el dron debe recalcular su ruta durante la inspección, lo que genera ineficiencias al tener que ajustar su trayectoria continuamente.

Gimbal

Solucionar estos inconvenientes es crucial para la inspección ya que la presencia de viento en parques eólicos es muy frecuente. La solución que se consideró fue usar un gimbal para contrarrestar los efectos causados por el viento y obtener una imagen clara, sin importar la orientación del dron.

Una de las dificultades fue encontrar un modelo de dron con gimbal compatible en la simulación, ya que los modelos disponibles de PX4 son para versiones antiguas del simulador y ya no son útiles.

Siguiendo con la búsqueda se encontró en el repositorio de modelos compatibles con PX4 para el simulador un modelo del mismo dron que utilizábamos (x500_depth), pero con un gimbal incorporado (x500_gimbal). Este nuevo modelo incluye una cámara, aunque carece de LiDAR, el cual se pudo añadir sin mayores dificultades utilizando como referencia el archivo de configuración del modelo x500_depth. Sin embargo, un desafío importante fue que el modelo x500_gimbal no cuenta con funcionalidad para mover el gimbal.

Posteriormente, se descubrió un Pull Request (PR) abierto en el repositorio de PX4 y realizado por el mismo desarrollador que creó el modelo x500_gimbal que agregaba la funcionalidad para poder controlar el gimbal. Agregando los cambios del PR y algunos ajustes más, se logró tener un modelo de dron con gimbal y que permita orientar la cámara en la orientación necesaria en cada momento.

Lamentablemente, al día de la fecha el PR no está terminado por lo que hay características que ofrece el protocolo que no están implementadas, como por ejemplo poder utilizar GIMBAL_DEVICE_FLAGS o GIMBAL_MANAGER_FLAGS que permiten que el manager se encargue de estabilizar la imagen de manera simple. Se decidió realizar los cálculos para la estabilidad por cuenta propia, sin que esto implique un deterioro de la solución.

Se utilizó el protocolo MAVLink (ver Anexo 5.4.4) con el método `set angles`, que permite asignar al gimbal el ángulo que debe tener en relación al dron.

Nuestra solución consiste en ajustar los ángulos del gimbal en sentido opuesto a los valores de alabeo e inclinación que publica el dron en cada actualización, lo que mantiene la cámara estabilizada. En cuanto a la orientación, se configuró para que siempre apunte hacia el frente, lo cual resolvió el problema de alineación.

El dron transmite su estado a una frecuencia de 60 Hz, lo que permite realizar correcciones constantes al gimbal para asegurar su estabilidad en tiempo real.

Este proceso que estabiliza la cámara se ejecuta de manera continua durante toda la operación y se inicializa al comienzo del vuelo.

Gracias a este enfoque, el dron logra mantener una posición relativa estable frente al aerogenerador, independientemente de las condiciones de viento. Esto facilita la generación de puntos de inspección en el algoritmo y mejora la estabilidad y calidad de las imágenes, asegurando que las inspecciones sean consistentes, sin importar las condiciones climáticas.

3.5. Máquina de Estados del Sistema de Control del Dron

Las soluciones mencionadas, por sí solas, no tienen un impacto significativo; sin embargo, al integrarse en la máquina de estados, adquieren un valor relevante en el contexto de la inspección.

Se optó por implementar un nodo de ROS 2 que funciona como una máquina de estados. En este sistema, cada estado recibe entradas y, en función de cálculos específicos, toma decisiones que proporcionan instrucciones al dron sobre los movimientos a realizar. Esta estructura es esencial, ya que en cada etapa de la inspección es necesario interactuar con diversos componentes de distintas maneras específicas para cada una. Por ejemplo, la información proporcionada por la imagen capturada por la cámara y LiDAR no se utiliza de la misma forma al intentar posicionar el dron ortogonalmente respecto al aerogenerador que cuando se está inspeccionando la superficie del aspa.

Cada estado se deriva de un estado base previamente definido, que contiene toda la información necesaria sobre el aerogenerador a inspeccionar. Este estado base incluye una suscripción al tópicos en el que el dron notifica su llegada a un setpoint, así como el callback correspondiente, y un atributo el cual la máquina de estados esperará a que se marque como `True` para poder avanzar al siguiente estado.

Se define una tabla que contiene los datos de los aerogeneradores de viento, incluyendo su posición GPS, así como las dimensiones de sus componentes (base, aspas y rotor). Esta información permitirá configurar el sistema para seleccionar un aerogenerador específico en función de su identificador (id) antes de realizar la inspección. En función del aerogenerador seleccionado, cada estado del programa podrá calcular los valores relevantes utilizando los parámetros correspondientes para ejecutar las tareas de inspección.

3.5.1. Transición entre Estados

La inspección de las cuatro superficies de cada aspa del aerogenerador se realizará mediante dos ejecuciones de la máquina de estados, donde se inspeccionarán dos superficies por aspa en cada ejecución. Esta estrategia se fundamenta en dos factores principales.

En primer lugar, como se indicó en la Sección 2.2, el giro de las aspas de un aerogenerador generalmente está limitado a un rango de 90 grados. Debido a esta restricción, es necesario rotar el aerogenerador sobre su eje longitudinal (giro de orientación) para permitir la inspección completa de todas las superficies de las aspas.

El segundo factor es la autonomía limitada de los drones, cuya capacidad de batería generalmente no permite completar la inspección en una sola operación continua. Por lo tanto, resulta imprescindible que el dron regrese a su punto de despegue en algún momento para realizar el cambio de batería.

La división de la inspección en dos ejecuciones de la máquina de estados permite aprovechar este intervalo para cambiar la batería del dron y rotar el aerogenerador sobre su eje vertical, garantizando así una inspección completa y eficiente de las aspas.

En cada pasada, se inspeccionará la superficie del aspa que esté frente al dron. Entre una pasada y la siguiente, se deberán rotar las aspas para exponer una nueva superficie aún no evaluada. Una de las ejecuciones de la máquina de estados se realizará con el aerogenerador de frente al dron, y la otra, con el aerogenerador de espaldas. Esta diferenciación es crucial, ya que el dron deberá adoptar diferentes comportamientos en cada caso. Se identifican dos diferencias principales entre ambas situaciones.

La primera diferencia radica en la distancia de seguridad que el dron debe mantener respecto a

las aspas del aerogenerador. Durante la inspección frontal, el dron se posicionará a una distancia de 10 metros, mientras que, en la inspección trasera, se mantendrá a 20 metros. Esta variación se debe a que, en la inspección trasera, la góndola y la torre del aerogenerador están más cerca del dron que las aspas, lo que requiere una mayor distancia para evitar colisiones. Se evaluó la posibilidad de utilizar el zoom de la cámara para capturar imágenes más detalladas, dado que la distancia a las aspas es mayor. Aunque, tras realizar pruebas, no se consideró necesario implementar esta opción, sigue siendo una alternativa a considerar en caso de que se requiera en el futuro.

Finalmente, el procesamiento y filtrado de la información obtenida a través del LiDAR también se ajusta según la orientación del aerogenerador. Cuando el aerogenerador está de frente, se filtran las imágenes del LiDAR en función de la proximidad de las aspas. En cambio, cuando se realiza la inspección trasera, las aspas se encuentran más alejadas del dron, y el LiDAR detecta primero la góndola y la torre, las cuales deben ser ignoradas en los cálculos.

Por estas razones, al ejecutar el programa, se debe establecer un indicador booleano que defina si la inspección es frontal o trasera, junto con los parámetros específicos de la misión que describan las dimensiones del aerogenerador a inspeccionar.

En la Figura 3.23 se observa un diagrama simplificado de la máquina de estados y las transiciones entre los estados.

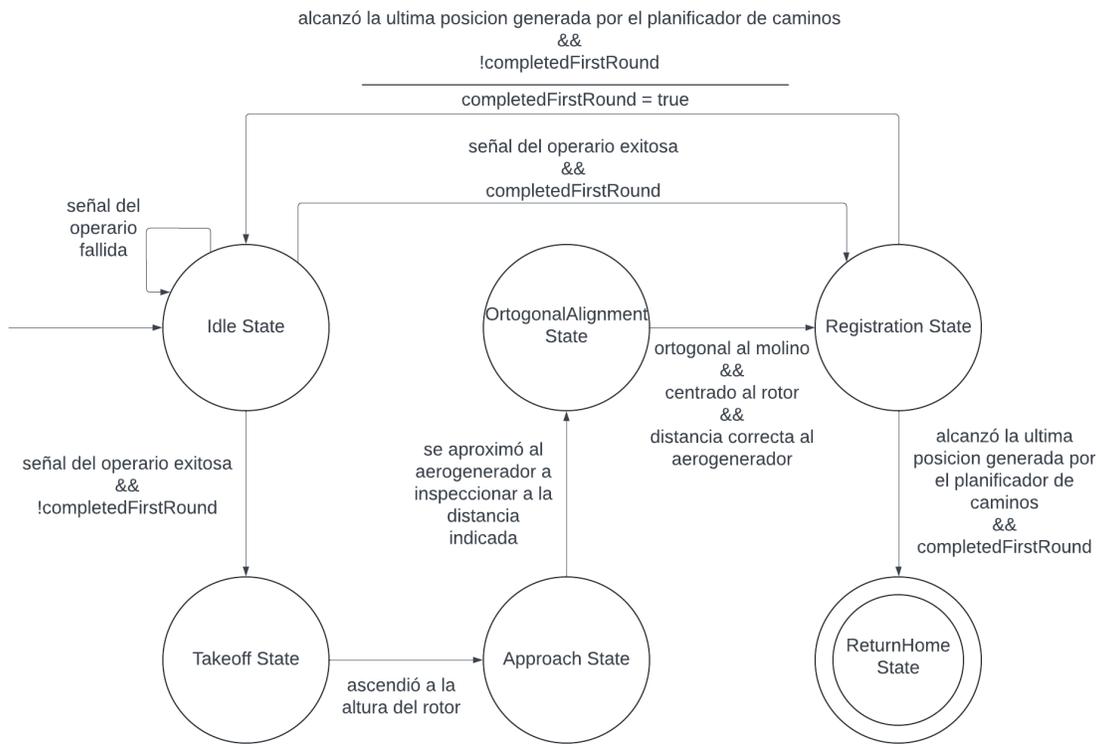


Figura 3.23: Diagrama simplificado de la máquina de estados utilizada para llevar a cabo la inspección

Los estados definidos en la máquina de estados son los siguientes: el estado `IdleState` representa la fase de espera antes del despegue y durante la rotación de las aspas del aerogenerador. El estado `TakeoffState` corresponde a la etapa de despegue del dron. Posteriormente, el estado `ApproachState` abarca la fase de aproximación al aerogenerador. En la etapa de posicionamiento ortogonal al aerogenerador se encuentra el estado `OrtogonalAlignmentState`. Durante la inspección de las aspas se activa el estado `RegistrationState`, y finalmente, el estado `ReturnHomeState` corresponde a la etapa de retorno del dron a tierra.

El estado inicial es el `IdleState`, que contiene el parámetro de misión con las medidas del aerogenerador a inspeccionar, la variable que indica si la inspección es frontal o trasera, y el booleano `completedFirstRound` configurado inicialmente en `false`. Esta variable cambiará a `true` en un punto específico de la inspección para gestionar la transición entre estados. Dado que la descripción detallada de este y otros estados se presentará en las subsecciones siguientes, a continuación, se abordarán únicamente las transiciones.

En el estado `IdleState`, mientras las señales enviadas por el operador generen un error, se permanecerá en este estado. Cuando se reciba una señal exitosa, y considerando que `completedFirstRound` es inicialmente `false`, se procederá al `TakeoffState`.

Desde `TakeoffState`, la transición al `ApproachState` ocurrirá una vez que el dron alcance la altura especificada del aerogenerador, y luego se avanzará al `OrtogonalAlignmentState` al aproximarse a la distancia adecuada, que, como ya fue mencionado, varía según la orientación de la inspección.

En el `OrtogonalAlignmentState`, la transición al `RegistrationState` se realizará cuando el dron se alinee ortogonalmente con respecto al plano formado por las aspas del aerogenerador. Durante la primera ejecución de este estado, dado a que `completedFirstRound` es `false`, al alcanzar el último setpoint proporcionado por el planificador de caminos, se actualizará esta variable a `true` y la máquina de estados retornará al `IdleState`.

Una vez de vuelta en el `IdleState`, se esperará otra señal exitosa del operador que indique que se terminó de girar las aspas sobre su eje longitudinal (giro de pala). Ahora, con `completedFirstRound` en `true`, se avanzará directamente al `RegistrationState`, donde el planificador de caminos se ejecutará nuevamente para recorrer las posiciones calculadas.

Finalmente, al alcanzar la última posición generada en la segunda ejecución del planificador de caminos, la máquina de estados pasará al `ReturnHomeState`, donde se llevarán a cabo las tareas necesarias para que el dron regrese a su punto de despegue, concluyendo así la ejecución del programa.

El proceso descrito se realizará dos veces, una para la inspección trasera del aerogenerador y otra para la delantera.

3.5.2. Estado `IdleState`

El estado `IdleState` desempeña dos funciones principales:

1. Establecer las condiciones iniciales y preparar al dron antes de iniciar la misión. Se leen los parámetros de entrada ingresados por el operador (identificador del aerogenerador a inspeccionar y si se trata de una inspección trasera o delantera) y se almacenan para su uso en estados posteriores.
2. Mantener al dron en espera hasta que un operador lo reactive manualmente mediante un servicio, lo que permitirá avanzar al siguiente estado y continuar con las tareas programadas.

El estado `IdleState` se visita en dos momentos distintos durante el ciclo de la máquina de estados. En primer lugar, se presenta como el estado inicial al comenzar la inspección. Durante esta fase, se configuran las condiciones necesarias para la misión y se espera la señal del operador para proceder. Es importante destacar que, aunque el operador envíe la señal para continuar, el dron puede no avanzar y generar un mensaje de error. Esto ocurrirá hasta que se confirme que todos los sistemas y sensores están completamente inicializados y listos para operar, ya que hasta ese entonces se ignorarán las llamadas al servicio. En este caso, se requerirá que el operador espere unos segundos hasta que la inicialización se complete y vuelva a invocar el servicio.

La segunda visita a este estado tiene lugar al finalizar la inspección de los tres lados frontales de las aspas. En esta etapa, el dron permanece en espera mientras el operador rota las aspas para inspeccionar otras superficies. Una vez que el operador ha completado la rotación y notifica a través del servicio, el dron avanza automáticamente al siguiente estado.

En este estado del sistema se realiza una serie de chequeos previos para notificar al operador sobre posibles fallas antes de avanzar. Estos chequeos se centran en tres áreas principales:

- **Posicionamiento:** Comprueba la validez y precisión de la posición y orientación del sistema, incluyendo velocidad angular, actitud, y posiciones local y global, además de la posición de origen.
- **Baterías:** Verifica el estado de las baterías, detectando posibles problemas de carga insuficiente o condiciones no saludables.
- **Estado de los motores:** Evalúa fallos de propulsión, desequilibrios y fallas de los motores.

Estos chequeos permiten al operador identificar y resolver problemas antes de pasar al siguiente estado del sistema.

3.5.3. Estado `TakeoffState`

En el estado `TakeoffState`, se gestiona el despegue inicial del dron para dar inicio a la inspección del aerogenerador. Durante esta etapa, se llevan a cabo las siguientes acciones:

1. Se obtiene la altura del rotor utilizando el parámetro de misión almacenado, proporcionado previamente por el operador. Esto permite que el dron ajuste su configuración de forma precisa según los requisitos específicos de cada misión.
2. Se define un callback que se activa cuando el dron alcanza un setpoint. Al recibir esta confirmación, el sistema avanza automáticamente al siguiente estado de la misión. Esta transición fluida entre estados es fundamental, ya que garantiza que el dron esté listo para proceder con las siguientes etapas.

De esta forma, se asegura que el dron ascienda correctamente hasta la altura del rotor especificada y que la máquina de estados avance de manera controlada hacia el siguiente estado.

3.5.4. Estado `ApproachState`

El estado `ApproachState` tiene la responsabilidad de acercar al dron al aerogenerador, posicionándolo adecuadamente antes de iniciar la inspección de las aspas. Esta etapa es crucial para garantizar que el dron esté en la ubicación correcta para realizar una evaluación eficiente.

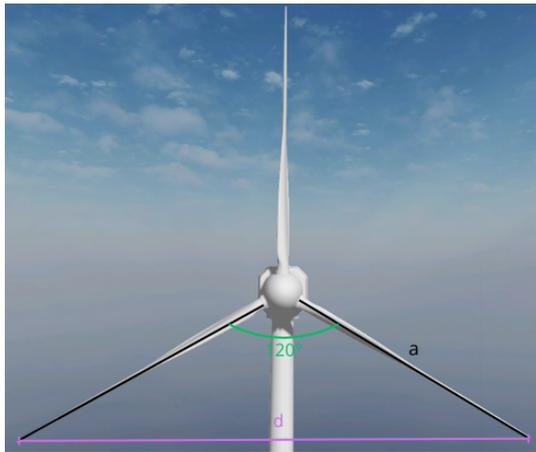
Al ingresar a este estado, de forma análoga al estado anterior, se accede a las medidas del aerogenerador proporcionadas por el operador para obtener la coordenada de la base del aerogenerador a inspeccionar. En este caso, no es necesario que el dron alcance la latitud y longitud exactas de la base; estas sirven únicamente como referencia. La posición en la que se quiere situar al dron se encuentra a una distancia específica antes de la base, y para determinarla se consideran dos elementos clave. Primero, se tiene la distancia horizontal desde la base del aerogenerador hasta la punta del rotor en las medidas del aerogenerador.

Segundo, conociendo la longitud de las aspas y que el ángulo entre aspas es 120° se puede calcular la distancia d observada en la Figura 3.24a utilizando la Ecuación 3.1 que se basa en la relación del seno.

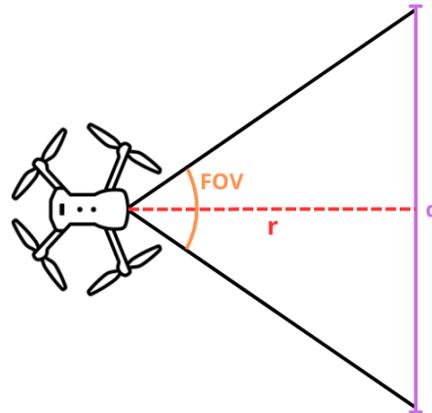
$$d = 2 \cdot \sin\left(\frac{120^\circ}{2}\right) \cdot a \quad (3.1)$$

Después, utilizando la distancia d calculada y el campo de visión (FOV) de la cámara, se puede obtener la distancia r a la que se debe mantener el dron para cubrir la distancia d . Esta distancia r se visualiza en la Figura 3.24b y se calcula según la Ecuación 3.2 fundamentada en la función tangente.

$$r = \frac{d}{2 \cdot \tan\left(\frac{\text{FOV}}{2}\right)} \quad (3.2)$$



(a) Vista de frente de un aerogenerador.



(b) Vista en desde arriba de un dron y su Field Of View (FOV).

Figura 3.24: Imágenes ilustrativas para el cálculo de la distancia deseada del dron respecto al aerogenerador.

A esta distancia r se pensó en agregarle un margen adicional para asegurar que el aerogenerador se capture correctamente en la imagen, incluso en situaciones en las que no esté directamente frente al dron. Sin embargo, después de realizar pruebas, se observó que iniciar la siguiente etapa más alejado del aerogenerador resultaba en un aumento en el tiempo de ejecución de la misma. Se

concluyó que no es necesario cubrir toda la distancia d en la imagen para obtener buenos resultados y que, al acercarse un poco más, el proceso se completa más rápidamente sin comprometer la calidad de los resultados. Por ello, se decidió ajustar la distancia r , reduciéndola a $2/3$ de su valor original, basado en la experimentación.

Luego, esta distancia ajustada r se suma a la distancia horizontal desde la base del aerogenerador hasta la punta del rotor, dando como resultado la distancia deseada entre el dron y el aerogenerador. Finalmente, se publica un mensaje a un tópicos específico en el controlador que incluye la latitud y longitud de la base del aerogenerador, junto con la distancia calculada a la que se desea mantener el dron de la misma. Si el mensaje es recibido correctamente, el controlador procesa la coordenada y distancia recibidas del tópicos y calcula las coordenadas a las que se dirigirá el dron (waypoint).

Además, de igual forma al estado anterior, se define un callback que se activa cuando el dron alcanza el waypoint. Al recibir esta confirmación, se registra un mensaje informativo y el dron avanza automáticamente al siguiente estado de la misión.

3.5.5. Estado `OrthogonalAlignmentState`

El estado `OrthogonalAlignmentState` tiene como objetivo alinear al dron ortogonalmente respecto al aerogenerador y centrar el rotor en la imagen, preparando así las condiciones óptimas para una inspección satisfactoria. Este estado es crucial, ya que una alineación correcta minimiza el riesgo de colisiones o desviaciones durante la misión.

En este estado, el dron realiza una serie de ajustes en su posición y orientación a través de etapas secuenciales donde se chequea el estado.

El nodo que analiza el stream de la cámara, llamado `image_subscriber`, emite cuatro mediciones que son fundamentales en esta etapa de la inspección:

- diferencia entre la distancia al aspa izquierda y la distancia al aspa derecha con un signo que indica sentido de rotación (para más detalles ver Sección 3.4.1)
- distancia al rotor
- grados que tiene que rotar sobre si mismo el dron para que el rotor quede en el centro de la imagen horizontalmente
- porcentaje de la altura de la imagen en la que esta el rotor, es un numero entre 0 y 1 que teniendo el FOV de la cámara, el aspect ratio y la distancia al dron permite calcular cuantos metros tiene que subir o bajar el dron para que el rotor quede centrado verticalmente

El procedimiento fue inspirado por el método realizado por Stokkeland et al. 2015, se utiliza una estructura cíclica de decisiones para verificar y ajustar condiciones específicas de alineación en distintos ejes (vertical, horizontal y orientación) hasta alcanzar la configuración deseada. El Algoritmo 1 muestra un pseudocódigo de la lógica en `OrthogonalAlignmentState`.

Cada iteración del algoritmo anterior se corre cuando se reciben los datos mencionados anteriormente por parte del paquete `image_subscriber`.

Decimos que la alineación es estable cuando no se requirió ninguna corrección en las últimas 20 iteraciones del bucle. Es decir, que los últimos 20 datos recibidos del `image_subscriber` indican que la alineación es correcta. El valor 20 se obtuvo de forma experimental. En la Figura 3.25 se observa en RVIZ el recorrido realizado por el dron desde que despega hasta que se coloca ortogonalmente.

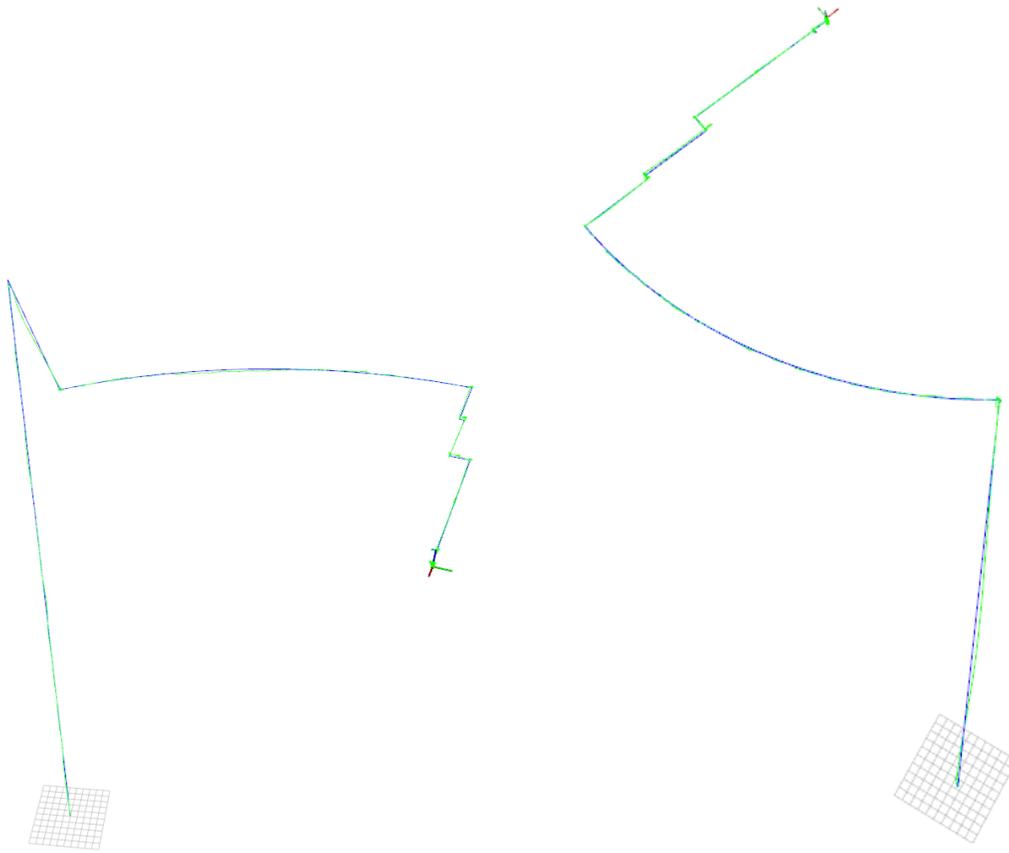
Algorithm 1 Algoritmo Simplificado de Alineación Ortogonal (Modificado)

```
1: Cambiar al modo de procesamiento de imágenes para alineación (el modo alternativo es para
   correcciones durante la inspección).
2: procedure AL RECIBIR NUEVA IMAGEN(datos.imagen)
3:   1. Obtener mediciones: ángulo, posición vertical y desviación.
4:   2. Verificar alineación vertical:
5:   if el rotor no está centrado verticalmente then
6:     Enviar comando para ajustar la altura.
7:     return
8:   end if
9:   3. Verificar alineación horizontal:
10:  if el rotor no está centrado horizontalmente then
11:    Enviar comando para rotar sobre su propio eje de guiñada.
12:    return
13:  end if
14:  4. Verificar ortogonalidad:
15:  if el dron no está ortogonal al aerogenerador then
16:    Enviar comando para ajustar orientación y posición.
17:    return
18:  end if
19:  5. Verificar estabilidad de la alineación:
20:  if la alineación actual es estable then
21:    if se ha alcanzado la distancia deseada then
22:      Avanzar al siguiente estado.
23:      return
24:    else
25:      Enviar comando para acercarse al aerogenerador.
26:      return
27:    end if
28:  end if
29: end procedure
```

Para obtener mediciones más estables y confiables del nodo de inspección de la imagen, decidimos utilizar un enfoque basado en un arreglo de las últimas 50 mediciones. En lugar de basarnos en una única medición, calculamos la mediana de este conjunto de 50 valores y usamos ese valor central para compararlo con umbrales establecidos. Este método nos permite reducir el impacto de posibles valores atípicos o ruido en los datos.

Los umbrales que definimos sirven como márgenes de tolerancia para las diferentes correcciones que se aplican al rotor y al aerogenerador en general. En lugar de buscar ajustes perfectos, los umbrales permiten que el sistema funcione de manera eficiente sin quedar atrapado en pequeños desajustes que podrían causar oscilaciones o sobrecorrecciones innecesarias; por consecuencia, agilizando el proceso.

Primero, para mantener el rotor centrado verticalmente, se definió un umbral de $\pm 0,1$ en torno al valor medio, que es 0.5. Este margen garantiza que el rotor permanezca aproximadamente centrado sin intentar alcanzar una precisión excesiva.



(a) Trayectoria que realiza el dron para ponerse ortogonal vista en diagonal

(b) Trayectoria que realiza el dron para ponerse ortogonal vista desde arriba

Figura 3.25: Ejemplo de trayectoria que recorre el dron desde que despega hasta ponerse ortogonal al aerogenerador.

En cuanto a los grados para la rotación sobre sí mismo, se estableció un umbral de ± 3 grados.

Por último, cuando se requiere rotar el dron con centro en el aerogenerador, utilizamos un umbral de $\pm 0,5$.

Finalmente, una vez que se van validando las diferentes etapas, el dron se va acercando hasta llegar a la distancia de seguridad respecto al aerogenerador, donde en este momento se espera hasta que todas las validaciones sean correctas y se avanza al siguiente estado.

3.5.6. Estado RegistrationState

El estado `RegistrationState` se encarga de llevar a cabo la inspección de las aspas del aerogenerador.

Para esto, al igual que en los estados anteriores, se obtienen los parámetros de misión almacenados con los datos relevantes del aerogenerador para este estado: el diámetro del rotor y la longitud de las aspas. A partir de estas medidas y de la posición actual del dron, se ejecuta el planificador de caminos.

Como se mencionó anteriormente, el planificador de caminos genera una ruta óptima para recorrer las tres superficies frontales de las aspas. La ruta generada proporciona para cada punto su posición y un identificador del aspa correspondiente, de manera que todos los puntos asociados con la misma aspa comparten el mismo identificador. La posición de cada punto se utiliza para indicar al dron hacia dónde debe dirigirse, mientras que el identificador permite determinar si el dron está recorriendo un aspa o si está en transición entre aspas. Si el identificador del siguiente punto es igual al del punto actual, significa que el dron continúa recorriendo la misma aspa; si es diferente, indica que está cambiando de una aspa a otra. Esta información es crucial por tres motivos.

El primero de estos motivos es poder regular la velocidad del dron. Al inspeccionar un aspa, es importante que el dron se desplace a una velocidad reducida, para ayudar a que las imágenes capturadas sean de alta calidad. Sin embargo, cuando se traslada de una aspa a otra, se puede aumentar la velocidad, ya que las imágenes no son necesarias en ese tramo, reduciendo así el tiempo total de inspección.

La segunda razón es la captura selectiva de imágenes. Con esta información, se asegura que las imágenes se capturen únicamente durante el recorrido de las aspas, evitando registros innecesarios mientras el dron se desplaza entre ellas.

En tercer lugar, esta información permite realizar ajustes en la posición del dron para mantener su alineación correcta con las aspas durante la inspección. Aunque el planificador de caminos proporciona las posiciones hacia las que debe moverse el dron, pueden surgir desviaciones debido a factores externos. Estos ajustes son necesarios únicamente cuando se está inspeccionando una aspa, ya que durante los desplazamientos entre aspas no se observa nada de interés.

Para controlar la velocidad y realizar ajustes durante la inspección de un aspa, se añaden puntos intermedios (setpoints) entre las posiciones definidas previamente. Estos setpoints permiten mantener una velocidad constante y reducida durante el recorrido, además de corregir desviaciones utilizando la información obtenida de las imágenes capturadas por el dron utilizando el LiDAR. Ver Subsección 3.4.2 para más detalles de cómo se utiliza el LiDAR para corregir la trayectoria.

En este estado, el sistema se suscribe a un tópico específico mediante el cual el nodo encargado de procesar los datos de la cámara del dron envía de forma continua información sobre la posición del centroide calculado, que actúa como un punto de referencia para el aspa. Esta posición se proporciona en porcentajes horizontales y verticales dentro de la imagen, además de incluir la distancia al punto de referencia.

Se recopilan varias mediciones para calcular valores medianos estables y confiables de la posición relativa del dron respecto al aspa. Si se detectan desviaciones significativas en la alineación horizontal, vertical o en la distancia al aerogenerador, se calcula un vector de corrección para ajustar la posición del dron. Este vector de corrección se publica en otro tópico, y el controlador recibe esta información para realizar los ajustes necesarios, manteniendo al dron dentro de los parámetros establecidos. Este proceso se repite hasta que se avance al siguiente estado de la misión.

En lo que respecta a la posición vertical, se busca que el punto de interés se encuentre entre el 30 % y el 70 % de la imagen. Si se encuentra fuera de este rango, se agrega un factor de corrección que mueve el dron 2.5 metros en la dirección opuesta a la ubicación del punto en la imagen. Es decir, si el punto está en el intervalo inferior, el dron debe desplazarse hacia abajo.

De manera análoga, se aplica un ajuste a la posición horizontal. En este caso, el intervalo

establecido es del 10% al 90% de la imagen, y la corrección desplaza al dron 4.0 metros en la dirección opuesta.

En cuanto a la distancia al aerogenerador, se busca que esta sea lo más cercana posible a lo observado durante la inspección profesional (ver Sección 2.2). Para ello, se definen valores mínimos y máximos aceptables: para la inspección frontal entre 8 y 11 metros, para la inspección trasera entre 19 y 22 metros. En este caso, si en algún momento se detecta que el dron está fuera del rango establecido, la corrección calculada hará que el dron se desplace hacia el punto medio entre el valor mínimo y el máximo. Esto evita que, si el dron se acerca o aleja demasiado, tenga que corregir constantemente su posición. Al apuntar al punto medio, se logra que el dron mantenga una distancia óptima sin necesidad de ajustes continuos, lo que mejora la estabilidad y eficiencia de la inspección.

De igual manera a los estados anteriores, en este estado se define un callback que se activa al alcanzar un setpoint. Cuando se confirma que se ha alcanzado el último setpoint generado por el planificador de caminos, se procede a avanzar automáticamente al siguiente estado.

3.5.7. Estado `ReturnHomeState`

El estado `ReturnHomeState` se encarga de gestionar el retorno del vehículo a su punto de partida tras completar la inspección. Al ingresar en este estado, se publica un mensaje en un tópico específico para indicar que debe iniciarse el trayecto de regreso del dron.

El controlador del dron, que está suscrito a este tópico, responde limpiando las pilas de setpoints para asegurarse de que no queden instrucciones pendientes y desactiva la posibilidad de añadir nuevos setpoints. A continuación, se envía un comando de navegación que instruye al dron a regresar a su ubicación de lanzamiento.

3.5.8. Máquina de estados con una única ejecución

Se evaluó la posibilidad de que el dron realizara la inspección de las cuatro superficies de las aspas en una sola ejecución de la máquina de estados, sin que el dron regresara a su punto de despegue hasta completar la inspección completa.

El flujo de la máquina de estados comenzaba de manera similar a como se realiza actualmente, con la diferencia de que, al salir del `RegistrationState`, se transitaba hacia un estado distinto: `RotationState`, y de este de vuelta al `RegistrationState`. Este ciclo se repetía hasta que, al salir del `RegistrationState`, se determinaba que todas las superficies del aerogenerador habían sido inspeccionadas, lo que implicaba que se había pasado cuatro veces por este estado. En ese momento, se procedía al estado `ReturnHomeState`, finalizando la inspección.

El `RotationState` se ejecutaba tres veces durante la inspección. En la primera y tercera iteración, el comportamiento era similar al que se utiliza actualmente en el `IdleState`. En estas instancias, el dron simplemente esperaba a que el operador enviara una señal a través de un servicio específico para reanudar la inspección. La idea era que, durante este estado, el operador pudiera rotar las aspas del aerogenerador sobre su eje longitudinal para permitir la inspección de otras superficies.

Sin embargo, el comportamiento del segundo paso por el `RotationState` era diferente. Al entrar en este estado por segunda vez, se enviaba un mensaje al controlador del dron para alejarlo del aerogenerador una distancia equivalente a la longitud de las aspas. Una vez alcanzada esta distancia de seguridad, el dron esperaba la señal del operador para reanudar la inspección. Al recibir

dicha señal, se enviaba otro mensaje al controlador para que el dron se acercara nuevamente al aerogenerador, regresando a la posición inicial. Al completar este proceso, se procedía al estado `RegistrationState`. El propósito de este alejamiento era permitir que el dron permaneciera en una posición segura mientras el operador rotaba el aerogenerador sobre su eje vertical, de modo que se pudieran inspeccionar las superficies restantes.

Finalmente, este enfoque fue descartado cuando se consideró la autonomía del dron y se evaluó el tiempo promedio necesario para completar una inspección completa. Se determinó que, debido a las limitaciones de la batería, era inevitable que el dron tuviera que regresar a la base para un cambio de la misma. Como resultado, se simplificó la máquina de estados, aprovechando el momento del cambio de batería para sincronizarlo con la rotación del aerogenerador, optimizando así el proceso de inspección.

3.6. Adquisición de imágenes

Los movimientos realizados durante la inspección por sí solos no son suficientes si no se puede analizar el resultado del recorrido.

Para la adquisición de imágenes, se decidió tomar una fotografía cada cierta distancia recorrida, de manera que las imágenes puedan ser analizadas posteriormente. Es de suma importancia que todas las partes del segmento del aspa a inspeccionar sean fotografiadas para poder ser correctamente analizadas.

Para este fin, se establece una distancia específica: cada vez que el dron se desplace dicha distancia y esté en la fase de inspección, tomará una fotografía. Utilizando el campo de visión (*FOV*) de la cámara, la distancia mínima al aerogenerador r y la relación de aspecto de las imágenes, se puede obtener una estimación de la distancia horizontal d que ocupa cada imagen. Esta distancia d se visualiza en la Figura 3.24b y se calcula según la Ecuación 3.3 fundamentada en la función tangente.

$$d = 2r \cdot \tan\left(\frac{\text{FOV}}{2}\right) \quad (3.3)$$

Conceptualmente, se podría capturar una imagen por cada una de estas distancias. Sin embargo, esto no sería adecuado, ya que las tomas quedarían demasiado ajustadas, y esto bajo la suposición de que las aspas estuvieran completamente horizontales, lo cual no ocurre en la práctica ya que el aerogenerador asume la posición de Y invertida. Para aspas en otras orientaciones, como diagonal o vertical, esta distancia resultaría insuficiente para cubrir adecuadamente la inspección. Por lo tanto, se optó por utilizar una relación de aspecto de 4:3 y reducir en un 60% la distancia entre tomas fotográficas. Se determinó capturar una imagen en cada intervalo ajustado, generando un solapamiento entre imágenes que asegura la completitud de la inspección.

Para poder realizar la adquisición de imágenes, se utiliza lo siguiente:

- **Tópico de la cámara:** se utiliza para obtener el *frame* a fotografiar.
- **Posición del dron:** cada vez que llega un mensaje, se calcula la distancia respecto al *setpoint* de la última foto tomada; si esta distancia es mayor al umbral previamente mencionado y, además, se encuentra en un estado donde debe tomar la foto, entonces llama a la función de capturar la foto que además guarda la posición de la última foto.

- **Mensajes de llegada del dron a *setpoints*:** se utilizan para determinar en qué parte del recorrido se encuentra el dron. Principalmente interesan los mensajes de inicio y finalización de la inspección de un aspa.

Un problema que se presentó fue que podía ocurrir que la punta de un aspa no quedara fotografiada si la última foto no la cubría. Tomar la foto en el momento en que llega el mensaje de aspa completada tampoco lo soluciona, porque es posible que el dron ya esté dirigiéndose a otro punto. Para solucionarlo, el dron emite un mensaje cuando está por completar el aspa, y el nodo, al recibirlo, toma una nueva foto sin importar la distancia a la posición de la última foto.

Capítulo 4

Experimentación

Para estudiar y comprender el desempeño del sistema, se llevaron a cabo pruebas en diferentes entornos simulados en Gazebo, un simulador de robots que permite recrear escenarios físicos realistas (ver Anexo 5.4.2), evaluando tanto casos típicos como escenarios límite.

El parque *Los Caracoles* sirvió como entorno de simulación para el proyecto. Para recrear un entorno lo más realista posible, se descargaron imágenes de relieve y satelitales desde el sitio <https://visualizador.ide.uy>, las cuales sirvieron de base para construir un modelo fiel del parque eólico.

En cuanto a los aerogeneradores, se investigó la disponibilidad de modelos 3D correspondientes a los utilizados en *Los Caracoles* para utilizar en la simulación. El modelo empleado en dicho parque es el Vestas V80/2000; sin embargo, no se encontraron modelos digitales accesibles en línea de este ni de otros aerogeneradores Vestas, ni de aquellos utilizados en otros parques eólicos de Uruguay. Como solución, se optó por utilizar un modelo genérico de aerogenerador ¹ disponible en la web para completar la simulación. Este modelo genérico tiene dimensiones adaptadas para la simulación, con las siguientes características:

- **Altura de la torre:** 79 metros.
- **Diámetro del rotor:** 5 metros.
- **Longitud de las aspas:** 39 metros.

Para las pruebas, se utiliza siempre el mismo tipo de aerogenerador en la misma posición GPS. Las únicas modificaciones al entorno son el viento, la rotación del aerogenerador (para inspeccionar la parte delantera y trasera) y la posición inicial del dron.

La luz solar se posiciona directamente frente al aerogenerador, imitando el procedimiento que realizan los operarios manualmente. Es importante destacar que nuestro sistema no depende de imágenes a color al momento de posicionar al dron de forma ortogonal ni para recorrer los puntos de inspección. La decisión de colocar la fuente de luz frente al aerogenerador se toma únicamente para que los resultados sean lo más similares posible a las inspecciones realizadas de forma manual y para optimizar la calidad de las imágenes capturadas por la cámara.

Se consideran 2 etapas diferentes del proceso de inspección a evaluar por separado:

¹<https://www.printables.com/model/302850>

- Etapa de aproximación: el dron se acerca al aerogenerador, centra el rotor y se coloca ortogonal al plano de las aspas del aerogenerador.
- Etapa de inspección: se realiza la inspección de las aspas.

Para obtener los datos relevantes en los estados de interés, se implementa un nodo dedicado a registrar todas las mediciones necesarias. Este nodo se inicializa con un estado de inicio y un estado de fin, lo que le permite registrar datos únicamente en los estados que se desee.

Se aprovechan los tópicos proporcionados por ROS 2 para escuchar las comunicaciones hacia y desde el dron. La máquina de estados envía un mensaje cada vez que ocurre un cambio de estado. Cuando se detecta el cambio al estado de inicio, se comienzan a almacenar los datos. Al recibir el mensaje correspondiente al estado de fin, los datos se guardan en un archivo JSON. Posteriormente, este archivo se procesa junto con los demás datos recopilados en otras ejecuciones bajo las mismas condiciones iniciales para calcular métricas como la mediana y la desviación estándar de diferentes atributos.

4.1. Simulaciones

4.1.1. Entornos de hardware

Debido a que las pruebas requieren un tiempo extenso, se utilizaron dos computadoras distintas. Si bien esto no es lo ideal para mantener la consistencia en los resultados, fue necesario para optimizar el tiempo y lograr una mayor cantidad de pruebas en paralelo. Ambas computadoras se configuraron para tener un entorno de software idéntico, minimizando así posibles diferencias en los resultados debidas al entorno de ejecución. Cada caso de prueba se ejecutó siempre en la misma computadora. La Tabla 4.1 muestra ambos entornos de hardware.

Computadora	Procesador	RAM	Tarjeta Gráfica
PC 1	Intel Core i7-7700K	32 GB	NVIDIA RTX 2070
PC 2	Intel Core i5-1035G1	16 GB	NVIDIA GeForce MX110

Tabla 4.1: Entornos de hardware utilizados para las pruebas

4.1.2. Entorno de software

Para realizar las pruebas, se utilizó un entorno de software específico. La Tabla 4.2 detalla el software utilizado.

Software	Versión
Sistema Operativo	Ubuntu 22.04
Simulador	Gazebo Garden
Framework de Robótica	ROS 2 Humble

Tabla 4.2: Entorno de software utilizado para las pruebas

4.2. Mediciones realizadas

En las pruebas, se miden tres aspectos principales: el tiempo de operación, la desviación de la trayectoria comparado a la planificada y el resultado obtenido. Las primeras dos son mediciones objetivas, mientras que la tercera es una medición subjetiva. El tiempo de operación es crucial para determinar la eficiencia de la inspección y garantizar que se complete sin interrupciones.

Este factor es importante por dos razones principales:

- **Batería del dron:** Un tiempo de operación elevado puede implicar que la inspección no se complete correctamente, ya que podría ser necesario interrumpirla para cambiar las baterías del dron.
- **Eficiencia operativa:** Al acortar la duración de cada inspección, se minimiza el tiempo de detención del aerogenerador, permitiendo un regreso más rápido a la operación y reduciendo el impacto económico de la inactividad. Además, reducir el tiempo de inspección permite realizar más inspecciones en menos tiempo, mejorando el mantenimiento preventivo al detectar problemas tempranamente y evitando fallos.

Para medir el error o desviación en la trayectoria, se considera el segmento que une el punto anterior de la trayectoria con el destino actual, y se mide la distancia mínima entre este segmento y el dron. El controlador del dron emite la desviación dos veces por segundo, siempre y cuando el segmento no exceda 1 metro. Esto se debe a que PX4 no garantiza que realizará una línea recta al cambiar de un setpoint a otro; por lo tanto, si el trayecto no está segmentado, no tiene sentido comparar con una línea recta, ya que no es lo esperado cuando las distancias son mayores.

El resultado subjetivo evalúa si la operación fue satisfactoria, es decir, si el dron se posicionó correctamente para la siguiente etapa o si la inspección puede considerarse válida. En cuanto al posicionamiento ortogonal, la evaluación subjetiva implica verificar si el dron se posicionó adecuadamente para la siguiente inspección, especialmente en relación con la ortogonalidad respecto a las aspas. Para la inspección en sí, se busca detectar que las aspas del aerogenerador estén siempre en la imagen, idealmente lo más cerca posible del centro.

Además de estas mediciones principales, también hemos tomado otras medidas que consideramos secundarias. Estas incluyen parámetros como la cantidad de correcciones necesarias realizadas. Aunque estas métricas son importantes, las consideramos de menor relevancia en comparación con el tiempo de operación y el resultado obtenido, ya que no influyen de manera tan directa en la eficiencia y validez de la inspección.

4.3. Pruebas realizadas

Dado que las simulaciones en Gazebo son no deterministas —es decir, incluso bajo las mismas condiciones iniciales no producen resultados idénticos debido a factores como la física simulada y la aleatoriedad inherente al software—, se optó por realizar diez pruebas por cada escenario. Este enfoque permite obtener una base estadística mínima sobre los resultados, disminuyendo la influencia de posibles anomalías y evitando depender de una única ejecución que podría ser atípica.

Además, repetir las simulaciones múltiples veces permite calcular medidas estadísticas como la mediana y la desviación estándar. Estas métricas ofrecen una comprensión más precisa del rendimiento y la estabilidad del sistema bajo distintas condiciones, lo que resulta esencial en un entorno de simulación no determinista.

El número de diez pruebas por escenario fue seleccionado considerando el tiempo requerido para cada simulación, el cual, debido a las limitaciones del hardware utilizado, varía entre 10 y 20 minutos en las mejores condiciones. De haber realizado más pruebas por escenario, se habría tenido que reducir la cantidad de escenarios evaluados, limitando la amplitud y diversidad del estudio realizado.

Los diferentes escenarios son definidos por los siguientes factores:

1. **Posición inicial:** Para el caso de las pruebas de alineación y posicionamiento ortogonal el dron inicia en el suelo a cierta distancia, mientras que para las pruebas de inspección el dron inicia directamente en frente al rotor.
 - *En Frente:* el dron está ortogonal al plano de las aspas y centrado respecto al rotor.
 - *Izquierda:* el dron se encuentra descentrado, más hacia la izquierda del rotor.
 - *Derecha:* el dron se encuentra descentrado, más hacia la derecha del rotor.
2. **Inspección:**
 - *Delantera:* se lleva a cabo la inspección delantera del aerogenerador.
 - *Trasera:* se lleva a cabo la inspección trasera del aerogenerador.
3. **Viento**, con velocidades de 10 m/s. Estas se encuentran ilustradas en la Figura 4.1.:
 - *Viento izquierda.*
 - *Viento derecha.*
 - *Viento adelante.*
 - *Sin viento.*
4. **Distancia**, al aerogenerador. Este parámetro aplica únicamente para las pruebas de posicionamiento ortogonal del dron. Para las pruebas realizadas de las inspecciones de las aspas, la distancia no es considerada en el escenario ya que el dron opera siempre a 10 metros y 20 metros del aerogenerador cuando la inspección es delantera y trasera respectivamente:
 - *Cercana:* a 60 metros del aerogenerador.
 - *Lejana:* a 170 metros del aerogenerador.

4.3.1. Alineación y Orientación del Dron Frente al Aerogenerador

Para evaluar la capacidad del dron de posicionarse de manera ortogonal respecto a las aspas del aerogenerador y centrado al rotor, se diseñaron pruebas que involucraron acercamientos desde diferentes ángulos (siempre por el lado correspondiente a la inspección), partiendo de diversas distancias iniciales y bajo distintas condiciones de viento. Para las pruebas de alineación y orientación, para la posición inicial izquierda, el dron se ubica hacia la izquierda del aerogenerador, presentando una desviación angular de 42° . Para la posición inicial derecha, el dron se posiciona hacia la derecha del aerogenerador, con una desviación angular de 47° .

Los resultados de estas pruebas se presentan en la Tabla 4.3. En la Figura 4.2 y Figura 4.3 se muestran los mismos datos, destacando respectivamente la desviación de la trayectoria y del tiempo.

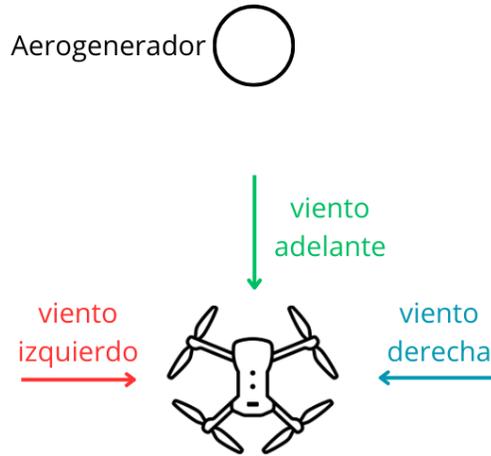


Figura 4.1: Se muestra el dron frente al aerogenerador y las posibles direcciones del viento: en rojo, el viento proveniente del lado izquierdo; en verde, el viento de frente; y en azul, el viento que llega desde la derecha.

Posición inicial	Inspección	Distancia (m)	Viento	MDT (m)	StD MDT (m)	MTS (s)	StD MTS (s)
En frente	Delantera	60	Sin viento	0.0753	0.1282	286.0	33.1845
		170	Sin viento	0.1249	0.1960	290.5	29.5312
	Trasera	170	Sin viento	0.1303	0.1879	176.0	21.7173
			Viento derecha	0.1588	0.2957	169.0	29.0002
Izquierda	Delantera	60	Viento izquierdo	0.3603	0.1866	330.0	31.4968
	Trasera	60	Viento izquierdo	0.3137	0.1826	210.5	52.4424
Derecha	Delantera	60	Sin viento	0.2965	0.1526	341.0	37.7333
			Viento adelante	0.2986	0.1455	371.0	82.1584
	Trasera	60	Viento adelante	0.2767	0.1270	197.5	21.9281

Tabla 4.3: Resultados obtenidos al evaluar la ortogonalidad en diversos escenarios. Las métricas relevadas son la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT), así como la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS). Para los escenarios que incluyen viento, se asumió siempre una velocidad constante de 10 m/s. Las pruebas fueron realizadas en la PC 2.

En todas las pruebas realizadas, el dron logró posicionarse de manera ortogonal al plano de las aspas y centrado respecto al rotor, cumpliendo con la distancia adecuada según el tipo de inspección, lo que evidencia la capacidad del sistema para operar de manera efectiva incluso en escenarios desafiantes.

Al analizar los resultados obtenidos en las pruebas, se observa un impacto significativo de las condiciones iniciales y del viento sobre el desempeño del dron en su alineación ortogonal frente al aerogenerador. En términos generales, el dron logró posicionarse adecuadamente en todos los casos, aunque las condiciones de viento y la posición inicial influyeron en la precisión y el tiempo requerido para la alineación.

Una clara diferencia se presenta al comparar las inspecciones delanteras con las traseras. Las inspecciones traseras muestran tiempos de alineación ortogonal menores, ya que el dron termina de alinearse a una distancia mayor del aerogenerador, lo que reduce la cantidad de iteraciones

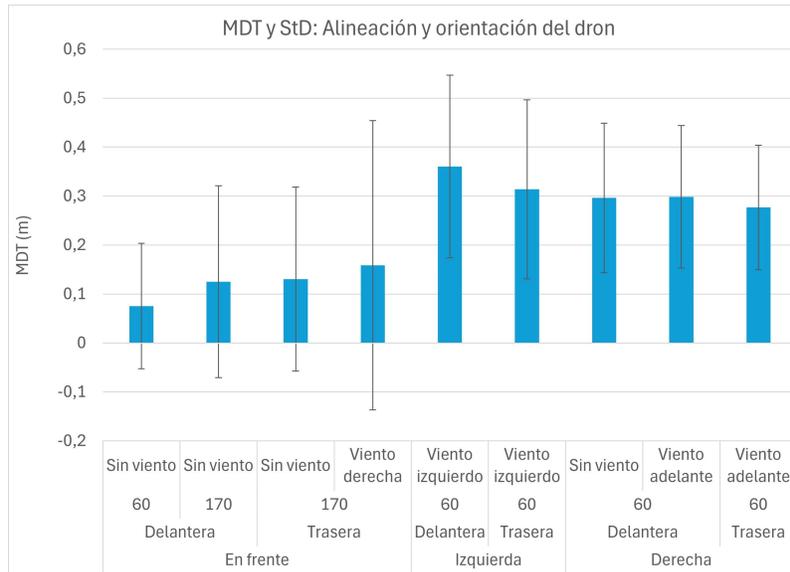


Figura 4.2: Gráfica de la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT) durante la etapa de alineación y orientación del dron frente al aerogenerador.

necesarias para alcanzar una posición estable. Por ejemplo, en una inspección trasera con viento desde la izquierda, la mediana del tiempo para alinearse fue de 210.5 segundos, mientras que la de una inspección delantera con igual posición inicial, distancia y viento tomó 330.0 segundos. Además, las desviaciones estándar en las inspecciones traseras son generalmente menores, lo que indica mayor consistencia en estas maniobras.

La presencia de viento también juega un papel determinante en los resultados. Los escenarios sin viento muestran una mayor estabilidad, reflejada en menores valores de desviación estándar tanto para la distancia como para el tiempo de alineación.

La posición inicial del dron también influye significativamente. Cuando el dron parte desde una posición en frente, los tiempos de alineación son considerablemente menores debido a que comienza más cerca de una orientación adecuada respecto al aerogenerador. Por el contrario, las posiciones iniciales desde la izquierda o la derecha requieren ajustes más extensos, lo que se traduce en tiempos mayores y mayores desviaciones estándar.

En conclusión, el desempeño del dron es más eficiente en condiciones “ideales”, es decir, cuando comienza frente al aerogenerador y sin viento. Las inspecciones traseras presentan una ventaja adicional, ya que son más rápidas y consistentes que las delanteras debido a la menor cantidad de ajustes necesarios para lograr la alineación adecuada. No obstante, si bien las condiciones adversas, como la presencia de viento o posiciones iniciales desalineadas, aumentan tanto la variabilidad como el tiempo requerido para la alineación, los resultados demuestran un buen desempeño general.

4.3.2. Inspección de las Aspas del Aerogenerador

Para la inspección, se utiliza como punto de partida una posición posible obtenida del paso anterior. Con este fin, se coloca el dron frente al aerogenerador en la tierra y se ajusta la máquina

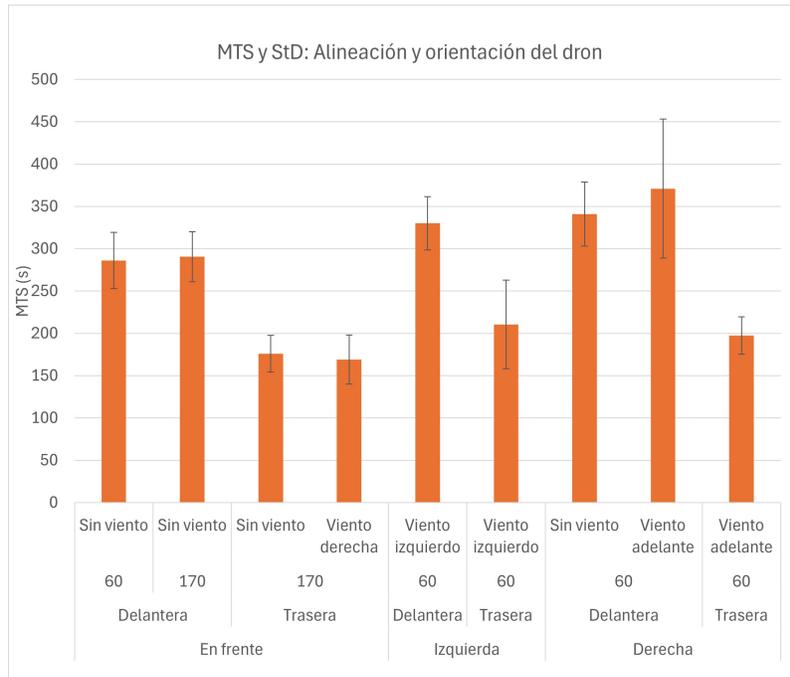


Figura 4.3: Gráfica de la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS) durante la etapa de alineación y orientación del dron frente al aerogenerador.

de estados para transicionar directamente a la fase de inspección luego del despegue, en lugar de acercarse y posicionarse de forma ortogonal. Esto tiene como beneficio que las condiciones para la inspección no dependen de estados pasados, sino que el estado inicial es más parecido al deseado.

Dron ortogonal al aerogenerador centrado en el rotor

Una de las pruebas se realiza colocando el dron ortogonal al aerogenerador, centrado en el rotor. Los resultados se pueden ver en la Tabla 4.4. En la Figura 4.4 y Figura 4.5 se pueden observar los mismos datos con énfasis en la desviación de la trayectoria y del tiempo, respectivamente.

Inspección	Viento	MDT (m)	StD MDT (m)	MTS (s)	StD MTS (s)	# correcciones
Delantera	Sin viento	0.2460	0.0883	968.0	30.9504	21
	Viento derecha	0.2850	0.0982	1024.0	35.7410	15
Trasera	Sin viento	0.2551	0.0808	1129.0	38.3244	14
	Viento adelante	0.2584	0.0753	1149.5	34.5393	16

Tabla 4.4: Resultados de la inspección en escenarios donde el dron inicia la inspección ortogonal al aerogenerador. Las métricas consideradas son: la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT), la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS), además del número de correcciones realizadas. En los escenarios con viento, se asumió una velocidad constante de 10 m/s. Estas pruebas fueron realizadas en la PC 1.

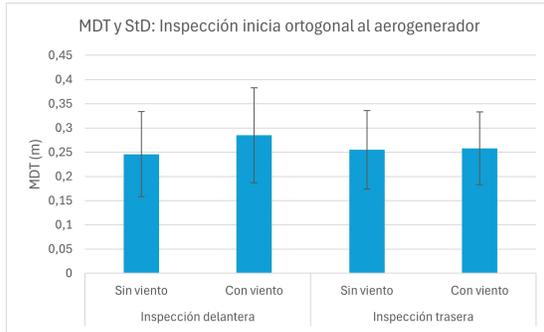


Figura 4.4: Gráfica de la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT) en escenarios donde el dron inicia la inspección de manera ortogonal al aerogenerador, considerando inspecciones delantera y trasera, con y sin la presencia de viento.

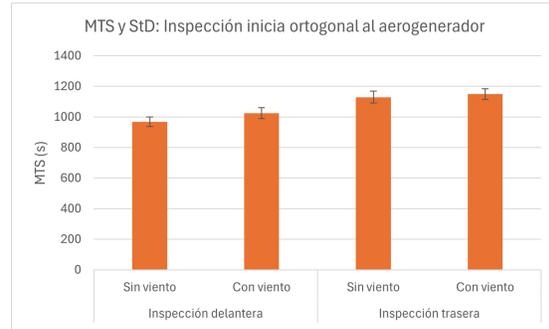


Figura 4.5: Gráfica de la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS) en escenarios donde el dron comienza la inspección de forma ortogonal al aerogenerador, abarcando inspecciones delantera y trasera, tanto con como sin viento.

Dron ortogonal al aerogenerador a la izquierda del rotor

Se toma como posición de partida una situación que no ha ocurrido en las pruebas de ortogonalidad, pero se busca saber cómo se comportaría el sistema ante una situación inicial adversa para la inspección. En este caso, el dron se encuentra desplazado hacia un lado del rotor, en lugar de estar correctamente centrado. Los resultados obtenidos se presentan en la Tabla 4.5. Asimismo, en la Figura 4.6 y en la Figura 4.7 se visualizan estos mismos datos, destacando específicamente la desviación de la trayectoria y el tiempo, respectivamente.

Inspección	Viento	MDT (m)	StD MDT (m)	MTS (s)	StD MTS (s)	# correcciones
Delantera	Sin viento	0.4199	0.1323	809.0	102.1853	24
	Viento izquierdo	0.4152	0.1421	816.0	111.9212	26
Trasera	Sin viento	0.4370	0.1397	976.0	89.8323	12
	Viento izquierdo	0.4474	0.0896	677.0	66.0425	12

Tabla 4.5: Resultados de la inspección en escenarios donde el dron inicia la inspección desde el lado izquierdo del rotor. Las métricas consideradas son: la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT), la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS), además del número de correcciones realizadas. En los escenarios con viento, se asumió una velocidad constante de 10 m/s. Estas pruebas fueron realizadas en la PC 2.

En cuanto a las condiciones de viento, el sistema se comporta de manera satisfactoria en todas las pruebas realizadas con 10 m/s, sin importar la dirección del viento. Como es de esperar, la desviación en la trayectoria aumenta con velocidades de viento más altas, pero siempre se mantiene dentro de un rango aceptable menor a 0.45m. Esto indica que el sistema es capaz de compensar los

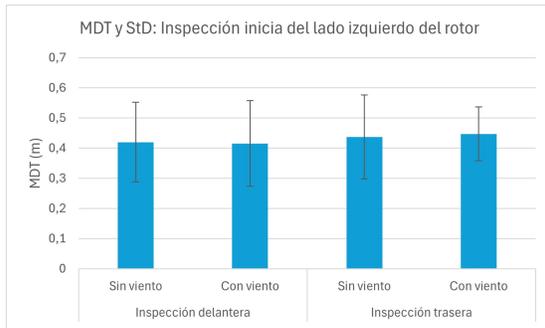


Figura 4.6: Gráfica de la Mediana de la Desviación de la Trayectoria (MDT) y su Desviación Estándar (StD MDT) en escenarios donde el dron inicia la inspección del lado izquierdo del rotor, considerando inspecciones delantera y trasera, con y sin la presencia de viento.

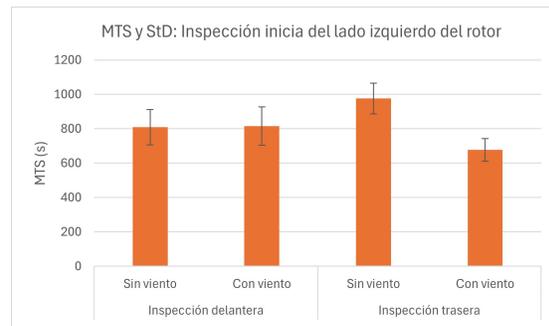


Figura 4.7: Gráfica de la Mediana del Tiempo Simulado (MTS) y su Desviación Estándar (StD MTS) en escenarios donde el dron inicia la inspección del lado izquierdo del rotor, abarcando inspecciones delantera y trasera, tanto con como sin viento.

efectos del viento hasta cierto punto, manteniendo una trayectoria lo suficientemente precisa para una inspección efectiva.

Adicionalmente, para cada uno de los entornos se intentó realizar una prueba adicional con viento de 15 m/s para evaluar el rendimiento y ocurrió que el dron no era capaz de despegar, por lo que no se puede garantizar que el dron pueda volar ante vientos similares o mayores.

El resto de las inspecciones fueron satisfactorias en cuanto a su finalización y a que las aspas estuvieran dentro del campo de visión de la cámara en todo momento. Para lograr esto, fue crucial el análisis en tiempo real del flujo de datos del LiDAR para detectar el aspa y realizar correcciones en la posición del dron.

Es destacable que, incluso comenzando la inspección en una condición no favorable—como estar posicionado lateralmente alejado del rotor—el sistema logra recomponerse ante una falla en la etapa anterior de posicionamiento, lo que demuestra su robustez y capacidad de adaptación. Sin embargo, es importante aclarar que si al inicio no se captura una parte del aspa (por ejemplo, si el dron no comienza la inspección desde la base del aspa, sino más cerca de la punta), esa sección no será registrada posteriormente. El dron no detecta que ha quedado una parte sin inspeccionar ni retrocede para cubrirla. El plan de vuelo teórico continúa como si el aspa fuera más larga de lo que realmente es, y el sistema entra en modo de corrección tantas veces como sea necesario hasta llegar al último punto para esa aspa. Esto sucede porque el dron detecta que es aspa podría quedar fuera de su campo de visión y ajusta su posición en consecuencia. Por lo tanto, la primera aspa no recibirá una inspección completa, pero a partir de las siguientes, el dron continuará el recorrido correctamente, ya que el error se corrige durante la primera pasada gracias a las correcciones realizadas.

Se puede observar en Tabla 4.5 y Tabla 4.4 la importancia de poder corregir la trayectoria planeada por el planificador de caminos (ver Sección 3.2) principalmente cuando se comienza la inspección desde una posición no favorable como es no centrado en el rotor sino que a la izquierda del mismo. La cantidad de correcciones realizadas en la inspección del frente es hasta un 50 % más. Esto se debe a que al llegar al final del aspa lo planificado es seguir (ya que solo se recorrió una

parte). Al intentar seguir, el aspa empieza a posicionarse cerca del borde de la imagen por lo que el dron busca corregir y de esta manera aumentan la cantidad de correcciones. Esto no sucede con la inspección trasera debido a que el porcentaje del aspa visible es mayor al posicionarse alrededor de 10 metros más lejos.

Imágenes en las inspecciones

Tomando una foto cada cierta distancia recorrida, más detalles en Sección 3.6, se capturan entre 6 y 8 imágenes por aspa en el aerogenerador de prueba utilizado. El solapamiento es alrededor de un 25 % según la orientación de la misma y la relación de aspecto de la cámara.

La Figura 4.8 muestra el resultado de una pasada en particular. Se puede observar el solapamiento, intencional, que busca asegurarse de que no falte alguna parte del aspa por buscar minimizar la cantidad de imágenes tomadas.

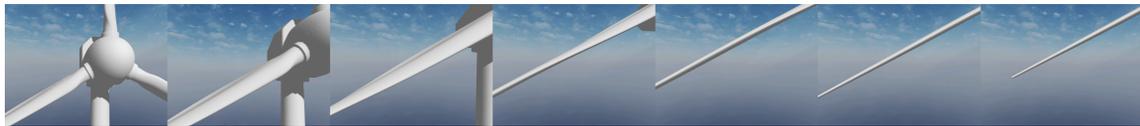


Figura 4.8: Imágenes tomadas por el dron al realizar una pasada por el aspa inferior izquierda en una inspección frontal.

4.4. Análisis de las experimentaciones

Las pruebas realizadas evidencian que el sistema es capaz de posicionar y controlar el dron de manera efectiva, así como llevar a cabo la inspección de las aspas del aerogenerador en diversos escenarios y condiciones ambientales. No obstante, es importante destacar que, debido a la simplificación de los modelos físicos utilizados, podrían existir discrepancias entre los resultados de la simulación en Gazebo y la realidad. A continuación, se presenta un análisis detallado de los resultados obtenidos.

4.4.1. Capacidad de posicionamiento ortogonal

El dron logró posicionarse ortogonalmente respecto a las aspas del aerogenerador desde distintas posiciones iniciales y bajo diferentes condiciones de viento. Al aproximarse desde el frente a distancias cortas y sin viento, se observó una alta precisión en el posicionamiento, con desviaciones mínimas en la trayectoria y tiempos de aproximación razonables.

Al aumentar la distancia inicial y en presencia de viento, se registró un ligero incremento en las desviaciones y en los tiempos de operación. Sin embargo, estos incrementos fueron marginales y no afectaron significativamente el desempeño general del sistema.

4.4.2. Desempeño durante la inspección

Durante la fase de inspección, asumiendo posiciones de partida obtenidas del paso anterior, el dron siguió de cerca la trayectoria planificada. Comenzando ortogonal al aerogenerador, tanto en el

lado delantero como trasero, el sistema mantuvo desviaciones dentro de límites aceptables, incluso al enfrentar vientos de 10 m/s.

Al inspeccionar desde posiciones laterales, las desviaciones aumentaron ligeramente, lo cual es atribuible a que el sistema tuvo que hacer más correcciones (debido a que el aspa quedaría fuera de la imagen) las cuales no tienen puntos intermedios y por lo tanto PX4 busca la mejor manera de ir de un punto al otro, no necesariamente en línea recta. Esto resulta en mayores distancias del dron a la recta entre el setpoint previo a la corrección.

4.4.3. Impacto de las condiciones ambientales

Los resultados indican que el sistema opera eficazmente en condiciones de viento que son típicas en parques eólicos. El control del dron no se vio significativamente afectado por vientos moderados provenientes de distintas direcciones. Esto es esencial para aplicaciones reales, donde las condiciones climáticas pueden variar y el sistema debe adaptarse sin comprometer la seguridad.

De todos modos, notamos que al aumentar el viento y en distancias entre setpoints largas, el dron no realiza una recta como si se podía observar en condiciones con vientos normales a bajos, lo cual indica que para soportar mayores vientos se tendrían que realizar ajustes en este sentido.

4.4.4. Eficiencia temporal y energética

Sumando los tiempos de operación registrados en las dos etapas, se tiene que las inspecciones están siempre por debajo de los 30 minutos, incluso luego de inspeccionar dos de las cuatro caras de cada aspa en este tiempo. Drones como el utilizado durante la visita al parque eólico *Los Caracoles* (DJI M200) tienen una autonomía máxima de 37 minutos ² por lo que restaría alrededor de un 20% de autonomía.

La holgura es aún mayor con el dron recientemente adquirido por la Facultad de Ingeniería, un DJI M350, que cuenta con un tiempo de vuelo máximo de 55 minutos ³. Con este tiempo podría llegar a ser matemáticamente posible inspeccionar las cuatro caras de cada aspa (rotando el aerogenerador sobre el eje en el medio) sin aterrizar aunque muy justo, teniendo en cuenta que la rotación lleva tiempo y además el dron tendría que volver a realizar alineación para colocarse ortogonal nuevamente.

4.4.5. Eficiencia computacional

Con el objetivo de analizar los recursos computacionales necesarios para operar el sistema en un entorno real se realizó una corrida en la PC 1, monitoreando el uso de CPU y RAM. En esta evaluación, se excluyeron las tareas relacionadas con Gazebo, ya que su carga computacional está relacionada únicamente con la simulación. Los resultados mostraron que las tareas relacionadas con el sistema tuvieron un consumo aproximado de 25% de CPU y 450 MB de RAM a lo largo de la ejecución.

Es importante señalar que, aunque este análisis indica que no se detectó un alto costo computacional en nuestro caso, no podemos determinar los requisitos técnicos mínimos para que el sistema se integre eficientemente en una computadora de abordo o cuál sería la opción más conveniente.

²<https://www.dji.com/global/support/product/matrice-200-series-v2>

³<https://enterprise.dji.com/matrice-350-rtk/specs>

4.5. Ejemplos de Ejecuciones del Sistema

Con el objetivo de demostrar el funcionamiento del sistema desarrollado para la inspección, se ha preparado una lista de reproducción con ejemplos de ejecuciones del mismo.

La lista de reproducción está disponible en el siguiente enlace:

Lista de reproducción del sistema.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones

Se diseñó e implementó un sistema para la inspección de las aspas de aerogeneradores utilizando un dron autónomo, integrando técnicas avanzadas de visión por computadora, como la transformada de Hough, y tecnología LiDAR para lograr un posicionamiento preciso del dron en relación con el aerogenerador. Además, se desarrollaron algoritmos de planificación de trayectorias óptimas basados en el Problema del Viajante Asimétrico. El sistema incluye un módulo de corrección en tiempo real y un módulo de adquisición de imágenes para documentar el estado de las aspas.

El sistema propuesto ha demostrado ser efectivo y robusto en la simulación de inspecciones de aerogeneradores utilizando drones. Los resultados respaldan su viabilidad para aplicaciones reales, permitiendo realizar inspecciones bajo diversas condiciones, incluyendo escenarios con viento. El dron logró completar tanto el acercamiento y posicionamiento ortogonal como la inspección de las aspas desde diferentes puntos sin verse afectado más allá de las limitaciones físicas propias del dispositivo.

Este sistema no solo tiene el potencial de mejorar la consistencia en las inspecciones de aerogeneradores, sino que también contribuye a mejorar la seguridad y reducir los costos operativos en los parques eólicos. Aunque no es parte del alcance del proyecto, se pueden integrar algoritmos de aprendizaje automático para la detección automática de fallos, una integración de este estilo abre la posibilidad de automatizar por completo las inspecciones en el futuro. Por otra parte, nuestra implementación es de código abierto, lo que permite a otros utilizarlo como base para el desarrollo de futuros avances e innovaciones en este campo.

En cuanto a la selección de herramientas, se optó por utilizar tecnologías ampliamente reconocidas en el ámbito de la robótica, lo cual resultó en una experiencia mayoritariamente fluida. Sin embargo, surgieron dificultades importantes en la comunicación entre ROS 2 y PX4 a través del protocolo de mensajería MAVLink, ya que no se pudo gestionar correctamente la concurrencia de MAVSDK (la implementación en Python de MAVLink) utilizando la biblioteca rospy. Como solución, se optó por utilizar los tópicos de ROS 2 con mensajes uORB en lugar de MAVLink.

Además, encontrar un modelo de dron con gimbal funcional para Gazebo fue un desafío, y finalmente se utilizó uno cuya implementación de funcionalidades no estaba completamente finalizada. Por otro lado, para Gazebo Classic (la versión original y generación anterior de esta plataforma de simulación), fue más sencillo identificar modelos de dron con gimbal completamente funcional.

Al inicio del proyecto, el equipo no contaba con experiencia previa en robótica, por lo que una parte significativa del tiempo inicial se dedicó a familiarizarse con las herramientas y tecnologías requeridas. Este proceso de aprendizaje fue esencial para poder afrontar los desafíos del proyecto. Con el tiempo, se logró un dominio sólido de las tecnologías involucradas, lo que nos permitió avanzar con éxito en el desarrollo y prueba del sistema, consolidando nuestras habilidades y dejándonos bien preparados para futuros desafíos en proyectos similares.

5.2. Trabajo futuro

A pesar de los avances logrados en este proyecto y del cumplimiento de los objetivos establecidos, existen algunos aspectos que, debido a las limitaciones de tiempo, no fueron implementados en la versión actual del sistema. A continuación, se presentan las principales áreas que podrían ser mejoradas o expandidas en futuras versiones.

5.2.1. Uso de una flota de drones para la inspección

Una extensión lógica del proyecto sería realizar la inspección utilizando una flota de drones en lugar de un solo dron. Este enfoque permitiría reducir significativamente el tiempo de la inspección, ya que, por ejemplo, tres drones podrían inspeccionar simultáneamente cada una de las tres aspas del aerogenerador. Sin embargo, esto introduce una mayor complejidad en la coordinación y planificación de rutas. Se necesitarían algoritmos de sincronización y comunicación entre los drones para evitar colisiones.

5.2.2. Módulo para evitar obstáculos

Durante el desarrollo del proyecto, se asumió como hipótesis de trabajo la ausencia de obstáculos en el trayecto del dron hacia el aerogenerador. No obstante, para mejorar la robustez y aplicabilidad del sistema en entornos reales, se plantea como trabajo futuro la implementación de un módulo de evitación de obstáculos. Este módulo permitiría al dron detectar y esquivar automáticamente elementos en su entorno, como pueden ser aves o incluso otros drones con los que pudiera cruzarse en su camino.

5.2.3. Reanudación de inspecciones interrumpidas

Implementar un sistema que permita mantener el estado de la inspección sería de gran utilidad en situaciones donde el dron se quede sin batería o deba interrumpir la operación por cualquier motivo. De esta manera, al reanudar la misión, el dron podría continuar desde el punto donde se detuvo, evitando la repetición innecesaria de secciones ya inspeccionadas.

5.2.4. Planes de contingencia ante errores

El manejo de errores del sistema puede mejorarse para garantizar mayor seguridad y adaptabilidad en situaciones críticas. Actualmente, el usuario puede abortar la misión manualmente en cualquier momento enviando un mensaje a un tópico.

El dron cuenta además con varios mecanismos de failsafe preconfigurados que se activan automáticamente en distintas emergencias. En la solución se mantuvo la configuración predeterminada del modelo del dron utilizado. Entre los principales mecanismos, se incluyen:

- Pérdida de Conexión con el Control Remoto o Estación Base (RC Loss): Se activa el modo Return-to-Home (RTH), haciendo que el dron regrese automáticamente a su punto de origen.
- Failsafe de Batería Baja: Al alcanzar un nivel de batería del 15 %, el sistema emite una advertencia. Si la batería desciende al 7 %, se activa el failsafe, y el dron entra en modo RTH.

Aunque estas configuraciones básicas ofrecen una capa básica de seguridad, sería recomendable personalizarlas para atender las necesidades específicas del proyecto. Esto incluiría optimizar los parámetros de seguridad y diseñar nuevos planes de contingencia para garantizar un desempeño más robusto en situaciones de emergencia.

En este aspecto, también sería favorable la posibilidad de que el dron pueda ser operado directamente mediante un control remoto por el operador. Esto es factible en la implementación gracias a la existencia del modo “Position” en PX4, utilizado para el manejo remoto¹.

5.2.5. Mejora de la integración del Planificador de Caminos

Como se mencionó en la Subsección 3.2.2, el planificador de caminos acepta dos ángulos que representan los giros de fase y orientación del aerogenerador respecto a su posición ideal, lo que permite considerar situaciones en las que el aerogenerador no se encuentre en su alineación óptima. Esta posición ideal corresponde a un aerogenerador perfectamente alineado en Y invertida y ortogonal al dron.

Actualmente, no se dispone de un método para calcular o detectar estos ángulos, por lo que no se están pasando como parámetros en la solución implementada. Esta limitación impide aprovechar al máximo las capacidades del planificador. A pesar de ello, como se ha mencionado en diversas secciones del documento, se emplean otros mecanismos que permiten mantener al dron en una posición adecuada respecto al aerogenerador, garantizando un funcionamiento correcto incluso cuando la alineación no es perfecta.

No obstante, incorporar los ángulos como parámetros optimizaría el rendimiento del planificador, reduciendo la necesidad de correcciones en la trayectoria del dron y disminuyendo la carga sobre los otros mecanismos de ajuste. Aunque la solución actual funciona correctamente, la implementación de esta mejora reduciría la cantidad de ajustes requeridos.

Por otro lado, como se menciona en la Subsección 3.2.3, en la solución implementada, el planificador de caminos se ejecuta de manera “online”. Aunque esta implementación no presenta una diferencia significativa en la solución actual, genera un overhead innecesario que podría afectar la eficiencia general. Esto se debe a que los cálculos realizados podrían haberse efectuado antes del despegue del dron, dado que las dimensiones del aerogenerador ya están disponibles de antemano.

Aunque este overhead es pequeño, podría volverse más relevante si se agregaran más puntos de inspección. Para mejorar este aspecto, se podría realizar el cálculo antes del despegue del dron. Luego, una vez que el dron se haya posicionado ortogonalmente respecto al plano de las aspas del aerogenerador, los puntos generados por el planificador podrían ser rotados mediante transformaciones matriciales. De esta forma, el recorrido generado se ajustaría a la posición real del aerogenerador y a la ubicación del dron en el momento de la inspección, proporcionando un camino

¹https://docs.px4.io/main/en/flight_modes/mc/position.html

similar al que se obtiene en la solución actual, pero reduciendo la cantidad y complejidad de los cálculos realizados después del despegue. Esto es clave, ya que optimizar el uso de la batería del dron es fundamental.

5.2.6. Mejoras en la adquisición de imágenes

Actualmente, el sistema captura fotografías a intervalos de distancia recorrida. Una mejora futura sería implementar un sistema de recolección de imágenes más avanzado que permita definir un grado específico de superposición entre imágenes.

Esto permitiría la construcción de un modelo visual continuo de las aspás—una representación detallada y sin interrupciones de la superficie—y también se podría generar un “mosaico” de imágenes, una composición bidimensional que alinea múltiples fotos para mostrar áreas extensas. Con suficiente información tridimensional, el modelo podría representar con precisión la estructura monitoreada, sirviendo como entrada para módulos de análisis avanzados que realicen la identificación de fallas.

Además, será necesario explorar e implementar un mecanismo que permita asociar las imágenes recolectadas con sus ubicaciones en el aerogenerador, lo que permitiría la localización de posibles fallas en caso de ser detectadas.

5.2.7. Pruebas Unitarias

Si bien se realizaron pruebas de funcionamiento del sistema en Gazebo, no se llevaron a cabo pruebas unitarias de los módulos. Implementar estas pruebas en el futuro permitiría una evaluación más exhaustiva de cada componente, facilitando la identificación de posibles errores y mejorando la robustez del sistema.

5.2.8. Implementación en el Mundo Real

Para implementar el sistema en una situación práctica, se podría utilizar uno de los kits que ofrece PX4², algunos de los cuales incluyen el dron X500, que es el mismo empleado en las simulaciones. Además, en el mismo sitio donde se venden los kits, se proporciona documentación detallada que puede ser útil también para otros casos.

Si no se desea utilizar uno de estos kits, es posible emplear cualquier dron equipado con una placa Pixhawk o controladores de vuelo similares, ya que PX4 es compatible con una amplia variedad de opciones³. Asimismo, hay disponibles diversos dispositivos compatibles, incluyendo algunos vendidos por DJI⁴.

El dron debe contar con un gimbal y los sensores necesarios para la misión, como una cámara RGB y un LiDAR. Alternativamente, en lugar de una cámara RGB estándar, se podría utilizar una cámara RGB-D, que proporciona información de profundidad junto con las imágenes.

El sistema de autopiloto PX4 se ejecuta en la computadora de vuelo del dron, mientras que QGroundControl debe instalarse en un dispositivo que no esté en el dron. Las opciones de conexión entre el dron y QGroundControl son: vía USB (no aplicable en este caso), telemetría por radio o WiFi. Esto permite la transmisión de datos en tiempo real.

²https://docs.px4.io/main/en/frames_multicopter/holybro_x500v2_pixhawk6c.html

³https://docs.px4.io/main/en/flight_controller/autopilot_manufacturer_supported.html

⁴https://docs.px4.io/main/en/frames_multicopter/dji_f450_cuav_5plus.html

En cuanto a la arquitectura de ROS 2, debe correr en una computadora a bordo. Hay algunas placas Pixhawk como la “Holybro Pixhawk Jetson Baseboard”⁵ que ya incluyen una computadora y, en caso de necesitarse más poder computacional, se puede usar una por separado. La documentación⁶ menciona un Raspberry Pi⁷ como una opción liviana y sugiere otras opciones menos conocidas en caso de requerir mayor poder computacional.

Asimismo, es posible que se deban ajustar ciertos parámetros para asegurar que el dron cumpla con los objetivos de la misión de manera eficaz y segura.

Por último, es crucial realizar pruebas para verificar la seguridad del sistema antes de su implementación final.

El sistema también podría verificar si el dron cuenta con la batería suficiente para completar la misión y si se cuenta con almacenamiento interno para guardar las imágenes de la inspección.

5.2.9. Seguridad Informática

Uno de los riesgos más preocupantes son los ataques de interceptación de comunicaciones, como el ataque Man in the Middle, que permitirían a actores malintencionados obtener control no autorizado del dron. Además, existe el riesgo de que un atacante realice un ataque DDoS (Denegación de Servicio Distribuida) sobre el controlador, lo que podría interrumpir su funcionamiento.

Para mitigar estos y otros riesgos potenciales, se deben implementar medidas de seguridad como el cifrado de las comunicaciones y la verificación de identidad mediante firmas digitales. También se sugiere la utilización de redes privadas para aumentar la seguridad del sistema.

⁵https://docs.px4.io/main/en/companion_computer/holybro_pixhawk_jetson_baseboard.html

⁶https://docs.px4.io/main/en/companion_computer/#companion-computer-options

⁷https://docs.px4.io/main/en/companion_computer/pixhawk_rpi.html

Bibliografía

- Amer, A., Mehndiratta, M., le Fevre Sejersen, J., Pham, H. X., & Kayacan, E. (2023). Visual Tracking Nonlinear Model Predictive Control Method for Autonomous Wind Turbine Inspection. *2023 21st International Conference on Advanced Robotics (ICAR)*, 431-438. <https://doi.org/10.1109/ICAR58858.2023.10406329>
- Car, M., Markovic, L., Ivanovic, A., Orsag, M., & Bogdan, S. (2020). Autonomous Wind-Turbine Blade Inspection Using LiDAR-Equipped Unmanned Aerial Vehicle. *IEEE Access*, 8, 131380-131387. <https://doi.org/10.1109/ACCESS.2020.3009738>
- Castelar Wembers, C., Pflughaupt, J., Moshagen, L., Kurenkov, M., Lewejohann, T., & Schildbach, G. (2024). LiDAR-based automated UAV inspection of wind turbine rotor blades. *Journal of Field Robotics*, 41, 1116-1132. <https://doi.org/10.1002/rob.22309>
- Castorrini, A., Cappugi, L., Bonfiglioli, A., & Campobasso, S. (2020). Assessing wind turbine energy losses due to blade leading edge erosion cavities with parametric CAD and 3D CFD. *Journal of Physics: Conference Series*, 1618, 052015. <https://doi.org/10.1088/1742-6596/1618/5/052015>
- Fuentes, A. (2024). *Problema del agente viajero* [Accedido: 6 de octubre de 2024]. Universidad Autónoma del Estado de Hidalgo. <https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html>
- Gazebo. (s.f.). *What is Gazebo?* [Accedido: 12 de octubre de 2024]. https://classic.gazebosim.org/tutorials?tut=guided_b1#WhatIsGazebo
- Google. (2024). *Traveling Salesperson Problem* [Accedido: 6 de octubre de 2024]. <https://developers.google.com/optimization/routing/tsp>
- Grindley, B., Phillips, K., Parnell, K. J., Cherrett, T., Scanlan, J., & Plant, K. L. (2024). Over a decade of UAV incidents: A human factors analysis of causal factors. *Applied Ergonomics*, 121, N.PAG. <https://doi.org/10.1016/j.apergo.2024.104355>
- Gu, W., Hu, D., Cheng, L., Cao, Y., Rizzo, A., & Valavanis, K. P. (2020). Autonomous Wind Turbine Inspection using a Quadrotor. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 709-715. <https://doi.org/10.1109/ICUAS48674.2020.9214066>
- Hwang, S.-G., An, Y.-K., Yang, J.-H., & Sohn, H. (2024). Remote Inspection of Internal Delamination in Wind Turbine Blades using Continuous Line Laser Scanning Thermography. *International Journal of Precision Engineering and Manufacturing - Green Technology, Preprints*, 1-14. <https://doi.org/10.1007/s40684-020-00192-9>
- Kavlakoglu, E., & Winland, V. (2024). *What is k-means clustering?* [Accedido: 14 de noviembre de 2024]. IBM. <https://www.ibm.com/topics/k-means-clustering>
- Levy-Kramer, J. (2018, abril). *k-means-constrained*. <https://github.com/joshlk/k-means-constrained>

- MAVLink. (s.f.). *MAVLink Developer Guide* [Accedido: 13 de noviembre de 2024]. <https://mavlink.io/en/>
- Memari, M., Shakya, P., Shekaramiz, M., Seibi, A. C., & Masoum, M. A. S. (2024). Review on the Advancements in Wind Turbine Blade Inspection: Integrating Drone and Deep Learning Technologies for Enhanced Defect Detection. *IEEE Access*, *12*, 33236-33282. <https://doi.org/10.1109/ACCESS.2024.3371493>
- Okoli, C. (2015). A Guide to Conducting a Standalone Systematic Literature Review. *Communications of the Association for Information Systems*, *37*, pp-pp. <https://doi.org/10.17705/1CAIS.03743>
- Open Source Robotics Foundation. (s.f.[a]). *Getting Started with ROS* [Accedido: 12 de octubre de 2024]. ROS Project. <https://www.ros.org/blog/getting-started/>
- Open Source Robotics Foundation. (s.f.[b]). *Why ROS?* [Accedido: 12 de octubre de 2024]. ROS Project. <https://www.ros.org/blog/why-ros/>
- PX4 Development Team. (s.f.[a]). *Software Overview* [Accedido: 16 de noviembre de 2024]. PX4 Autopilot Project. <https://px4.io/software/software-overview/>
- PX4 Development Team. (s.f.[b]). *Using the ECL EKF* [Accedido: 13 de noviembre de 2024]. PX4 Autopilot Project. https://docs.px4.io/main/en/advanced_config/tuning_the_ecl_ekf.html
- QGroundControl. (s.f.). *QGroundControl User Guide* [Accedido: 16 de octubre de 2024]. <https://docs.qgroundcontrol.com/master/en/qgc-user-guide/>
- Rao, Y., Xiang, B. J., Huang, B., & Mao, S. (2019). Wind Turbine Blade Inspection Based on Unmanned Aerial Vehicle(UAV) Visual Systems. *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*, 708-713. <https://doi.org/10.1109/EI247390.2019.9062226>
- Stokkeland, M., Klausen, K., & Johansen, T. A. (2015). Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 998-1007. <https://doi.org/10.1109/ICUAS.2015.7152389>
- Tanriverdi, H., Karakuş, G., & Ulukan, A. (2023). Wind turbine inspection with drone: Advantages and disadvantages. *Journal of Energy Systems*, *7*(1), 57-66. <https://doi.org/10.30521/jes.1148877>
- Uruguay XXI. (2023). Energías renovables en Uruguay [Accessed: 2024-09-03]. <https://www.uruguayxxi.gub.uy/es/centro-informacion/articulo/energias-renovables/>
- Yu, J. (2014). *Traveling Salesman Problem* [Accedido: 14 de noviembre de 2024]. Cornell University Computational Optimization Open Textbook. https://optimization.cbe.cornell.edu/index.php?title=Traveling_salesman_problem

Anexo A

5.3. Glosario

Ángulo de guiñada (yaw) Refiere a la rotación del dron alrededor de su eje vertical, lo que controla la dirección en la que apunta la parte frontal del dron.

Ángulo de cabeceo (pitch) Describe la inclinación hacia adelante o hacia atrás, controlando el movimiento sobre el eje lateral.

Ángulo de alabeo/balaceo (roll) Refiere a la inclinación lateral sobre el eje longitudinal del dron.

ATSP (Asymmetric Travelling Salesman Problem) Variante del Problema del Viajante de Comercio (TSP) donde las distancias entre las ciudades no son simétricas, es decir, el costo de ir de una ciudad A a una ciudad B puede ser diferente al de ir de B a A.

EKF (Extended Kalman Filter) Es una versión del filtro de Kalman diseñada para estimar el estado de sistemas no lineales. En PX4, se utiliza para fusionar datos de sensores como GPS, acelerómetros y giroscopios, estimando de manera precisa el estado del dron (como la posición, velocidad, orientación y aceleración).

FOV (Field Of View) Es el área o ángulo visible que un dispositivo, como una cámara o el ojo humano, puede capturar o percibir en un determinado momento.

Gazebo Es un simulador de robótica de código abierto que permite modelar, simular y visualizar robots, sensores y entornos 3D.

GCS (Ground Control Station) Es una estación en tierra que permite monitorear y controlar un dron. Se utiliza para recibir datos en tiempo real, como imágenes y telemetría, y enviar comandos de control al dron durante su operación.

Gimbal Es un dispositivo de estabilización utilizado para mantener un objeto (como una cámara o sensor) nivelado y estable, independientemente del movimiento o rotación de la plataforma a la que esté conectado. Esto se logra mediante un sistema de ejes que permite el movimiento libre en diferentes direcciones.

LiDAR (Light Detection And Ranging) Es una tecnología de medición remota que utiliza luz en forma de láser para medir distancias. Se emplea para crear mapas 3D de superficies y objetos.

MAVLink (Micro Air Vehicle Link) Es un protocolo de comunicación ligero y eficiente utilizado para intercambiar mensajes entre drones y estaciones de control, como sistemas de control de vuelo y estaciones terrestres.

PX4 Es un software de autopiloto de código abierto para vehículos aéreos, terrestres y marinos, utilizado para controlar drones y otros vehículos autónomos.

QGroundControl Es una estación de control terrestre de código abierto. Es compatible con varios sistemas de autopiloto, como PX4 y ArduPilot.

RViz Es un visualizador 3D para el marco de trabajo del Sistema Operativo de Robot (ROS).

ROS Es un conjunto de herramientas y bibliotecas que facilita el desarrollo de aplicaciones para robots.

Setpoint Un punto de referencia dinámico en el espacio que se proporciona al sistema de control de un robot o dron para indicar la posición, velocidad, u orientación deseada en un instante específico.

Tópico Son canales de comunicación que permiten a los nodos intercambiar mensajes de forma asíncrona. Un nodo puede publicar mensajes en un tópico, y otros nodos pueden suscribirse para recibir esos mensajes.

TSP (Travelling Salesman Problem) El Problema del Viajante de Comercio busca encontrar el camino más corto que pase por un conjunto de puntos (ciudades) de un grafo, visitando una vez cada una y regresando al punto de inicio.

UAV (Unmanned Aerial Vehicle) Vehículo aéreo no tripulado que opera sin un piloto a bordo. Sinónimo de dron.

Waypoint Un punto fijo en una ruta predefinida que el robot o dron debe seguir como parte de una misión planificada.

YOLO(You Only Look Once) Es un algoritmo de detección de objetos en imágenes que predice las ubicaciones y categorías de los objetos en una sola pasada a través de una red neuronal.

Anexo B

5.4. Herramientas y Tecnologías

5.4.1. ROS 2

ROS (Robot Operating System) es un conjunto de herramientas de software de código abierto utilizado para crear aplicaciones robóticas. Proporciona una colección de herramientas y bibliotecas que permiten integrar componentes como sensores, actuadores y algoritmos, facilitando el desarrollo de sistemas robóticos complejos. Es ampliamente usado en la investigación académica, la enseñanza y la industria, y ha creado un ecosistema robusto gracias a su comunidad global.

ROS 2, una evolución de ROS, mejora aspectos como la comunicación distribuida y la escalabilidad, haciéndolo adecuado para sistemas más complejos y entornos industriales. Soporta múltiples plataformas (Linux, Windows, macOS, sistemas embebidos) y aplicaciones en tiempo real. Su diseño flexible permite su aplicación en diversos dominios, desde robótica doméstica hasta sistemas autónomos en industria, automoción y exploración espacial (Open Source Robotics Foundation s.f.[b]).

ROS se distribuye en versiones conocidas como “distribuciones” o “distros”, de las cuales se mantiene soporte para múltiples de ellas. Algunas distribuciones tienen soporte a largo plazo (LTS), lo que significa que son más estables y han sido sometidas a pruebas exhaustivas. Otras versiones son más recientes y tienen ciclos de vida más cortos, pero ofrecen compatibilidad con plataformas y versiones más actuales de los paquetes ROS que las componen (Open Source Robotics Foundation s.f.[a]).

Para el desarrollo del proyecto se utilizó ROS 2 Humble Hawksbill, la versión LTS recomendada de ROS 2 a día de hoy. Se recomienda utilizarlo con Ubuntu Linux 22.04 o Windows 10; en nuestro caso, preferimos el sistema Linux.

5.4.2. Gazebo

Gazebo es un simulador dinámico en 3D que permite simular robots en entornos complejos, tanto interiores como exteriores.

Gazebo proporciona una simulación física con un grado de fidelidad mucho mayor a motores de juegos, así como una suite de sensores y diversas interfaces tanto para usuarios como para programas. Las aplicaciones típicas de Gazebo abarcan la prueba de algoritmos de robótica, el diseño de robots y la realización de pruebas de regresión en escenarios realistas.

Entre las características destacadas de Gazebo se incluyen múltiples motores físicos, una extensa biblioteca de modelos de robots y entornos, una amplia variedad de sensores, así como interfaces

gráficas y programáticas convenientes (Gazebo s.f.).

Una release de Gazebo consiste en un conjunto de bibliotecas de Gazebo versionadas que siguen el versionado semántico. La versión principal de una biblioteca no cambia con cada release de Gazebo, lo que permite desarrollar código sin preocuparse por cambios disruptivos.

En un principio se quiso utilizar Gazebo Harmonic por ser el último release LTS a la fecha, pero debido a problemas de compatibilidad con ROS 2 se utilizó Gazebo Garden, el release anterior.

5.4.3. QGroundControl

QGroundControl es una estación de control terrestre de código abierto, que permite control de vuelo y configuración de vehículos aéreos no tripulados, diseñada para sistemas que utilizan PX4 o ArduPilot. Entre sus características principales se encuentran la planificación de misiones de vuelo autónomo, visualización en tiempo real del mapa de vuelo y la posición del vehículo, así como la posibilidad de gestionar múltiples vehículos (QGroundControl s.f.).

5.4.4. MAVLink

MAVLink (Micro Air Vehicle Link) es un protocolo de mensajería liviano utilizado para la comunicación con drones y entre los componentes a bordo del dron. MAVLink fue lanzado por Lorenz Meier en 2009 y actualmente tiene una amplia base de colaboradores. Sigue un patrón de diseño híbrido de publicación-suscripción y punto a punto: los flujos de datos se publican como tópicos, mientras que los protocolos de configuración como el de misión o parámetros son punto a punto con retransmisión. Los mensajes se definen en archivos XML, y cada archivo define el conjunto de mensajes soportados por un sistema MAVLink específico, conocido como “dialecto” (MAVLink s.f.).

5.4.5. PX4

PX4 es un software de control de vuelo de código abierto diseñado para drones y otros vehículos no tripulados. El proyecto ofrece un conjunto flexible de herramientas para que los desarrolladores de drones compartan tecnologías y creen soluciones personalizadas para aplicaciones de drones.

PX4 forma parte de Dronecode, una organización sin fines de lucro administrada por la Fundación Linux, que promueve el uso de software de código abierto en vehículos aéreos. Dronecode también alberga QGroundControl y MAVLink (PX4 Development Team s.f.[a]).

5.4.6. EKF

El ECL EKF (Filtro de Kalman Extendido de la Biblioteca de Estimación y Control) es un componente clave en el sistema PX4 que permite la fusión de datos de diversos sensores para estimar el estado del vehículo. El EKF procesa medidas de sensores como el IMU, magnetómetro y GPS para obtener estimaciones precisas de variables críticas como la posición, velocidad y orientación.

Utiliza una técnica conocida como “fusion time horizon” (horizonte de tiempo de fusión) para compensar los retrasos y asegurar que las actualizaciones de estado se sincronicen con los datos de entrada. Este método define un período planificado en el que el sistema recopila, analiza y combina los datos entrantes, incluso cuando estos llegan con distintas latencias.

En PX4, el EKF permite modos de operación que se adaptan a las combinaciones de sensores disponibles, brindando flexibilidad para abordar condiciones cambiantes, como la presencia de anomalías magnéticas o la necesidad de operar sin magnetómetro (PX4 Development Team s.f.[b]).

5.4.7. Google OR-Tools

Google OR-Tools es una librería de código abierto diseñada para resolver problemas complejos en áreas como el enrutamiento de vehículos, la programación lineal, entre otras.

Particularmente, para la resolución del Problema del Viajante (ver Anexo 5.5.1 para más información), esta herramienta permite configurar la estrategia inicial de búsqueda de soluciones (*first_solution_strategy*) y la metaheurística de búsqueda local (*local_search_metaheuristic*), además de permitir definir un límite de tiempo para la ejecución.

La *first_solution_strategy* determina cómo se construye la solución inicial, utilizando métodos rápidos y sencillos, mientras que la *local_search_metaheuristic* se aplica para refinar y mejorar la solución inicial mediante técnicas de optimización (Google 2024).

5.5. Conceptos

5.5.1. Problema del Viajante Asimétrico (asymmetric TSP)

Un grafo $G(V, E)$ está compuesto por:

- V : un conjunto de vértices o nodos, $V = \{v_1, v_2, \dots, v_n\}$, donde cada v_i representa una ciudad.
- E : un conjunto de aristas que conectan a los nodos, $E = \{(v_i, v_j) : 1 \leq i, j \leq n, i \neq j\}$. En el caso del TSP asimétrico, las aristas no son necesariamente simétricas, es decir, el costo de ir de v_i a v_j puede no ser el mismo que el costo de ir de v_j a v_i .

El **Problema del Viajante (TSP)** es un problema de optimización que consiste en encontrar el camino más corto (o el ciclo) que pase por un conjunto de nodos en un grafo $G(V, E)$, visitando cada nodo exactamente una vez y regresando al nodo de origen, de manera que el costo total del recorrido se minimice.

El **Problema del Viajante Asimétrico (ATSP)** es una variante del TSP en la que los costos de viajar de un nodo a otro no son necesariamente los mismos en ambas direcciones. Es decir, el costo de ir de un nodo v_i a un nodo v_j puede diferir del costo de ir de v_j a v_i . En este caso, el objetivo sigue siendo encontrar el ciclo más corto que visite todos los nodos exactamente una vez, pero teniendo en cuenta que los costos entre nodos pueden ser asimétricos (Yu 2014).

5.5.2. K-Means Clustering

K-means clustering es un algoritmo de aprendizaje no supervisado, lo que significa que no requiere datos etiquetados para funcionar. Agrupa puntos de datos en k clusters, minimizando la distancia entre los puntos y los centroides de sus respectivos clusters. Funciona de manera iterativa, asignando cada punto al centroide más cercano y actualizando las posiciones de los centroides basándose en la media de los puntos asignados a cada cluster. Este proceso se repite hasta que los centroides convergen o se alcanza un número máximo de iteraciones. Aunque es un método sencillo, k-means es sensible a las condiciones iniciales y a los valores atípicos, lo que puede afectar la calidad de los resultados (Kavlakoglu y Winland 2024).

5.5.3. K-Means Constrained

K-means constrained es un paquete en Python que permite realizar agrupamiento K-means, pero con la opción de establecer un tamaño mínimo y/o máximo para cada grupo. Esto significa que se puede controlar cuántos elementos puede tener cada clúster, algo útil en situaciones donde se necesita un equilibrio en el tamaño de los grupos. La herramienta está basada en la implementación de KMeans de scikit-learn y utiliza una API similar con algunos ajustes (Levy-Kramer 2018, abril).