

Instituto Tecnológico de Costa Rica

Área Académica Ingeniería Mecatrónica



**Diseño del Sistema de Localización para
Vehículos Aéreos no Tripulados Desarrollados
por Intel Labs**

Informe de Proyecto de Graduación para optar por el
título de Ingeniero en Mecatrónica con el grado
académico de Licenciatura

Miguel Andrés Solís Segura

Noviembre de 2022



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

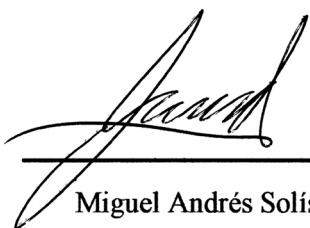
DECLARATORIA DE AUTENTICIDAD

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 25 de noviembre 2022



Miguel Andrés Solís Segura

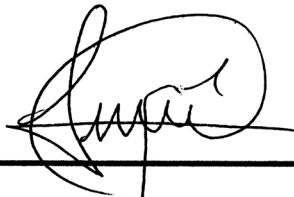
Céd: 117220717

**INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN**

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Miguel Andrés Solís Segura

Proyecto: Diseño del Sistema de Localización para Vehículos Aéreos no Tripulados Desarrollados por Intel Labs



MSc. -Ing. Felipe Meza Obando

Asesor

Cartago, 25 de noviembre 2022


INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

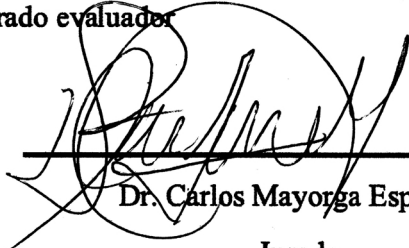
Estudiante: Miguel Andrés Solís Segura

Proyecto: Diseño del Sistema de Localización para Vehículos Aéreos no Tripulados Desarrollados por Intel Labs


Miembros del jurado evaluador



Dr. -Ing. Juan Luis Crespo Mariño
Jurado



Dr. Carlos Mayorga Espinoza
Jurado



MSc. Ing. Jaime Mora Meléndez
Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 25 de noviembre 2022

RESUMEN

En este reporte se presenta el diseño de un sistema de localización en tres dimensiones para su uso en aeronaves autónomas no tripuladas, desarrolladas por Intel Corporation. Dichos autómatas serán destinados para su uso en situaciones de inspección, búsqueda y rescate, por lo que es imprescindible contar con una ubicación precisa en todo momento. Para este propósito, se propone un sistema de bajo costo que fusiona información de carácter visual, inercial y geoespacial, proveniente de un juego de cámaras estéreo, una unidad de medición inercial, un módulo de navegación global por satélite y un barómetro. De esta manera, se logra rastrear la ubicación de la aeronave en espacios abiertos con un error medio de 28 centímetros. Dicho rendimiento supera a los sistemas comerciales en el mismo rango de precio, y se mantiene competitivo en relación al costo-desempeño respecto a sistemas comerciales más precisos y costosos.

Palabras clave: fusión de señales, odometría visual inercial, sistema de localización, vehículo aéreo no tripulado

ABSTRACT

This report presents the design of a three-dimensional location system for use in autonomous unmanned aircraft, developed by Intel Corporation. These automatons will be used in inspection, search and rescue situations, so it is essential to have a precise location at all times. For this matter, a low-cost system is proposed that fuses visual, inertial and geospatial information from a set of stereo cameras, an inertial measurement unit, a global satellite navigation module and a barometer. In this way, it is possible to track the location of the aircraft in open spaces with an average error of 28 centimeters. The system outperforms commercial products in the same price range, and remains cost-performance competitive against more accurate and expensive commercial systems.

Keywords: signal fusion, visual-inertial odometry, localization system, unmanned aerial vehicle

ÍNDICE GENERAL

1. Introducción	1
1.1. Contexto	1
1.2. Descripción del Problema	2
1.3. Síntesis del Problema	3
1.4. Objetivos	3
1.4.1. Objetivo General	3
1.4.2. Objetivos Específicos	3
1.5. Estructura del Documento	4
1.6. Aporte de Ingeniería	4
2. Marco Teórico	5
2.1. Robot Operating System	5
2.2. Gazebo	6
2.3. Odometría	6
2.4. Problema de Deriva	7
2.5. Modelo Pinhole	8
2.6. Tecnología Operacional	9
3. Metodología	10
3.1. Identificación de las Necesidades	10
3.1.1. Recopilación de Datos sin Procesar	10
3.1.2. Interpretación de Datos en Términos de Necesidades	11
3.1.3. Organización de las Necesidades en Jerarquías	12

3.1.4.	Establecimiento de la Importancia Relativa	13
3.1.5.	Reflexión en los Resultados y el Proceso	13
3.2.	Especificaciones del Sistema	15
3.2.1.	Lista de Métricas	15
3.2.2.	Comparativa con Sistemas Similares	15
3.2.3.	Establecimiento de Valores Objetivo	20
3.2.4.	Reflexión en los Resultados y el Proceso	20
3.3.	Generación de Conceptos	22
3.3.1.	Aclaración del Problema	22
3.3.2.	Búsqueda Externa	25
3.3.3.	Búsqueda Interna	32
3.3.4.	Exploración Sistemática	32
3.3.5.	Reflexión sobre las Soluciones y el Proceso	36
3.4.	Selección del Concepto	36
3.4.1.	Elaboración de Matriz de Selección	37
3.4.2.	Evaluación de Conceptos	38
3.4.3.	Ordenación de Conceptos	38
3.4.4.	Combinación y Mejora de Conceptos	40
3.4.5.	Selección de Conceptos	41
3.4.6.	Reflexión sobre los Resultados y el Proceso	41
4.	Propuesta de Diseño	42
4.1.	Diagnóstico del Sistema Actual	42
4.2.	Arquitectura para la Captación de Datos Geoespaciales	47
4.2.1.	Selección del Hardware	48
4.2.2.	Acoplamiento Mecánico	48
4.3.	Fusión de Señales	52
4.3.1.	Estimación Local	52
4.3.2.	Estimación Global	54
4.3.3.	Implementación Programada	55
4.4.	Escenario de Pruebas	60
5.	Resultados y Análisis	64
5.1.	Prueba de Concepto	64
5.1.1.	Diseño de Cribado	66
5.2.	Diseño de Superficie de Respuesta	68
5.3.	Experimentos Complementarios	73
5.3.1.	Corrección de la Deriva	73

5.3.2.	Condiciones de Poca Luz	74
5.3.3.	Navegación sin Datos Geospaciales	75
5.3.4.	Navegación con el Módulo Matek AP M9N-F4	77
5.4.	Análisis Económico	79
5.4.1.	Coste de Implementación	79
5.4.2.	Comparativa con Soluciones Comerciales	80
5.4.3.	Entorno de Simulación	82
5.5.	Especificaciones Finales	82
6.	Conclusiones y Recomendaciones	84
6.1.	Conclusiones	84
6.2.	Recomendaciones	85
	Bibliografía	87
A.	Material Complementario	95
A.1.	Incorporación del Módulo GNSS	95
A.2.	Definición de Factores de Optimización	102
A.2.1.	IMU-Cámaras	102
A.2.2.	GNSS	102
A.2.3.	Barómetro	102
A.3.	Repositorio de Documentación	104
A.4.	Repositorio de Implementación	105
B.	Documentación Adicional	107
B.1.	Plano del Cobertor para el Módulo GNSS	107
B.2.	Hoja de Datos de la Cámara de Rastreo RealSense T265	109
B.3.	Hoja de Datos del IMU BMI055	111
B.4.	Hoja de Datos del GNSS NEO-M9N-00B	113
B.5.	Hoja de Datos del Barómetro DPS310	114

ÍNDICE DE FIGURAS

2.1. Ejemplo de escenario de simulación en Gazebo [5]	6
2.2. Especificación de orientaciones [7]	7
2.3. Patrón de calibración típico [9]	8
3.1. Error relativo del algoritmo de Forster et al. [12]	17
3.2. Desempeño del algoritmo de Kerl et al. en una ruta cerrada [13]	18
3.3. Desempeño del algoritmo de Zhang y Singh [15]	19
3.4. Comparativa del algoritmo de Rebecq et al. [16]	19
3.5. Diagrama funcional de caja negra para el UAV	22
3.6. Diagrama con detalle de subfunciones del UAV	23
3.7. Ejemplo de los resultados obtenidos por Long et al. [21]	28
3.8. Comparativa de la restricción por elevación [23]	28
3.9. Error de latitud bajo la implementación de Wang et al. [29]	30
3.10. Enfoque de Qin et al. [32]	31
3.11. Vista general del paquete Intel Edge Insights	33
3.12. Concepto A	34
3.13. Concepto B	34
3.14. Concepto C	34
3.15. Concepto D	35
3.16. Concepto E	35
3.17. Concepto F	36
4.1. Modelo de aeronave Ardrone equipada con el RealSense T265	44

ÍNDICE DE FIGURAS

4.2.	Trayectorias ejecutadas en el experimento de diagnóstico	45
4.3.	Distribución del error para los experimentos de diagnóstico . .	46
4.4.	Análisis de varianza para el experimento de diagnóstico	47
4.5.	Tipos de antena para receptores GNSS	49
4.6.	Hardware y fijación mecánica para la captación de datos geo- espaciales	50
4.7.	Configuración para manufactura con Makerbot Replicator 2X	51
4.8.	Determinación de profundidad por medio de triángulos seme- jantes [51]	53
4.9.	Representación gráfica de los factores de localización	54
4.10.	Relación de los ángulos de visión	57
4.11.	Tramas de referencia	59
4.12.	Estructura base del terreno	60
4.13.	Diseño visual del terreno	61
4.14.	Modelo de pasto utilizado	62
4.15.	Corrección del renderizado de texturas	63
5.1.	Características detectadas en la imagen	65
5.2.	Factores significativos del diseño de criba	68
5.3.	Optimización para el diseño de cribado	69
5.4.	Factores significativos del diseño de superficie de respuesta . .	71
5.5.	Error real respecto a error predicho	71
5.6.	Residuales bajo la distribución t-student	72
5.7.	Superficie formada por la interacción entre el paralaje y la distancia entre características	72
5.8.	Optimización para el diseño de superficie de respuesta	73
5.9.	Comprobación del error medio predicho por el modelo con los parámetros optimizados	74
5.10.	Desempeño del sistema a largo plazo	75
5.11.	Condiciones experimentales de poca luz	76
5.12.	Desempeño del sistema en condiciones de poca luz	77
5.13.	Desempeño del sistema sin señal del módulo GNSS	78
5.14.	Comparativa entre estimación con sensores locales	78
5.15.	Desempeño del sistema empleando datos geoespaciales	79
A.1.	Identifique la zona designada para colocar el módulo GNSS. .	96
A.2.	Marque cuatro puntos sobre la superficie, formando un cua- drado de 30.5 mm de lado.	97

A.3. Realice cuatro agujeros sobre los puntos marcados, con un diámetro de 3 mm.	97
A.4. Introduzca el módulo M9N-F4 en el cobertor.	98
A.5. Asegúrese que el conector micro-USB se encuentra alineado con la abertura del cobertor.	99
A.6. Coloque el ensamble sobre la superficie de la aeronave, de forma que todos los agujeros queden alineados.	100
A.7. Finalmente, coloque los cuatro pernos de cabeza hueca en los agujeros, generando un ajuste firme con las tuercas. Conecte el módulo al procesador empleando el cable micro-USB.	101
A.8. Repositorio de material complementario	104
A.9. Repositorio de implementación programada	105
B.1. Características mecánicas [65]	109
B.2. Disposición de las cámaras [65]	109
B.3. Propiedades del sensor de la cámara [65]	110
B.4. Propiedades ópticas [65]	110
B.5. Características eléctricas del giroscopio [66]	111
B.6. Características eléctricas del acelerómetro [66]	112
B.7. Desempeño general del GNSS [67]	113
B.8. Características del barómetro [68]	114

ÍNDICE DE CUADROS

3.1. Importancia relativa de necesidades	14
3.2. Lista de métricas	16
3.3. Valores objetivo	21
3.4. Matriz de decisión para la generación del escenario de validaciones	39
3.5. Matriz de decisión para el subproblema de obtención de datos geoespaciales	39
3.6. Matriz de decisión para el subproblema de fusión de señales	40
4.1. Comparativa entre módulos GNSS	49
4.2. Masa unitaria de los elementos físicos del sistema diseñado	51
5.1. Factores para el diseño de cribado	66
5.2. Resultados del diseño de criba	67
5.3. Factores para el diseño de superficie de respuesta	70
5.4. Resultados del diseño de superficie de respuesta	70
5.5. Coste de los elementos físicos del sistema diseñado	80
5.6. Rendimiento de sistemas de localización comerciales	81
5.7. Especificaciones finales	83
A.1. Lista de partes para ensamble del módulo GNSS	96

LISTA DE ABREVIACIONES

Abreviación	Significado
CNN	Convolutional Neural Network
EKF	Extended Kalman Filter
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HRC	Human Robot Collaboration
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISR	Intelligent System Research
KF	Kalman Filter
LSTM	Long Short Term Memory
OT	Operational Tecnology
RNN	Recurrent Neural Network
ROS	Robot Operating System
RS	Remote Sensing
RTK	Real Time Kinematics
SD	Sistema Diseñado
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VIO	Visual Inertial Odometry

CAPÍTULO 1

INTRODUCCIÓN

1.1. Contexto

El proyecto será realizado en conjunto con la empresa Intel Corporation, una de las compañías líder en el sector de manufactura procesadores [1]. En las últimas dos décadas, la empresa ha apostado por incursionar en tecnologías emergentes fuera del sector tradicional que ha caracterizado sus actividades de negocio. En este sentido, la organización Intel Labs es uno de los pilares fundamentales dentro de la compañía dónde se promueve la investigación y desarrollo de nuevas tecnologías [2]. Dentro de esta organización, existen múltiples líneas de trabajo, abarcando desde computación cuántica hasta el desarrollo de técnicas innovadoras en ciberseguridad.

Concretamente, el proyecto será desarrollado con el grupo de Intelligent Systems Research (ISR), enfocado en desarrollar tecnologías de cooperación entre humanos y máquinas con el propósito de amplificar el potencial humano empleando señales multimodales. Este grupo realiza investigación fundamental y aplicada en los sectores de manufactura, educación, industria y computación de asistencia [3].

ISR se subdivide en departamentos que trabajan según las características de los proyectos que se asignan al grupo. El departamento de Human Robot Collaboration (HRC) trabaja, entre otros proyectos, en el desarrollo de vehículos aéreos no tripulados con el objetivo de despliegue en situaciones de emergencia. El estado actual de dichos vehículos permite la navegación

y búsqueda de objetivos, al mismo tiempo que genera un mapa del entorno en el que opera. No obstante, el sistema de localización requiere refinamiento y aún no se han diseñado pruebas físicas o de simulación que modelen situaciones de riesgo en entornos naturales.

1.2. Descripción del Problema

Los algoritmos de control para vehículos aéreos desarrollados por HRC tienen como objetivo final ser desplegados en entornos de riesgo con presencia humana, por lo que plantean el reto de enfrentarse a escenarios dinámicos y suponen un potencial peligro para las personas si no han sido evaluados diligentemente. El ciclo de desarrollo empleado por HRC consiste en dos etapas:

1. Diseño de bajo costo de un concepto, mediante pruebas exhaustivas de simulación. De esta manera se evalúa fácilmente el desempeño y se descarta la presencia de anomalías insospechadas en el funcionamiento que puedan manifestarse posteriormente.
2. Implementación física y pruebas de campo. Así se validan los resultados obtenidos en el proceso de simulación, y si es necesario se realizan refinamientos menores al concepto implementado.

Este proceso es iterativo, y se repite conforme se detecta la necesidad de agregar nuevas funcionalidades a la aeronave. Durante previas iteraciones, se detectaron puntos de mejora en el sistema de localización. Actualmente este se basa en inferencias realizadas con datos de una unidad de medición inercial (IMU) incorporada a la aeronave, pero no se cuenta con mediciones directas de carácter geoespacial para estimar la posición. Esto ocasiona que los errores en la estimación se vuelvan acumulativos, y al no existir mediciones libres de deriva para complementar las lecturas del IMU, el desempeño de la estimación decae paulatinamente.

Resolver esta problemática da origen al proyecto, cuyo propósito es rediseñar el sistema de localización de la aeronave autónoma, limitándose a la primera etapa del ciclo iterativo antes mencionado.

Adicionalmente, HRC posee un amplio interés en que las simulaciones se asemejen cada vez más a situaciones de rescate y riesgo reales, ya que esta es una de las principales aplicaciones que tendrán los vehículos autónomos una

vez que concluya la fase de investigación y desarrollo. Hasta el momento, las pruebas de simulación han consistido en entornos urbanos sencillos; por lo que un segundo eje de acción para el proyecto es el diseño de un escenario de pruebas dinámico, basado en un entorno natural. Este cumplirá el doble propósito de validar la propuesta de diseño y soportar futuros ciclos de iteración en el desarrollo del sistema.

La principal ventaja de implementar un diseño de robótica móvil en una simulación robusta responde a criterios económicos. Por un lado, es evidente la facilidad con la que se puede probar conceptos sin incurrir en gastos por la compra de equipo real. En segunda instancia, el tiempo de construcción física es significativamente menor, ya que desde la simulación se determinan los requisitos para integrar las mejoras, por lo que la implementación es más ágil y menos propensa a errores.

1.3. Síntesis del Problema

Identificar y ejecutar mejoras en el sistema de localización, para el control automático de vehículos aéreos no tripulados, desarrollados por el departamento de Human Robot Collaboration en la organización Intel Labs.

1.4. Objetivos

1.4.1. Objetivo General

- Diseñar un sistema de localización avanzada para vehículos aéreos no tripulados desplegados en situaciones de riesgo, desarrollados por Intel Labs.

1.4.2. Objetivos Específicos

- Diagnosticar el desempeño del sistema de localización actualmente operante.
- Establecer la arquitectura para la recolección autónoma de datos geoespaciales por parte de la aeronave.
- Confeccionar un sistema basado en software para el procesamiento de señales y estimación de la posición del vehículo aéreo.

- Verificar el desempeño del sistema de localización diseñado, mediante la generación de pruebas dinámicas de simulación robótica.

1.5. Estructura del Documento

A continuación se realiza una descripción breve de los capítulos que preceden la introducción. En el Capítulo 2 se definen aquellos conceptos teóricos indispensables, para que el lector comprenda el contenido del informe. En el Capítulo 3 se discute la metodología de trabajo seleccionada, abarcando sus etapas desde la identificación de las necesidades del proyecto, hasta la selección de conceptos de solución. En el Capítulo 4 se detalla el proceso de implementación de los conceptos de solución elegidos, evidenciando el seguimiento de la metodología propuesta. En el Capítulo 5 se muestran las diferentes estrategias de verificación que fueron aplicadas al diseño propuesto, así como el análisis económico referente al contexto del proyecto. Se concluye con las especificaciones finales logradas para el sistema. Finalmente en el Capítulo 6 se mencionan las conclusiones y recomendaciones pertinentes. El Apéndice A contiene material complementario de diversa índole. El Anexo B incluye documentos adicionales que fueron elaborados o consultados durante la ejecución del proyecto.

1.6. Aporte de Ingeniería

El principal aporte de ingeniería presentado en el informe, constituye el diseño de un sistema de localización de bajo costo en tres dimensiones, que emplea el procesamiento de datos inerciales, geoespaciales e imágenes para una estimación precisa de la posición de un vehículo autónomo.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Robot Operating System

Dentro de cualquier aplicación de robótica debe existir un marco de referencia para la adquisición y el procesamiento de señales. Con el objetivo de facilitar el manejo de distintos subsistemas, mejorar la escalabilidad y propiciar la colaboración; el laboratorio de investigación Stanford Artificial Intelligence Laboratory publica la primera versión de Robot Operating System (ROS) en el año 2007 [4].

ROS es un proyecto de código abierto, que proporciona una capa de abstracción de software para crear aplicaciones de robótica sin preocuparse por las particularidades del hardware subyacente. El núcleo de ROS es un marco de middleware para la transmisión de mensajes, síncronos o asíncronos, donde los procesos y los subprocesos pueden comunicarse y transportar datos entre sí, incluso cuando se ejecutan desde diferentes máquinas. Otra ventaja, es que ROS está organizado por paquetes, lo que ofrece una buena modularidad y reutilización, que se puede integrar y utilizar para cualquier aplicación robótica personalizada, con cambios mínimos.

ROS ha sido acogido y utilizado ampliamente en la comunidad de robótica, en aplicaciones que van desde la investigación, hasta en el uso de robots comerciales. Su versatilidad, extensa documentación y el hecho no ser privativo, lo han convertido en la opción predilecta de muchas organizaciones académicas e industriales.

2.2. Gazebo

Gazebo es una plataforma de simulación robótica. Destaca por su capacidad de simulación de físicas que proporciona un nivel de precisión que supera a los motores de juegos [5].

Es posible añadir una gran variedad de sensores, tales como cámaras monoculares, cámaras de profundidad, sistemas lidar, medidores inerciales, sensores de contacto, altímetros y magnetómetros. Cada sensor puede utilizar opcionalmente un modelo de distorsión para incorporar propiedades de ruido gaussiano o personalizado.

En cuanto al rendimiento, esta plataforma admite el uso de varios servidores para mejorar el desempeño computacional. Además utiliza información espacial, para cargar y descargar automáticamente elementos de simulación, lo que mejora drásticamente el rendimiento. Un ejemplo de escenario de simulación se encuentra en la Figura 2.1.



Figura 2.1: Ejemplo de escenario de simulación en Gazebo [5]

2.3. Odometría

La odometría es el estudio de datos provenientes de sensores para estimar el cambio de posición a lo largo del tiempo [6]. El término se acuña de las

palabras griegas *odos* (ruta) y *metron* (medir). Las aplicaciones en robótica se suelen utilizar para estimar la posición en relación con una ubicación inicial. En la mayoría de los casos, se requiere una recopilación de datos, calibración de instrumentos y procesamiento rápidos y precisos para que la odometría se utilice de manera efectiva en tiempo real.

La posición se suele indicar mediante tres coordenadas, típicamente x , y , y z , mientras que la orientación se acota empleando rotaciones en los tres ejes cartesianos, denotados por *roll*, *pitch* y *yaw* respectivamente. Una representación gráfica de las orientaciones se muestra en la Figura 2.2.

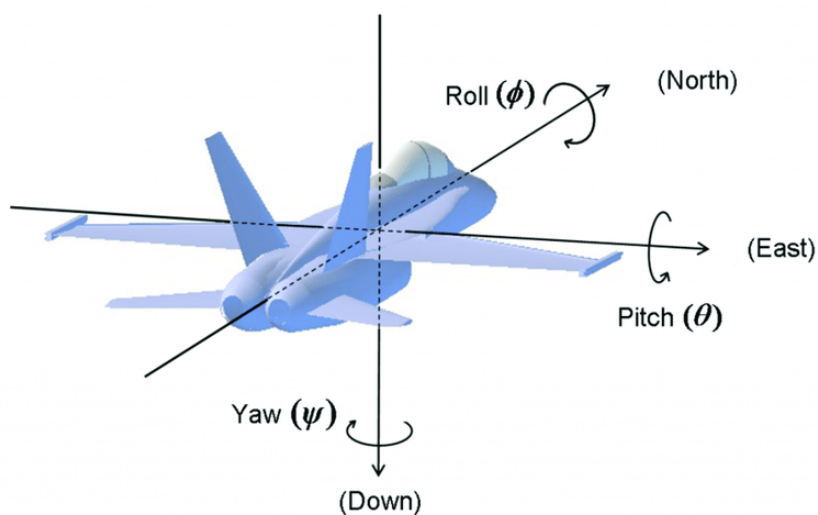


Figura 2.2: Especificación de orientaciones [7]

2.4. Problema de Deriva

El problema de deriva es un fenómeno ampliamente estudiado en la robótica móvil [8]. Es característico de sensores con lecturas locales y sucede por la acumulación de errores en estimaciones de la posición y orientación de un cuerpo. La deriva en un sistema dinámico ocurre cuando el sistema no puede detenerse instantáneamente. Por ejemplo, los sistemas dinámicos de segundo orden con aceleración limitada no pueden cancelar instantáneamente una

velocidad distinta de cero.

Este fenómeno es prácticamente inevitable cuando se emplean únicamente sensores locales para estimar la posición, cómo lo son codificadores rotatorios, unidades de medición inercial y escáneres láser. Si bien se han propuesto múltiples alternativas para contrarrestar sus efectos, la mejor forma de obtener estimaciones libres de deriva es mediante la adición de sensores globalmente referenciados.

2.5. Modelo Pinhole

Algunos sistemas de odometría visual requieren reconstruir un modelo preciso de las cámaras empleadas para ser viables. Normalmente se emplean parámetros intrínsecos a la captación de imágenes para definir estos modelos [9]. Una descripción típica según el modelo pinhole de la distorsión inducida por el lente de la cámara, involucra la determinación de la distorsión radial, denotada por los coeficientes $K1$, $K2$ y $K3$, y la distorsión tangencial, acotada por los coeficientes $P1$ y $P2$. Estos coeficientes pueden ser estimados directamente por algoritmos de calibración, empleando el análisis de imágenes con patrones conocidos, cómo el mostrado en la Figura 2.3.

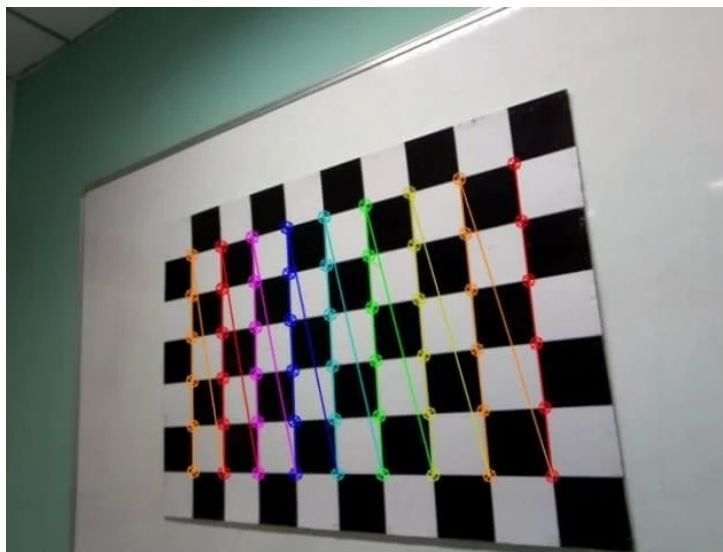


Figura 2.3: Patrón de calibración típico [9]

Adicionalmente, es necesario estimar el centro óptico de la imagen (cx ,

cy) y la distancia focal (fx , fy). Ambos se acotan en píxeles u otra unidad de distancia y se combinan en lo que se denomina la matriz de la cámara, según se indica en la Ecuación 2.1. Esta matriz es única y específica a cada cámara, por lo que una vez que se ha determinado, se puede reutilizar en otras aplicaciones de visión. Para cámaras que tengan píxeles perfectamente cuadrados, el valor de fx y fy será el mismo. En un modelo perfecto de cámara pinhole, las coordenadas cx y cy apuntan directamente al centro de la imagen.

$$\text{matriz de cámara} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

2.6. Tecnología Operacional

La Tecnología Operacional (OT) se refiere al conjunto de técnicas que abarcan hardware y software que detecta o provoca un cambio a través del monitoreo y control directo de dispositivos físicos, procesos y eventos [10].

Una evolución importante en la OT fue la capacidad de monitorear y controlar dispositivos físicos de forma remota. La influencia de tecnologías de la información, como Big Data y aprendizaje automático en OT, junto con las mejoras en la comunicación máquina a máquina y el internet de las cosas, permitieron amplias innovaciones con respecto a la gestión dispositivos físicos en procesos industriales, así como su diagnóstico y mantenimiento.

La OT se menciona cada vez más desde una perspectiva de ciberseguridad. Con infraestructuras industriales cada vez más complejas y la digitalización, se requiere atención adicional para la seguridad en sistemas de OT. Existen una diferencia crucial en términos de seguridad entre OT y tecnologías de la información, ya que el primero se encarga de proteger los procesos físicos, la seguridad y la integridad de las personas, mientras que el segundo se orienta proteger cómo se almacenan, transmiten y utilizan los datos en los procesos industriales.

Ejemplos de OT incluyen sistemas de control distribuido, controladores lógicos programables, unidades de terminales remotas, las interfaces humano máquina, los sistemas embebidos, la maquinaria, los equipos físicos en las plantas y variadores de frecuencia.

CAPÍTULO 3

METODOLOGÍA

En el Capítulo 1 se anticipó el uso de una metodología de diseño estructurada para realizar el proyecto. En concreto, se seleccionó la metodología descrita en el libro *Diseño y Desarrollo de Productos*, propuesta por Ulrich y Eppinger [11]. Los siguientes apartados demuestran el ejercicio de dicha metodología; desde la identificación de necesidades, hasta la selección de conceptos de solución.

3.1. Identificación de las Necesidades

La importancia de este apartado en la metodología de diseño dentro del contexto del proyecto, radica en asegurarse que la solución propuesta se enfoque en las necesidades del cliente, proporcione un punto de partida para delimitar las especificaciones del sistema y evite la omisión de algún requerimiento importante.

3.1.1. Recopilación de Datos sin Procesar

Debido a las restricciones geográficas, los requerimientos del proyecto fueron discutidos mediante una serie de videoconferencias con el asesor industrial, el ingeniero Rafael de la Guardia, que también ejerce el rol de cliente. El mecanismo para una documentación fidedigna de las interacciones fue la

grabación de audio. Posteriormente, los enunciados del asesor fueron procesados para su correcta formulación en términos de necesidades. Rafael es el investigador principal a cargo del desarrollo de las aeronaves, por lo que sus observaciones y comentarios gozan de una elevada comprensión de los posibles puntos de mejora y requerimientos generales del proyecto. Su estatus es el de usuario experto, según las clasificaciones que propone la metodología.

3.1.2. Interpretación de Datos en Términos de Necesidades

Para la interpretación de los datos crudos, se procuró seguir los siguientes lineamientos:

- Expresión en términos de funcionalidad, no de solución.
- Alta especificidad acorde al enunciado original de cliente.
- Enunciados en forma afirmativa.
- Expresión cómo atributos del producto.
- Evasión de los términos *debe* y *debería*.

Siguiendo estas normas, las necesidades identificadas para el sistema diseñado (SD), se indican en la siguiente lista:

- El SD recolecta información geoespacial de la ubicación de la aeronave.
- El SD mejora la exactitud actual del sistema de localización.
- La implementación física del SD tiene un costo inferior a 300 USD.
- El SD incluye un escenario de verificación en Gazebo.
- El escenario de verificación del SD corre fluidamente en el equipo de cómputo disponible.
- El SD recopila datos geoespaciales a una frecuencia igual o superior a 1 Hz.
- El SD es compatible con Intel Edge Insights.

- El escenario de verificación del SD asemeja una situación de riesgo por incendio.
- El SD integra los datos de la unidad de medición inercial para estimar la posición.
- El SD es compatible con el procesador Intel Atom x7-Z8750.
- El escenario de verificación del SD se diseña en base a un entorno natural.

3.1.3. Organización de las Necesidades en Jerarquías

Desde el apartado anterior se eliminaron las frases redundantes y necesidades muy similares entre sí. De esta manera se organizan con mayor facilidad las necesidades de la lista anterior en grupos de necesidades primarias y secundarias. La jerarquización sigue:

1. El SD mejora la exactitud actual del sistema de localización.
 - El SD recolecta información geoespacial de la ubicación de la aeronave.
 - El SD recopila datos geoespaciales a una frecuencia igual o superior a 1 Hz.
 - El SD integra los datos de la unidad de medición inercial para estimar la posición.
2. El SD incluye un escenario de verificación en Gazebo.
 - El escenario de verificación del SD corre fluidamente en el equipo de cómputo disponible.
 - El escenario de verificación del SD se diseña en base a un entorno natural.
 - El escenario de verificación del SD asemeja una situación de riesgo por incendio.
3. El SD es compatible con el procesador Intel Atom x7-Z8750.
 - El SD es compatible con Intel Edge Insights.
4. La implementación física del SD tiene un costo inferior a 300 USD.

3.1.4. Establecimiento de la Importancia Relativa

Tras la jerarquización de las necesidades, se envió una encuesta al asesor industrial para ponderar la importancia de las mismas. Para dicho fin se empleó una escala lineal para catalogar cada necesidad desde el uno (1) al cinco (5) dónde:

- (1): es estrictamente indispensable que el sistema diseñado satisfaga la necesidad tal cómo es planteada.
- (2): es necesario satisfacer la necesidad, pero se admite flexibilidad para aprobar su cumplimiento.
- (3): satisfacer la necesidad es altamente deseable, pero se admite su incumplimiento parcial de ser requerido.
- (4): satisfacer la necesidad es deseable, pero se admite su incumplimiento total de ser requerido.
- (5): se puede prescindir por completo de satisfacer la necesidad bajo cualquier circunstancia.

En base a esta escala, los resultados de la ponderación se resumen en el Cuadro 3.1.

3.1.5. Reflexión en los Resultados y el Proceso

Siguiendo la metodología de diseño, se han identificado las principales necesidades del proyecto, así cómo su importancia relativa. Adicionalmente a la ponderación de las necesidades, se solicitó al asesor industrial indicar cualquier necesidad adicional que no hubiese sido contemplada en los apartados anteriores; pero no se añadió ningún otro requerimiento. De esta manera se concluye que, si bien la lista de necesidades no es particularmente extensa, sí es suficiente bajo el contexto del proyecto y las expectativas que tiene el asesor industrial del mismo. Se tiene la certeza de que cualquier otro requerimiento que surja durante el desarrollo del proyecto será referente a particularidades de la ejecución; el enfoque y alcance general han sido descritos satisfactoriamente y permanecerán inmutables.

Cuadro 3.1: Importancia relativa de necesidades

ID	Necesidad	Importancia
1	El SD mejora la exactitud actual del sistema de localización.	1
2	El SD recolecta información geoespacial de la ubicación de la aeronave.	1
3	El SD recopila datos geoespaciales a una frecuencia igual o superior a 1 Hz.	1
4	El SD integra los datos de la unidad de medición inercial para estimar la posición.	1
5	El SD incluye un escenario de verificación en Gazebo.	1
6	El escenario de verificación del SD corre fluidamente en el equipo de cómputo disponible.	1
7	El escenario de verificación del SD asemeja una situación de riesgo por incendio.	2
8	El escenario de verificación del SD se diseña en base a un entorno natural.	2
9	El SD es compatible con el procesador Intel Atom x7-Z8750.	1
10	El SD es compatible con Intel Edge Insights.	1
11	La implementación física del SD tiene un costo inferior a 300 USD.	3

3.2. Especificaciones del Sistema

Si bien las necesidades establecen una dirección clara para el proyecto; por sí solas dejan un amplio margen de interpretación subjetiva para el diseño del sistema. La importancia del presente apartado radica en establecer especificaciones precisas y medibles de lo que debe cumplir el sistema diseñado. De esta forma se establecen criterios claros para determinar el grado de satisfacción en las necesidades.

3.2.1. Lista de Métricas

Las métricas deben indicar de la forma más directa posible, el nivel al cual el sistema diseñado satisface las necesidades planteadas. Todas las necesidades deben ser mapeadas a una o más métricas precisas y mensurables. En el Cuadro 3.2 se evidencia la propuesta de métricas para el diseño. Cabe resaltar que la columna *ID* relaciona cada métrica con su respectiva necesidad del Cuadro 3.1. Todas las métricas se han seleccionado para su comprobación por medio alguna de las siguientes vías:

- Especificaciones de los componentes utilizados.
- Inspección visual.
- Experimentación.

3.2.2. Comparativa con Sistemas Similares

Las métricas por sí mismas sólo establecen unidades de medición. Para verificar el cumplimiento de las necesidades, se requiere establecer valores objetivo. Si bien algunas necesidades responden a los intereses particulares del cliente, o incluso ya sugieren un punto de partida para los valores objetivo, otras están sujetas a comparación con el estado del arte en cuanto a sistemas de localización para vehículos autónomos. Este apartado contiene una revisión de trabajos de investigación referentes a sistemas de localización, que tiene por objetivo establecer un punto de comparación para desarrollar un sistema competitivo.

Empleando técnicas de odometría visual, Forster et al. [12] presentan un algoritmo que opera sobre las intensidades de los píxeles. Su enfoque elimina

Cuadro 3.2: Lista de métricas

ID	Métrica	Unidades	Importancia
1	Error medio de localización.	m	1
2	Información geoespacial recolectada.	lista	1
3	Frecuencia de actualización.	Hz	1
4	Porcentaje de integración con IMU.	%	1
5	Disponibilidad de verificación en Gazebo.	binaria	1
6	Factor de tiempo real.	%	1
7	Semejanza a situación de riesgo por incendio.	lista	2
8	Semejanza a entorno natural.	lista	2
9	Grado de compatibilidad con el procesador Intel Atom.	%	1
10	Grado de compatibilidad con Intel EI.	%	1
11	Costo económico de implementación.	USD	3

la necesidad de extracción de características en la imagen, lo cual resulta beneficioso en términos de costo computacional. El algoritmo tiene como propósito permitir navegar a vehículos aéreos autónomos en circunstancias dónde no es posible contar con datos geoespaciales. Los investigadores lograron una tasa de actualización de 50 Hz, y reportan el error en la estimación en términos relativos para cada eje cartesiano: 2.5 cm para el eje x , 5 cm para el eje y , 5 cm para el eje z . Los resultados se pueden observar en la Figura 3.1, dónde la línea azul constituye los resultados más precisos logrados.

La principal limitante encontrada en el enfoque anterior, radica en la restricción para la captación de imágenes. La cámara debe estar apuntando en todo momento hacia el suelo para rastrear la posición, y los resultados reportados se obtuvieron en experimentos dónde la altura promedio de vuelo fue de 1.2 metros. Usar este enfoque no sería idóneo para el tipo de aeronaves que diseña HRC, pero los resultados de precisión son notables y serán considerados.

En su artículo, Kerl et al. [13] exponen un método de localización y mapeo simultáneo para aplicaciones de robótica, basado en cámaras de color y profundidad. El método fue evaluado empleando conjuntos de datos de referencia disponibles públicamente y se reporta un buen desempeño en escenas con pocas texturas y estructuras. En la Figura 3.2 se muestra una

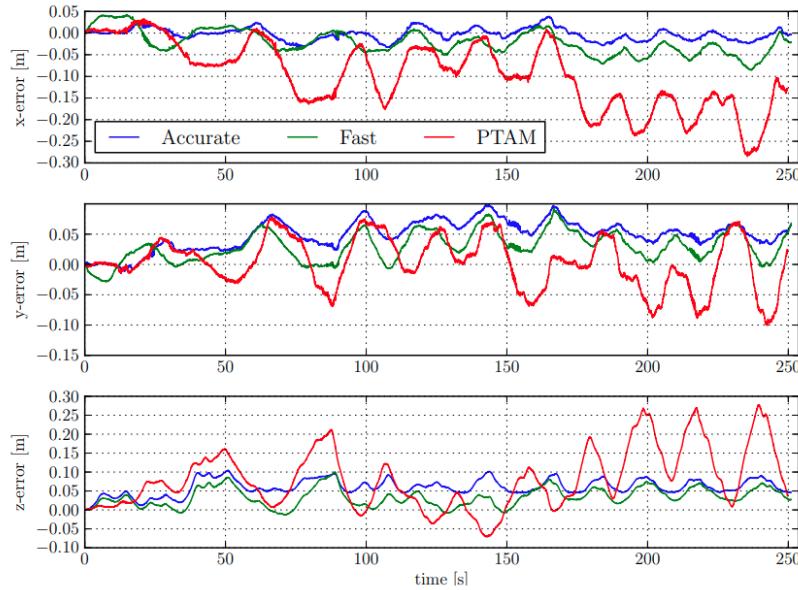


Figura 3.1: Error relativo del algoritmo de Forster et al. [12]

comparativa de dicho desempeño.

Los autores logran una reducción del error de deriva del 16 % con el método propuesto, y una reducción del error global de estimación en la trayectoria del 170 % frente a métodos convencionales. Como punto negativo, los datos utilizados para la validación son de trayectorias muy cercanas a objetos, algo necesario para poder emplear los datos del sensor de profundidad. Lo anterior sería un impedimento para el proyecto, ya que el uso de este tipo de sensores tendría poca utilidad en espacio abiertos. Aun así, la implementación se debe considerar ya que las aeronaves de HRC también disponen de cámaras estéreo y un sensor de profundidad [14].

Zhang y Singh [15] proponen un método de odometría y mapeo en tiempo real usando mediciones de rango de un lidar de 2 ejes. El método logra una baja deriva y una baja complejidad computacional sin la necesidad de medidas inerciales o de rango de alta precisión. Esto se logra al dividir el problema en dos algoritmos. Uno realiza la estimación de odometría a alta frecuencia pero baja fidelidad para estimar la velocidad del lidar. El otro algoritmo se ejecuta a una frecuencia de un orden de magnitud inferior para emparejamiento fino. La Figura 3.3 muestra una comparación del algoritmo

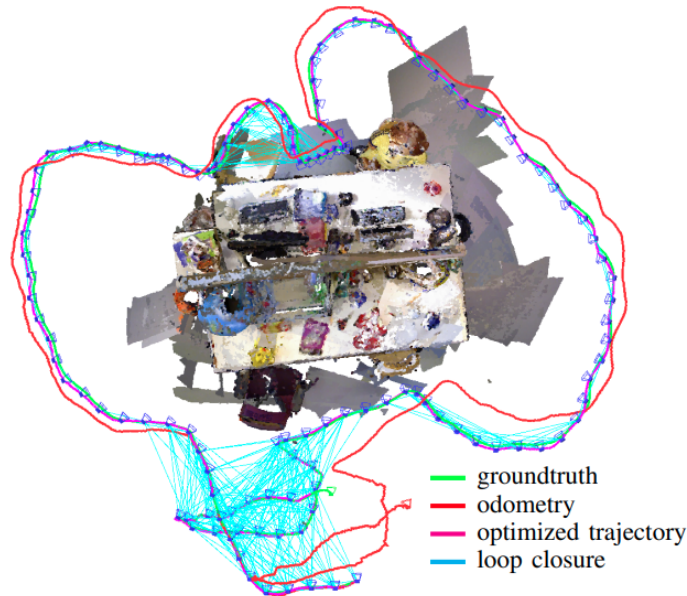


Figura 3.2: Desempeño del algoritmo de Kerl et al. en una ruta cerrada [13]

propuesto, con y sin asistencia de un dispositivo IMU, durante el recorrido por unas escaleras.

El error medio reportado con asistencia del IMU, oscila entre 0.9 % para un recorrido de 32 metros y 2.6 % para uno de 43 metros. Si bien este enfoque de localización no fue diseñado para aeronaves, tiene en común con el proyecto el uso de múltiples sensores para lograr una localización robusta, y particularmente el involucramiento de un IMU. No obstante, las distancias recorridas en la evaluación no son especialmente significativas para aplicaciones de búsqueda y rescate, que competen al presente proyecto.

Finalmente, se considera la implementación de Rebecq et al. [16]. Los investigadores desarrollaron un algoritmo de odometría visual basado en eventos. Este aprovecha con éxito las propiedades de las cámaras de eventos para rastrear los movimientos rápidos de la cámara mientras genera un mapa tridimensional del entorno. El algoritmo funciona bien en condiciones desafiantes de alto rango dinámico con fuertes cambios de iluminación. La visualización comparativa del desempeño se muestra en la Figura 3.4, donde la trayectoria real se denota por la línea roja y la estimada en azul.

Usando el método de odometría por eventos, se logró un error inferior al 0.2 % para la posición y menor a 3 grados para la orientación. De todos los

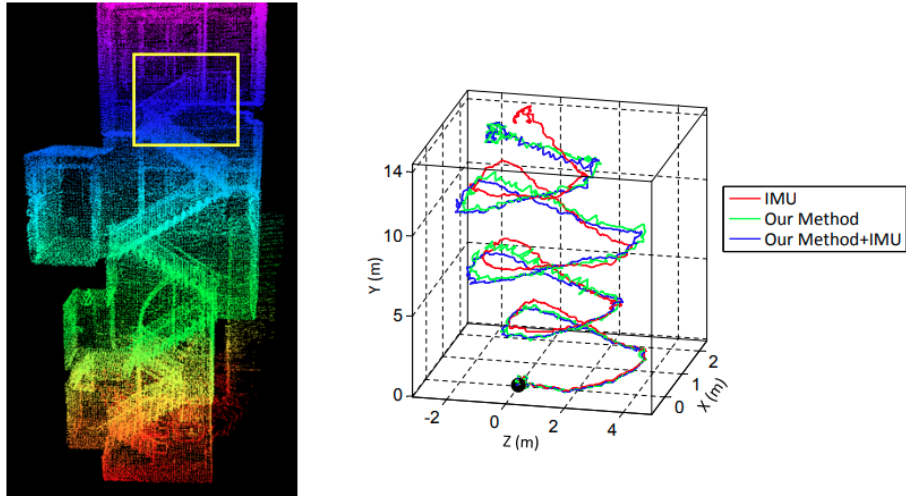


Figura 3.3: Desempeño del algoritmo de Zhang y Singh [15]

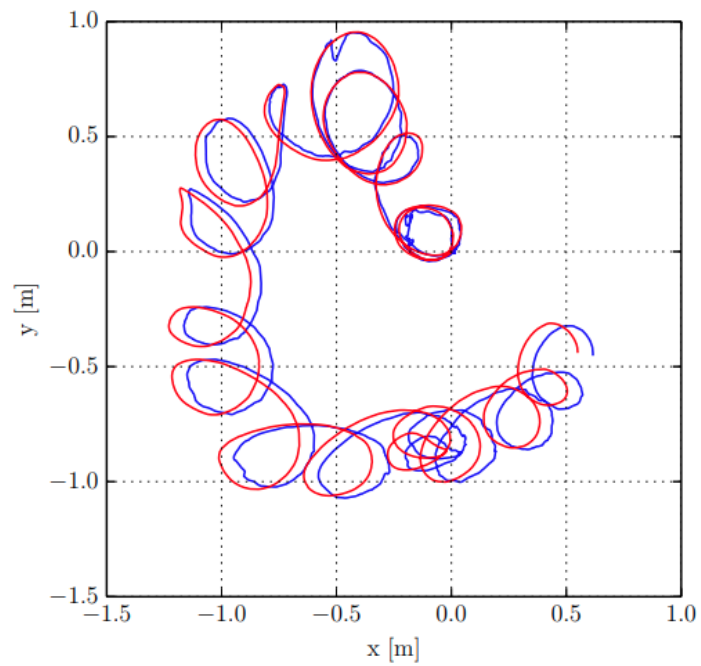


Figura 3.4: Comparativa del algoritmo de Rebecq et al. [16]

trabajos consultados, este reporta uno de los mejores desempeños, y sobresale la robustez con la que el sistema se repone ante cambios bruscos en la posición. El principal problema radica en que los datos de validación fueron tomados en espacios muy reducidos, por lo que no se tiene información sobre cuál sería el desempeño abarcando grandes extensiones de terreno.

3.2.3. Establecimiento de Valores Objetivo

En base a la comparación con sistemas de localización y las propias expectativas del cliente, se establecieron los valores objetivo para las métricas. Se emplean dos categorías para este propósito:

1. Un valor ideal, que constituye el mejor resultado que se aspira alcanzar durante el desarrollo del proyecto.
2. Un valor marginalmente aceptable, que constituye el cumplimiento mínimo de la necesidad asociada.

Los valores objetivo y marginalmente aceptables para el sistema diseñado se acotan en el Cuadro 3.3. Para la selección de los valores, también se tomó en consideración la importancia asignada por el cliente a cada necesidad.

3.2.4. Reflexión en los Resultados y el Proceso

Los valores objetivo no se han establecido bajo expectativas poco razonables. El error medio de localización se ha situado dentro de un rango conservador en comparación a otros trabajos consultados, ya que los mismos se validaron a baja altitud o en un rango de movimiento reducido, lo que implica condiciones más favorables para una alta exactitud de estimación.

Actualmente no se contempla la posibilidad de diseñar más de una versión del sistema de localización. Si bien las condiciones de operación y usos específicos del producto final pueden ser muy distintos entre sí, el cliente ha manifestado el deseo de obtener un sistema capaz de alta generalización; no varios sistemas adaptados para condiciones específicas. Esto se alinea con el concepto de navegación autónoma, dónde se espera que el control pueda tomar decisiones inteligentes según los cambios que ocurren en el medio.

De esta manera, las expectativas que se han propuesto con los valores objetivo, apuntan a la creación de un sistema de localización con un buen

Cuadro 3.3: Valores objetivo

ID	Métrica	Unidades	Valor Marginal	Valor Ideal
1	Error medio de localización.	m	<0.3	<0.1
2	Información geoespacial recolectada.	lista	latitud, longitud	latitud, longitud, altitud
3	Frecuencia de actualización.	Hz	>1	>5
4	Porcentaje de integración con IMU.	%	100	100
5	Disponibilidad de verificación en Gazebo.	binaria	disponible	disponible
6	Factor de tiempo real.	%	>96	>99
7	Semejanza a situación de riesgo por incendio.	lista	árboles quemados, marcas del terreno	árboles quemados, marcas del terreno, torres de alta tensión
8	Semejanza a entorno natural.	lista	árboles, césped, rocas	árboles, césped, rocas, terreno irregular
9	Grado de compatibilidad con el procesador Intel Atom.	%	100	100
10	Grado de compatibilidad con Intel EI.	%	>95	100
11	Costo económico de implementación.	USD	<300	<350

desempeño en espacios abiertos, capaz de fusionar señales de distinta naturaleza provenientes de sensores de bajo costo. Si bien su viabilidad se evaluará mediante simulación; el entorno que se pretende crear para dicho fin gozará de características que asemejan en gran medida a la realidad.

3.3. Generación de Conceptos

A partir de las necesidades detectadas y las especificaciones objetivo, se procede a la etapa de generación de conceptos. Durante este apartado se realiza una descripción aproximada de la tecnología con la que el sistema diseñado logra satisfacer las necesidades propuestas. Esta etapa es crucial para garantizar que todo el espacio de alternativas ha sido explorado.

3.3.1. Aclaración del Problema

Es menester realizar una comprensión del problema de fondo a resolver, y si es necesario subdividirlo en problemas más simples. La definición general del problema ya fue tratada en el Capítulo 1. Ahora, en conjunto con las necesidades y las especificaciones propuestas, se procede analizar el problema para su correcta descomposición.

Considerando que el sistema diseñado es a su vez un subsistema que coexiste con otros dentro del vehículo aéreo no tripulado (UAV), se comenzará por realizar una descripción general de caja negra desde la perspectiva más externa posible. La Figura 3.5 describe esta descomposición funcional.

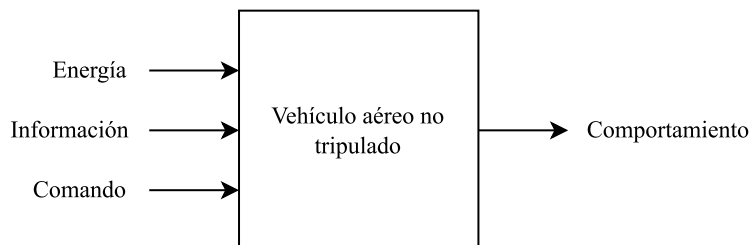


Figura 3.5: Diagrama funcional de caja negra para el UAV

Los términos que aparecen en la Figura 3.5 se refieren a:

- Energía: entrada de alimentación para todos los subsistemas del UAV. Actualmente se cuenta con una batería recargable que suministra a la

aeronave con energía eléctrica. Si bien no existen limitaciones teóricas para manipular la fuente de energía existente o incorporar nuevos mecanismos para suministrar energía; en la práctica esto supondría complicaciones adicionales ya que la masa añadida o la forma de la aeronave se verían afectados.

- Información: todos los datos acerca del entorno que permiten a la aeronave tomar decisiones. Evidentemente, el tipo de información que puede percibir el UAV es función de los distintos sensores que incorpore. Es un objetivo primordial del proyecto, ampliar esta entrada para mejorar las capacidades de localización de la aeronave.
- Comando: la orden de trabajo dada a la aeronave. La operación autónoma del UAV responde a las intenciones de un operario humano, que se encuentra en capacidad de ordenar un proceso de búsqueda de objetivos, el mapeo de un área extensa desconocida o el retorno a un punto de partida inicial.

Seguidamente, se debe añadir un nivel extra de detalle, para delimitar los subproblemas que se deben resolver. En la Figura 3.6, con líneas punteadas se distinguen los flujos y subfunciones sobre los que se tiene injerencia. Por motivos de legibilidad, se han realizado algunas simplificaciones que se discuten a continuación.

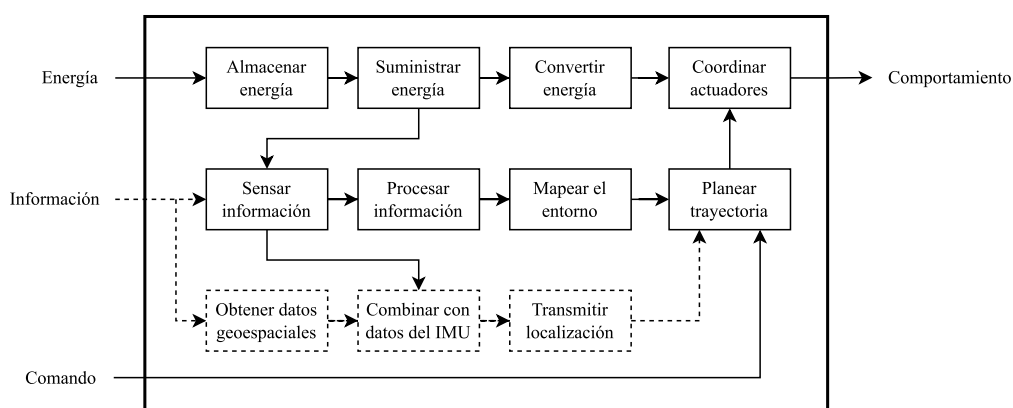


Figura 3.6: Diagrama con detalle de subfunciones del UAV

Entre las subfunciones ya implementadas de la aeronave, se identifican:

- Almacenar energía: es el rol que cumple actualmente la batería, convirtiendo energía eléctrica en química para su almacenamiento.
- Distribuir energía: es la función brindada por el procesador de la aeronave, que se alimenta con la energía de la batería y la despacha a los distintos sensores y actuadores de la aeronave, ya sea de forma directa o a través de una interfaz de potencia.
- Convertir energía: destacando principalmente el caso de los motores de las hélices, la función es provista por los actuadores incorporados a la aeronave.
- Sensar información: es la traducción de magnitudes físicas a señales eléctricas con las que puede trabajar el procesador. De esta función se encargan los múltiples sensores incorporados a la aeronave.
- Procesar información: comprende los diferentes algoritmos de procesamiento y análisis de datos efectuados para obtener una comprensión del entorno inmediato.
- Mapear el entorno: es la reconstrucción tridimensional de obstáculos que rodean a la aeronave. Es relevante tanto la forma como la posición de entidades dinámicas y estáticas.
- Planear trayectoria: comprende el post-procesamiento de datos para identificar rutas de movimiento seguras para la aeronave.
- Coordinar actuadores: es la ejecución de una trayectoria, en base a las capacidades inherentes de los actuadores y la ruta provista por los algoritmos de planeamiento.

Además de los elementos anteriormente mencionados, para la satisfacción de las necesidades se debe implementar:

- Obtener datos geoespaciales: proveer a la aeronave con los insumos necesarios para sensar su ubicación en el espacio. Técnicamente, esta función podría ser englobada en la de sensar información, pero se ha hecho la distinción por la obvia razón de que es un problema aún por resolver. Adicionalmente, el suministro de energía también es una entrada para esta función pero se ha omitido por legibilidad.

- Combinar datos con el IMU: si bien no es posible modificar la manera en la que se recopilan los datos de la unidad de medición inercial, si es factible alterar el procesamiento de los mismos. En conjunto con los datos geoespaciales, se aspira a mejorar la exactitud de la estimación de la pose de la aeronave.
- Transmitir de datos: se refiere a la incorporación de la pose estimada por la subfunción anterior, al resto de sistemas de la aeronave para un adecuado planeamiento de la trayectoria. Particularmente con esta subfunción se espera satisfacer la necesidad asociada a la compatibilidad con Intel Edge Insights.

Una vez que se identifican los subproblemas a tratar, se procede a recopilar información que sirva de guía para la resolución de los mismos.

3.3.2. Búsqueda Externa

La búsqueda externa tiene como fin encontrar soluciones a los subproblemas identificados durante la etapa de aclaración. Dentro de las múltiples fuentes disponibles de búsqueda externa, se optó por la revisión de literatura publicada, patentes y documentación oficial de software como fuentes predilectas. A continuación se describe la revisión de soluciones a los subproblemas anteriormente definidos.

Herramientas para la Creación del Escenario de Simulación

Comenzando por el diseño de un escenario que asemeje situaciones de riesgo dinámicas, se debe considerar que si bien Gazebo es una excelente plataforma de simulación física, no cuenta con herramientas integradas para hacer un diseño sofisticado de escenarios. El modelado 3D disponible en Gazebo se limita a figuras geométricas primitivas y la aplicación de texturas a los modelos consiste únicamente en la selección de un color uniforme. Si bien es posible importar modelos 3D de software como SolidWorks o Autodesk Inventor, estos destacan por su aplicación en el diseño de piezas de ingeniería, no en el diseño de escenarios de simulación completos. Por las razones anteriores, se evidencia la necesidad de emplear otra herramienta capaz de modelar un escenario de pruebas para Gazebo.

La documentación oficial de Gazebo [17] hace alusión a que los dos formatos permitidos para importación de modelos personalizados son el OBJ (.obj)

y el COLLADA (.dae). También recomienda emplear los software Blender o Sketchup para la elaboración de modelos. Como primera decisión de diseño referente a la generación del escenario de validación, se optó por utilizar el formato COLLADA. La razón es que permite almacenar también datos de animaciones para los modelos, algo que no es posible realizar con el formato OBJ. De esta manera se procedió a identificar las características de las dos opciones de software para modelado 3D.

Blender es un software gratuito y de código abierto [18] para la creación de contenido en 3D. Soporta todos los flujos de trabajo relacionados al desarrollo en 3D, incluyendo modelado, montaje, animación, simulación, renderizado, composición e incluso edición de video y creación de juegos. Los usuarios avanzados pueden hacer uso de la API de Blender para crear secuencias de comandos de Python para personalizar la aplicación y escribir herramientas especializadas. También cuenta con una documentación extensa y numerosos ejemplos de proyectos en distintos tópicos.

Sketchup es una suite de productos de suscripción [19] para el modelado 3D. Está particularmente enfocado para aplicaciones en planificación urbana, arquitectura, ingeniería civil y diseño industrial. Ofrece una gran variedad de herramientas que facilitan el diseño 3D, además de una amplia galería de modelos listos para incorporar. Si bien la herramienta cuenta con la capacidad de animar escenas para visualizar modelos desde distintos ángulos y para el estudio de sombras, no cuenta con la opción de animar específicamente los modelos.

Recepción de Datos Geoespaciales

Atendiendo al primer subproblema, la obtención de los datos geoespaciales, es de interés recalcar la necesidad de una medición directa de los datos. Muchas aplicaciones emplean sensores que miden aceleración, presión o giros para estimar posición de forma indirecta. De hecho esta función actualmente se realiza con el IMU incorporado. Se realiza la distinción, porque el interés del proyecto es recopilar información espacial referenciada a cualquier posición de la superficie terrestre de forma directa. Se ha estudiado la viabilidad de tres tecnologías distintas para lograr este cometido.

La primera técnica considerada es la teledetección (RS), que se define como la medición de las propiedades de un fenómeno mediante un dispositivo que no está en contacto íntimo con el objeto de estudio [20]. Si bien el concepto es general y se aplica a múltiples disciplinas, es una técnica

comúnmente empleada para medir fenómenos terrestres desde sensores ubicados en satélites o aeronaves. Long et al. [21] proponen un método de RS para la localización precisa de objetos basándose en redes neuronales convolucionales (CNN). El método consta de tres etapas:

1. Un algoritmo de búsqueda selectiva se encarga de identificar los distintos objetos presentes en la imagen.
2. La CNN se encarga de extraer las características de los objetos para posteriormente categorizarlos.
3. Finalmente el equipo desarrolló un algoritmo no supervisado basado en puntaje para generar un cuadro delimitador en los objetos de interés.

Este método podría ser aplicable al SD, ya que cómo se muestra en la Figura 3.7, el equipo desarrollador fue capaz de identificar aeronaves utilizando imágenes satelitales. Evidentemente, la aplicación de este método no sería implementada de forma práctica con este tipo de imágenes, sino que se podría emplear una aeronave de apoyo, volando de forma estacionaria a gran altitud sobre la zona de despliegue.

El segundo tipo de tecnología considerada es el Sistema de Información Geográfica (GIS). Este consiste en un sistema informático que analiza y muestra información geográficamente referenciada [22]. Utiliza datos que se adjuntan a una ubicación única. El tipo de datos que se pueden almacenar en un sistema GIS incluye elevación, hidrografía, cobertura boscosa, estructuras, caminos, entre otros.

Este tipo de tecnología podría ser de utilidad para obtener la ubicación de la aeronave, siempre y cuando se tengan disponibles los datos con antelación. Un ejemplo de su alcance, se encuentra en el trabajo de Larnaout et al. [23]. En su publicación, desarrollan un método de localización basado en GIS, donde un vehículo equipado con una cámara logra orientarse dentro de un entorno urbano denso, empleando los datos de elevación y modelos tridimensionales de estructuras aledañas.

El método propuesto se divide en:

1. Captura y procesamiento de imágenes.
2. Corrección de la trayectoria del vehículo, en base al emparejamiento con los datos de elevación. En la Figura 3.8, se muestra una comparativa de estimación de la posición con (izquierda) y sin (centro) este paso.



Figura 3.7: Ejemplo de los resultados obtenidos por Long et al. [21]

3. Restricción de la posición, en base a la nube de puntos generada por la cámara y los modelos tridimensionales del sistema GIS.

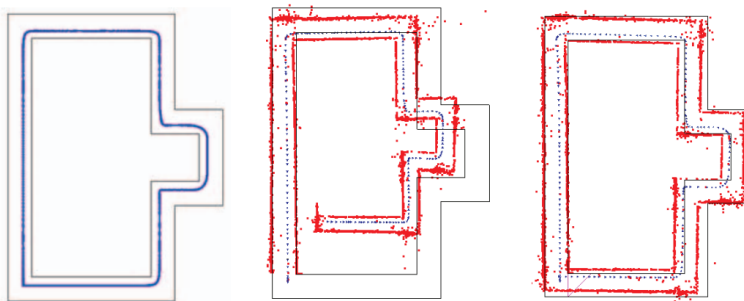


Figura 3.8: Comparativa de la restricción por elevación [23]

Los autores logran una tasa de actualización de la posición de 30 Hz con este método, lo que lo vuelve un candidato apto para solucionar el problema en las aeronaves autónomas, dado que cumple con la necesidad respectiva del proyecto.

Finalmente, el tercer paradigma de recolección de datos geoespaciales identificado cómo candidato a solución es el sistema global de navegación por satélite (GNSS). Este sistema consiste en una constelación de satélites [24], que proporcionan señales desde el espacio que transmiten datos de posicionamiento y tiempo a los receptores GNSS. Luego, los receptores usan estos datos para determinar su ubicación.

Esta tecnología cuenta con amplia adopción en el sector industrial, hecho evidenciado tras la consulta de algunas patentes. Zhou y Wang [25] patentan para la empresa Baidu, un método de control de vehículos aéreos no tripulados, mediante inteligencia artificial. Una de las entradas durante la etapa de entrenamiento, consiste en datos geoespaciales provistos por un GNSS.

Otro ejemplo del uso de esta tecnología, se encuentra por parte de la compañía DJI, uno de fabricantes líderes en vehículos aéreos no tripulados. Con el objetivo de soportar tareas de búsqueda, rescate, vigilancia y exploración, Gong y Fan [26] presentan un método para proporcionar cobertura de comunicación a un vehículo aéreo no tripulado en pleno vuelo. El objetivo es dotar a la aeronave con la capacidad de transmitir y recibir información de forma ininterrumpida desde unidades terrestres, por lo que su ubicación cobra especial relevancia.

Cómo ejemplo adicional, los inventores Abuelsaad et al. [27], de la empresa IBM, registran la patente de un vehículo aéreo no tripulado para la detección de riesgos de accidentes de tráfico. Se hace mención en múltiples instancias de la tecnología GNSS cómo medio para la localización de la aeronave.

Fusión de Señales

Independientemente del medio para la recolección de los datos geoespaciales, se debe contar con un método para fusionar dichos datos con los del IMU. Se descarta el empleo de algún filtro analógico, ya que los datos suministrados por el IMU ya se encuentran en formato digital. De esta manera el espacio de búsqueda se ve reducido a la implementación de algoritmos que tengan la capacidad de mezclar las dos fuentes de información para producir una estimación más confiable.

Adicionalmente se contempla la posibilidad de agregar otras fuentes de información ya disponibles en la aeronave. Concretamente, las cámaras estéreo incorporadas podrían emplearse para robustecer aun más el sistema de localización; no sólo para obtener una estimación más precisa, sino también para servir de respaldo en caso que las fuentes de datos geoespaciales o inerciales

se vean comprometidas.

El primer paradigma de solución explorado, consiste en los filtros de Kalman (KF). Esta es una familia de algoritmos basados en métodos probabilísticos para estimar una variable a partir de información inexacta y ruidosa. El método se basa en combinar información nueva, con la estimación del estado general a partir de información previa.

Caron et al. [28] utilizaron esta técnica para combinar datos de una instancia específica de la tecnología GNSS; el sistema de posicionamiento global (GPS), con los datos de aceleración de una unidad de medición inercial. Dicha técnica fue empleada para determinar la posición de un vehículo terrestre. Desarrollaron la aplicación empleando datos reales. El algoritmo propuesto puede incluir señales adicionales de ser necesario.

También esta técnica se ha empleado en vehículos aéreos no tripulados. Wang et al. [29] plantean el control de un UAV para aplicaciones de agricultura, mezclando señales de GNSS y un sistema de navegación inercial (INS). Hacen especial énfasis en la naturaleza compacta de la implementación y la baja demanda de recursos computacionales. Cómo puede apreciarse en la Figura 3.9, la señal del error se ve significativamente suavizada con el filtrado.

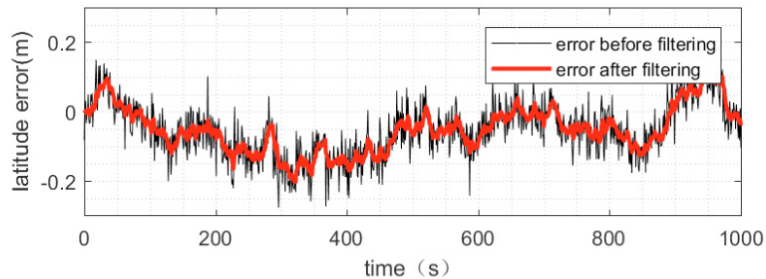


Figura 3.9: Error de latitud bajo la implementación de Wang et al. [29]

Uno de los principales inconvenientes de este método, es que presupone que la dinámica del sistema puede ser representada por ecuaciones lineales, lo cual no siempre es el caso. En respuesta a esta carencia, se han presentado múltiples variaciones que emplean métodos de linealización para dar mejores estimaciones. Los dos más populares corresponden al filtro de Kalman extendido (EKF) y el filtro de Kalman sin aroma (UKF) [30].

Cómo ya se mencionó, una componente importante de información disponible en las aeronaves, son las imágenes captadas por un juego de cámaras estéreo. Para tratar este tipo de información, los filtros de Kalman han sido

empleados con éxito [31] para eliminar el ruido de las imágenes captadas, pero estimar el cambio en la posición de un vehículo a partir de imágenes, se sale del enfoque de este tipo de filtros.

Por la razón anterior, se han propuesto diversas técnicas que logran incorporar el procesamiento de imágenes con la fusión de otro tipo de sensores. Un ejemplo es el método de odometría visual-inercial (VIO) propuesto por Qin et al. [32]. Los autores detallan la fusión de sensores localmente precisos, con sensores globales ruidosos pero libres del problema de deriva. Esto se logra optimizando las restricciones que se generan entre las lecturas de los distintos sensores, a partir de estados sucesivos según se indica en la Figura 3.10. Cada nodo representa la posición y orientación en coordenadas globales y las líneas representan las restricciones mutuamente impuestas

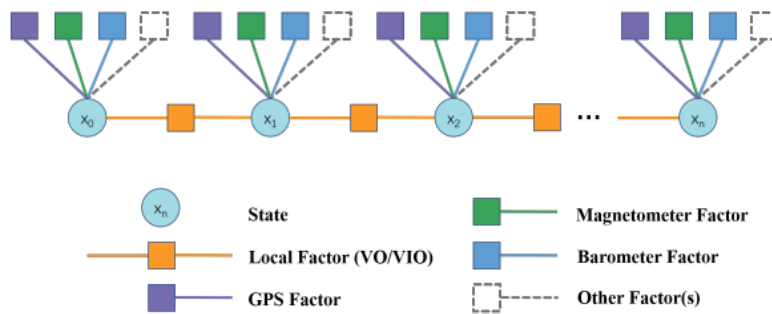


Figura 3.10: Enfoque de Qin et al. [32]

Con este abordaje, los investigadores lograron un error cuadrático medio de localización de 0.4 metros, en pruebas reales con un vehículo terrestre bajo trayectorias que superaron los 230 metros.

Un enfoque moderno ha sido el empleo de inteligencia artificial para la fusión de señales. Concretamente, se han usado aquellas arquitecturas capaces de operar con datos de series de tiempo. Dai et al. [33] emplearon redes neuronales recurrentes (RNN), para integrar un sistema de navegación de botes a partir de señales de GNS e INS. Los investigadores lograron entrenar un modelo capaz de estimar los errores en las señales de entrada, lo que les permitió a su vez corregirlos. Con este enfoque se logró una mejora del 60 % sobre los resultados obtenidos con un EKF.

Otro ejemplo de inteligencia artificial aplicada a la fusión de señales se encuentra en Kim & Bae [34]. Los investigadores utilizan una arquitectura

denominada memoria a corto plazo (LSTM), que consiste en una mejora durante el proceso de entrenamiento sobre la RNN convencional. Aplicando este método para el sistema de navegación de un vehículo terrestre, se obtuvo una mejora en la estimación del 40 % sobre el uso de un GNSS sin filtrado.

3.3.3. Búsqueda Interna

La búsqueda interna busca ejercitar el conocimiento personal y del equipo, para generar conceptos de solución. En este caso se cuenta con la experiencia de dos investigadores de HRC que han estado extensamente involucrados en el desarrollo de las naves autónomas.

El problema que se busca explorar y discutir en esta etapa de la metodología, es la incorporación de la ubicación estimada al resto de subsistemas presentes en la aeronave. Es de gran importancia que el flujo de datos se realice de forma eficiente y en sincronía para garantizar una operación adecuada del autómata.

La principal solución propuesta es el uso de la herramienta Intel Edge Insights. Este es un conjunto de herramientas de OT diseñadas para acelerar el desarrollo y la implementación de soluciones para el sector industrial. El software está destinado a implementarse en dispositivos de generación de datos y permite la ingesta de datos de vídeo y series temporales, el almacenamiento, la realización de análisis y la publicación de los resultados [35]. En concreto, se espera emplear el bus de datos según se muestra en la Figura 3.11 ya que está optimizado para los procesadores de la compañía.

Esta herramienta se ha empleado en aplicaciones tan variadas cómo la inspección de soldaduras para el fabricante de automóviles Audi [36], la detección de defectos en circuitos impresos [37], y la asistencia en la manufactura de productos farmacéuticos [38].

3.3.4. Exploración Sistemática

Tras la búsqueda de información, se vuelve evidente que algunos de los subproblemas pueden ser descompuestos con más detalle. Para el caso de la obtención de datos geoespaciales, se han identificado tres factores que cobran relevancia en la solución implementada:

- Recibir información, que es procesada para estimar la posición del UAV.
- Computar la posición, ya sea de forma externa o por la misma aeronave.

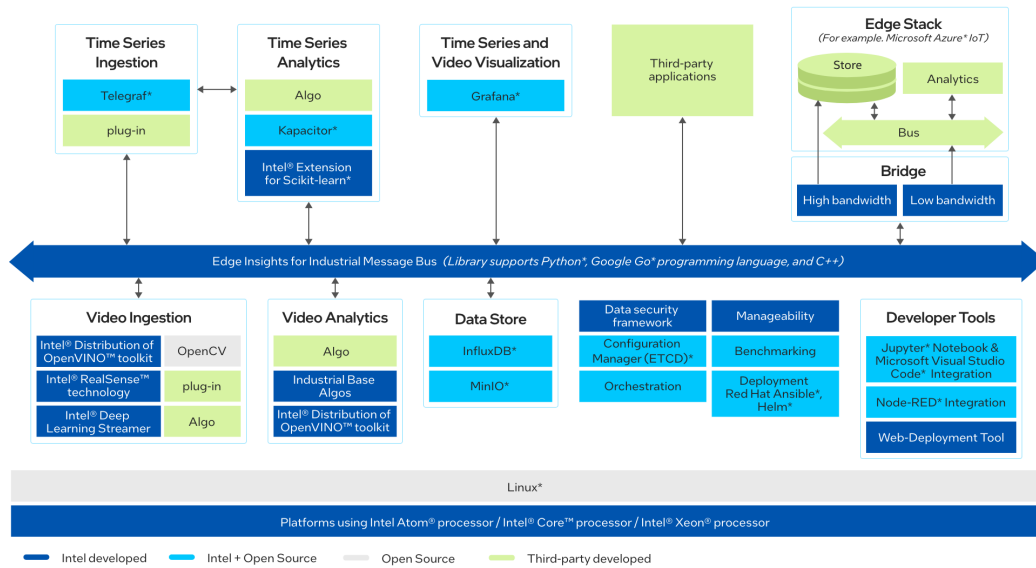


Figura 3.11: Vista general del paquete Intel Edge Insights

- Transmitir los datos, para que el algoritmo de localización pueda hacer uso de los mismos.

En base a estos factores, se generó una tabla de combinación de conceptos para identificar los candidatos a solución más prometedores. En las Figuras 3.12, 3.13 y 3.14 se muestran respectivamente los conceptos:

- A, que consiste en equipar un UAV de soporte con una cámara, para rastrear la posición de múltiples aeronaves de forma simultánea.
- B, cuyo propósito es procesar las mediciones de distancia respecto a distintos satélites, mediante un receptor equipado en la aeronave.
- C, dónde se precarga información del terreno de forma anticipada, para identificar la posición en base los elementos reconocidos.

En referencia a la fusión de las señales y estimación de la posición, se identifican dos puntos de influencia que determinarán el tipo de solución desarrollada:

- El uso de las cámaras estéreo para estimar la posición.

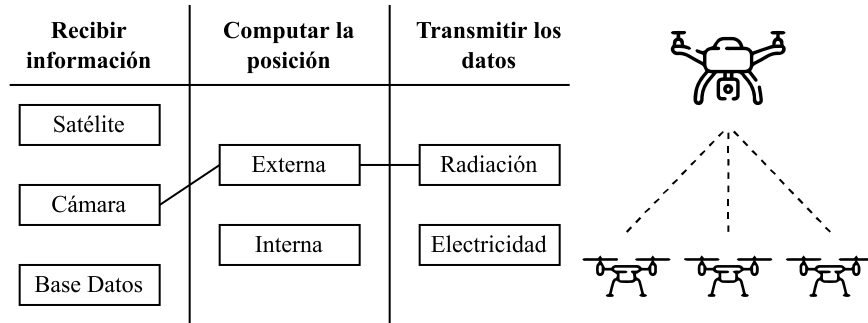


Figura 3.12: Concepto A

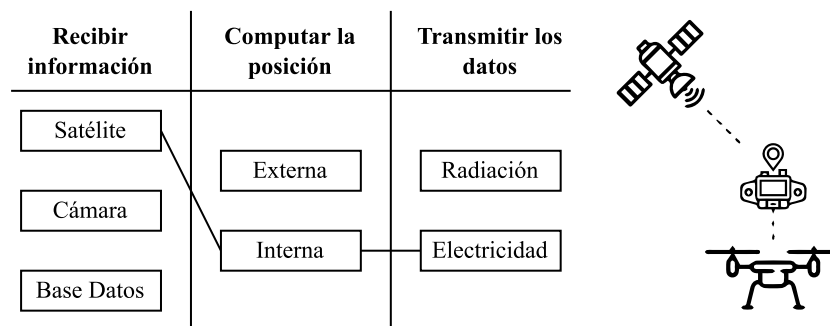


Figura 3.13: Concepto B

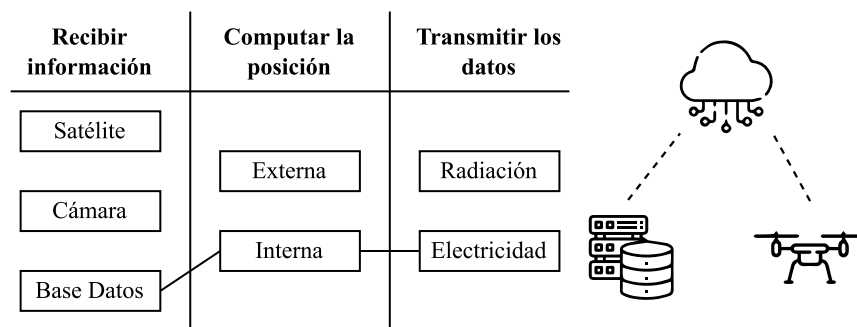


Figura 3.14: Concepto C

- El paradigma de fusión de señales.

En las Figuras 3.15, 3.16 y 3.17 se encuentran representaciones gráficas con las combinaciones más plausibles para realizar la estimación de la posición. Estos conceptos son:

- D, emplear redes neuronales recurrentes para aprender un modelo del ruido en las señales de entrada.
- E, emplear un filtro de Kalman sin aroma para estimar la ubicación más probable, en base a la lectura de los sensores y estados previos.
- F, estimar la localización optimizando las restricciones mutuas entre las lecturas de los sensores y las imágenes captadas

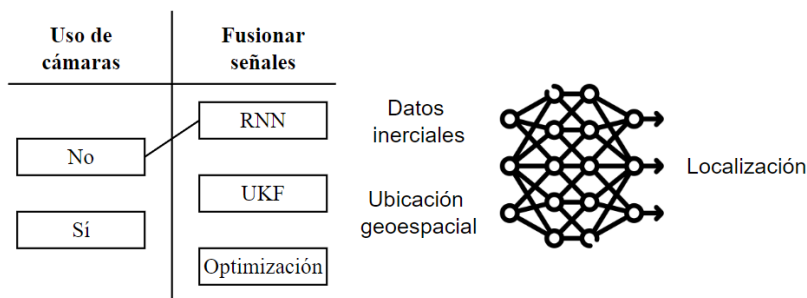


Figura 3.15: Concepto D

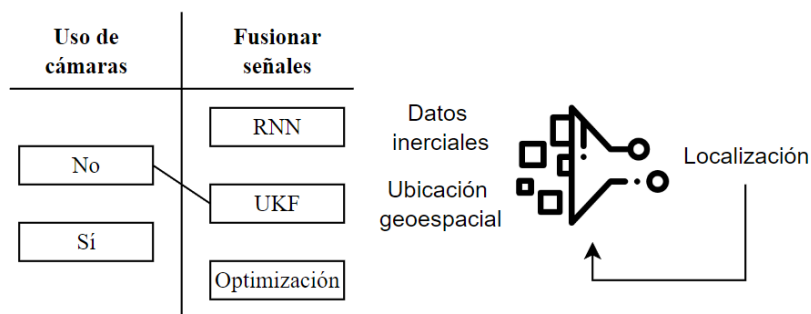


Figura 3.16: Concepto E

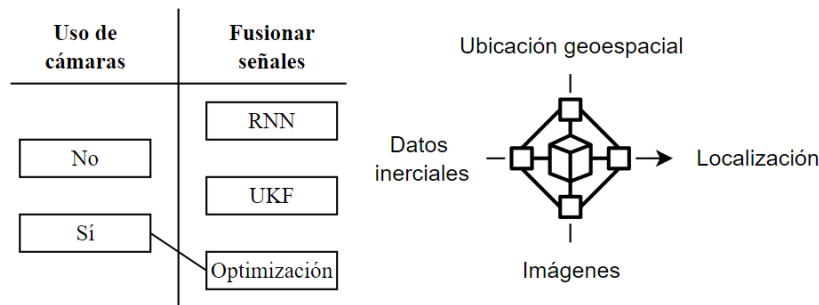


Figura 3.17: Concepto F

3.3.5. Reflexión sobre las Soluciones y el Proceso

Se puede afirmar que se ha explorado el espacio de solución adecuadamente para dos de los tres subproblemas identificados. Tanto para la recopilación de datos geoespaciales, cómo para la fusión de los mismos con el IMU se estudiaron alternativas con enfoques muy distintos.

La única excepción reside en el subproblema de transmitir la localización estimada, ya que la solución fue identificada durante la etapa de búsqueda interna y tiene cómo objetivo satisfacer una necesidad específica.

Con los conceptos que sí están abiertos a evaluación, se procederá a un análisis sistemático de los mismos para realizar una selección final. También se someterá a comparación las dos opciones contempladas para el diseño del escenario de simulación.

3.4. Selección del Concepto

La selección del concepto consiste en la etapa de evaluación para las soluciones prometedoras, sintetizadas en la etapa anterior. Se deben considerar tanto las necesidades del cliente cómo criterios inherentes a cada subproblema. De esta forma se comparan los puntos fuertes y débiles de cada concepto, para la selección de uno o varios y posteriormente ejecutar su desarrollo y evaluación.

3.4.1. Elaboración de Matriz de Selección

El método preliminar elegido para la discriminación de conceptos es el de matrices de decisión. Este es un método estructurado que se basa en puntuar los conceptos mediante criterios previamente establecidos, lo que ayuda a mantener la objetividad durante el proceso de selección. La siguiente lista detalla los factores seleccionados para cada subproblema.

- Motor de diseño para la simulación:
 - Costo económico, ya que es preferible reservar los gastos para la captación de los datos geoespaciales.
 - Curva de aprendizaje, debido a que el tiempo para el desarrollo del proyecto es limitado, siendo preferible emplear la mayoría del tiempo en el diseño del sistema de localización.
 - Posibilidad de animar modelos, porque sería de utilidad para extender las funcionalidades de la simulación en el futuro.
 - Soporte en línea disponible, ya que sería de utilidad en caso necesitar ayuda con problemas que puedan surgir durante la elaboración.
- Obtención de datos geoespaciales:
 - Posibilidad de control a múltiples objetivos, debido a que las aeronaves se pueden desplegar en grupos y un único control de objetivos podría abaratar costos y facilitar las ubicaciones relativas.
 - Sensibilidad a interferencias, ya que es necesario contar con un sistema robusto cuyo desempeño no se vea afectado con facilidad.
 - El coste computacional, porque el procesador a bordo de las aeronaves no es especialmente potente.
 - Necesidad de preparativos previos, que afectaría a la solución en casos donde los cuerpos de emergencia tengan un margen de tiempo reducido para atender la situación.
 - Dependencia espacial con la fuente de información, ya que esto podría ser un impedimento para abarcar grandes extensiones de terreno y gozar de autonomía.
 - Coste económico, ya que por obvia razones mientras más reducido sea el costo de implementación sin comprometer la calidad, mayor será el margen de ganancia.

- Estimación de la posición:
 - Exactitud esperada, ya que este valor no se conocerá hasta realizar las pruebas de verificación, por lo que se debe tomar una decisión informada en base a la literatura que se consultó.
 - Desarrollo con datos de simulación, porque la necesidad de datos reales para diseñar la solución generaría un conflicto directo con las condiciones del proyecto.
 - Coste computacional, por las mismas razones mencionadas en la obtención de datos geoespaciales.
 - Robustez, en caso de que uno de los sensores falle o reporte mediciones erróneas de forma consecutiva.
 - Escalabilidad, en caso de que se mejore o agreguen sensores a la aeronave, se debe tener un sistema generalista.

3.4.2. Evaluación de Conceptos

Para cada uno de los criterios propuestos, se asignará una puntuación según la siguiente escala:

- (+): el concepto destaca positivamente en este criterio respecto a los demás.
- (0): el concepto posee una posición neutral en este criterio respecto al resto de alternativas.
- (-): el concepto destaca negativamente en este criterio respecto a los demás.

3.4.3. Ordenación de Conceptos

Se comenzará por abordar el problema de la generación del escenario de pruebas. Dado que elaborar este escenario en Gazebo es un requisito del proyecto, se contemplan únicamente las dos alternativas aconsejadas por la documentación oficial para el diseño de escenarios, según se mencionó en la Sección 3.3.2. De forma explícita, en el Cuadro 3.4 se muestra la matriz de decisión, que compara las dos plataformas de desarrollo.

Cuadro 3.4: Matriz de decisión para la generación del escenario de validaciones

Criterio	Blender	Sketchup
Costo económico	+	-
Curva de aprendizaje	-	+
Posibilidad de animar modelos	+	-
Soporte disponible	0	+
Suma +	2	2
Suma 0	1	0
Suma -	1	2
Evauación neta	1	0
¿Continuar?	Sí	No

Atendiendo al subproblema de recolección de datos geoespaciales, se deben considerar las distintas combinaciones sintetizadas en la Sección 3.3.4. Todas se incluyeron en la matriz de decisión para este subproblema, que se muestra en el Cuadro 3.5.

Cuadro 3.5: Matriz de decisión para el subproblema de obtención de datos geoespaciales

Criterio	A	B	C
Posibilidad de control de múltiples objetivos	+	0	0
Sensibilidad a obstáculos e interferencias	-	0	+
Coste computacional	-	+	-
Necesidad de preparativos previos	0	0	-
Dependencia espacial con la fuente de información	-	0	+
Coste económico	-	+	0
Suma +	1	2	2
Suma 0	1	3	2
Suma -	4	0	2
Evauación neta	-3	2	0
¿Continuar?	No	Sí	No

Finalmente, y de forma similar al caso anterior, se evalúan los conceptos de solución D, E y F para estimar la ubicación de la aeronave. La comparativa se muestra en el Cuadro 3.6.

Cuadro 3.6: Matriz de decisión para el subproblema de fusión de señales

Criterio	D	E	F
Exactitud esperada	0	+	+
Desarrollo en simulación	+	-	+
Coste computacional	+	-	-
Robustez	-	+	+
Escalabilidad	0	0	+
Suma +	2	2	4
Suma 0	2	1	0
Suma -	1	2	1
Evauación neta	1	0	3
¿Continuar?	No	No	Sí

3.4.4. Combinación y Mejora de Conceptos

Combinar conceptos tiene cómo finalidad:

- Eliminar las características negativas de un concepto bueno en términos generales.
- Mejorar el concepto general de solución con modificaciones menores.
- Mezclar dos conceptos para neutralizar sus debilidades y potenciar sus bondades.

Por estas razones, es deseable reflexionar sobre la posibilidad de combinar conceptos para su mejora antes de proceder a su selección final. Se comenzará por evaluar el motor de desarrollo para la simulación. En este caso, los conceptos son mutuamente excluyentes ya que intentar realizar partes del diseño en las dos plataformas al mismo tiempo podría generar conflictos de compatibilidad y no aportaría ningún valor si con la opción ganadora se pueden llevar a cabo todas las tareas necesarias.

En el caso de la obtención de datos geoespaciales, sí existe una ventaja de emplear más de un método de captación de información para obtener una lectura más robusta, pero el principal inconveniente es la limitación impuesta por el coste económico, por lo que se apostará por ganar robustez durante la etapa de fusión de señales.

Al contrario del caso anterior, para fusionar las señales de entrada no existe ganancia en emplear más de un método, sino justo lo contrario, ya que

el coste computacional se elevaría y dado que las señales de entrada serían las mismas para todos los sistemas, no existiría una ventaja en este sentido. Se priorizará el desarrollar un sólo método robusto y bien optimizado.

3.4.5. Selección de Conceptos

En referencia a los puntos tratados en este capítulo, la solución completa para el problema en estudio será la incorporación de un sensor GNSS a la aeronave, cuyas lecturas se fusionarán con los datos de la unidad de medición inercial y las imágenes captadas por el sistema de cámaras estéreo mediante la optimización de las restricciones mutuamente impuestas entre todas las fuentes de datos. El resultado de la localización estimada se enviará al resto de subsistemas de la aeronave mediante el bus de datos ofrecido por Intel Edge Insights. El desempeño de este sistema se verificará con un escenario de simulación robótica ejecutado en Gazebo y diseñado en Blender.

3.4.6. Reflexión sobre los Resultados y el Proceso

Tras una revisión extensa para la búsqueda de conceptos de solución, se puede afirmar que las selecciones gozan de una comparación objetiva frente a otras alternativas que terminaron por ser descartadas. La propuesta fue discutida con el asesor industrial y cuenta con su aprobación.

El método para la captación de datos geoespaciales y la fusión de los mismos con información inercial, constituyen respectivamente propuestas de hardware y software de las que se espera obtener un rendimiento de localización superior frente a las opciones comerciales disponibles para vehículos aéreos dentro del costo presupuestado, ya que usualmente estas hacen especial énfasis en los componentes a nivel de hardware y no del tratamiento que se da a la información generada.

CAPÍTULO 4

PROPUESTA DE DISEÑO

En este capítulo se mostrará la implementación de los conceptos de solución elegidos, en base al seguimiento de la metodología de diseño. El contenido se ha dividido en secciones que se alinean con los objetivos propuestos en el Capítulo 1.

4.1. Diagnóstico del Sistema Actual

Se ha comentando que actualmente la localización se realiza sólo con las lecturas de un IMU. Esto vuelve vulnerable el desempeño en general del sistema ya que un fallo en este sensor ocasionaría el fallo total de la localización. Incluso cuando este opera de forma correcta, la localización está sujeta al problema de la deriva; una degradación del desempeño que ocurre con la acumulación de errores locales. Pese a lo anterior, el IMU es un insumo altamente valioso para estimar la posición y orientación del vehículo aéreo. Los autómatas están equipados con el dispositivo Intel RealSense T265 [39], que a su vez incorpora el IMU del fabricante Bosch modelo BMI055, un juego de cámaras estéreo y un sensor de profundidad. El IMU ofrece mediciones de aceleración lineal con tres grados de libertad a 62.5 Hz, y mediciones de velocidad angular con tres grados de libertad a 200 Hz, lo que lo vuelve el sensor con la mayor tasa de actualización de los tres que se emplean en la solución propuesta.

Debido a la importancia de este sensor en la solución final, es necesario

realizar un diagnóstico del desempeño que se tiene al emplear únicamente sus lecturas para estimar la posición. Esto no sólo es relevante para determinar el porcentaje de mejora comparativa que se obtendrá con el sistema diseñado, sino que también es un insumo útil a considerar durante la propia implementación del diseño, ya que puede arrojar información sobre las particularidades que se deben atender en el tratamiento de las lecturas.

Dado que en la extensión del proyecto, el sistema diseñado se verificará únicamente a través de simulación, una comparación justa con el sistema de localización actual parte de tener una versión virtual del IMU que asemeje los más fielmente posible sus características reales. Dicha instancia virtual debe ser utilizada sin alteraciones tanto en los experimentos de diagnóstico, como en los de verificación. A petición del cliente, estos experimentos serán realizados con el framework de simulación RotorS [40], que facilita la ejecución de pruebas con aeronaves no tripuladas en Gazebo.

Gazebo cuenta con diversos plugins que permiten la simulación de sensores. Estos se configuran a través de archivos con formato XML. Usando la documentación [41], se instanció el sensor IMU a partir de la configuración base de RotorS, alterando los siguientes parámetros:

- Masa del IMU.
- Densidad del ruido del giroscopio.
- Densidad del ruido del acelerómetro.

Para el valor de la masa del IMU se empleó la masa total del dispositivo RealSense T265, ya que el primero se encuentra incorporado dentro del segundo. De esta forma cuando se realicen experimentos con las cámaras se agregará únicamente la funcionalidad de las mismas sin añadir masa extra al sistema. Las características mecánicas del RealSense T265 se adjuntan en el Anexo B.2. Los valores de densidad de ruido se extrajeron directamente de la hoja de datos respectiva, según se indica en el Anexo B.3.

Una conversión básica fue requerida introducir la densidad de ruido del giroscopio en las unidades requeridas por Gazebo, según se detalla en la Ecuación 4.1.

$$n_{\text{gyro}} = 0.014 \frac{\text{°}}{\text{s} \cdot \sqrt{\text{Hz}}} \approx 2.443 \times 10^{-4} \frac{\text{rad}}{\text{s} \cdot \sqrt{\text{Hz}}} \quad (4.1)$$

Para el resto de parámetros modificados, las unidades reportadas en las hojas de datos son compatibles. Se debe recalcar que no todos los parámetros configurables de ruido fueron modificados, ya que algunos no estaban disponibles de forma pública en las hojas de datos por el fabricante Bosch. En este caso se optó por mantener los valores que vienen por defecto en RotorS. La inercia del sensor se considera despreciable. De esta forma la configuración completa de los parámetros sigue:

Código 4.1: Parámetros de simulación del IMU

```

1 <!-- BMI055 IMU -->
2 <xacro:imu_plugin_macro
3   ...
4   mass_imu_sensor="0.060"
5   gyroscope_noise_density="0.000244346"
6   gyroscope_random_walk="0.000038785"
7   gyroscope_bias_correlation_time="1000.0"
8   gyroscope_turn_on_bias_sigma="0.0087"
9   accelerometer_noise_density="0.001470997"
10  accelerometer_random_walk="0.006"
11  accelerometer_bias_correlation_time="300.0"
12  accelerometer_turn_on_bias_sigma="0.1960">
13   ...
14 </xacro:imu_plugin_macro>

```

Antes de proceder con las pruebas de diagnóstico en el simulador, se debe seleccionar un modelo de aeronave desde la biblioteca de RotorS. Tras una revisión de todas las opciones disponibles, se optó por seleccionar el modelo Ardrone, ya que sus características físicas y dimensiones son las que mejor se ajustan al diseño real de los prototipos de HRC [42]. En la Figura 4.1 se muestra la instancia de este modelo de aeronave en Gazebo.

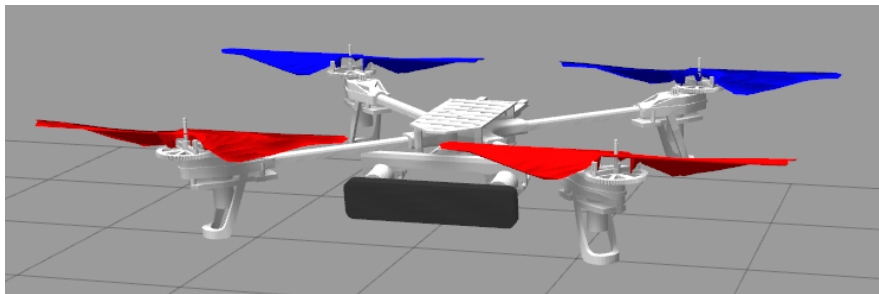


Figura 4.1: Modelo de aeronave Ardrone equipada con el RealSense T265

Para las pruebas de diagnóstico no se ha seleccionado ningún escenario en específico. Esto porque el método de localización se basa únicamente en las fuerzas que actúan en la aeronave y es agnóstico a las características del entorno. La elaboración del escenario de simulación, indispensable para verificar el sistema diseñado, se discute en la Sección 4.4.

Se ejecutaron tres pruebas de diagnóstico con la intención de identificar dependencias del desempeño del IMU respecto al tipo de trayectoria seguida. En cada prueba, la aeronave es operada manualmente en patrones de navegación fundamentalmente distintos entre sí, alrededor de un minuto. En la Figura 4.2 se muestra la comparación entre la ruta real y la estimación empleando el IMU para cada prueba. Todos los datos de este experimento, y los ejecutados en el Capítulo 5, pueden ser consultados en el Apéndice A.3.

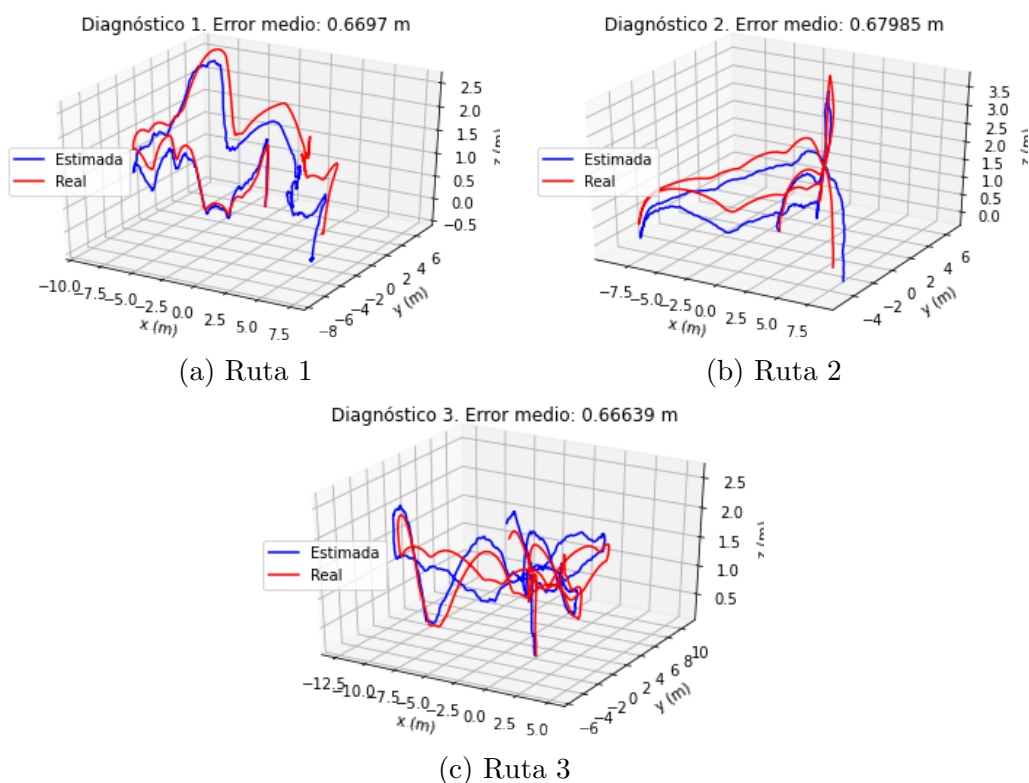


Figura 4.2: Trayectorias ejecutadas en el experimento de diagnóstico

La métrica de error seleccionada, es el valor medio de distancia euclidiana entre el punto estimado p_i y la ubicación real \hat{p}_i para cada instante de tiempo.

$$\text{error} = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \quad (4.2)$$

El valor del error para este experimento tiene una tendencia a la alza conforme aumenta el tiempo, por la deriva que se genera al emplear únicamente mediciones de naturaleza local. No obstante, se restringe la ventana temporal de la prueba para poder establecer un punto de comparación con el sistema diseñado. En el Capítulo 5 se ejecutan valoraciones del desempeño tanto en el corto como en el largo plazo, y adicionalmente se evalúa la evolución del error de estimación en el tiempo para el sistema actual y el sistema diseñado.

Para analizar el comportamiento del IMU entre las distintas pruebas, se recurrió al análisis de varianza de una vía. Esta herramienta se utiliza para comparar la media de tres o más muestras y determinar con un nivel elegido de confianza estadística, si estas difieren de forma significativa. La distribución de los datos se observa en la Figura 4.3.

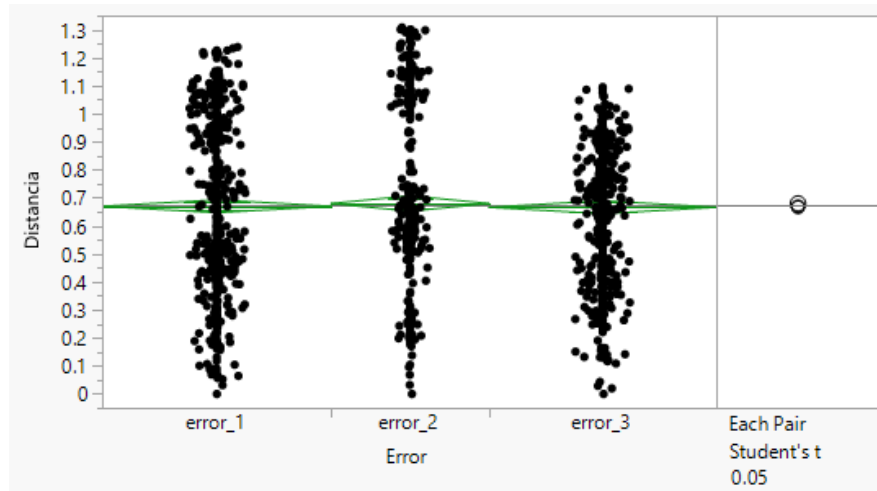


Figura 4.3: Distribución del error para los experimentos de diagnóstico

El nivel de confianza estadística elegido para esta prueba y las realizadas en el apartado de diseño de experimentos, es del 95 %. Este es el valor aceptable de incertidumbre empleado por la compañía Intel en la mayor parte de sus procesos productivos. Bajo estas condiciones el análisis de varianza mostrado en la Figura 4.4 indica que el valor de la prueba es de 0.7560, muy

superior al valor límite de 0.05. De esta manera se concluye que no es posible rechazar la hipótesis nula de la prueba y por tanto no se encuentran diferencias significativas entre las muestras.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
error	2	0.04441	0.022205	0.2798	0.7560
Error	1550	123.02003	0.079368		
C. Total	1552	123.06444			

Figura 4.4: Análisis de varianza para el experimento de diagnóstico

La principal interpretación de este resultado, se puede resumir en los siguientes puntos:

- El modelo del sensor y las características de ruido seleccionadas, tienen un desempeño independiente de la trayectoria seguida.
- Para dos o más ventanas temporales de longitud similar, la degradación en la estimación producida por el ruido del sensor es semejante.

De esta manera, empleando el nivel de significancia antes mencionado, y usando los datos de los tres vuelos, se acota el error medio de localización para un minuto de vuelo:

$$\text{error}_{\text{actual}} = 0.67108 \pm 0.01402 \text{ m} \quad (4.3)$$

4.2. Arquitectura para la Captación de Datos Geoespaciales

Tras una valoración de distintas opciones, en el Capítulo 3 se llegó a la conclusión de que el mejor mecanismo para habilitar la captación de datos geoespaciales es mediante la incorporación de un sensor GNSS.

De acuerdo con las especificaciones objetivo planteadas, el sensor seleccionado deberá ser capaz de operar con una tasa de refrescamiento igual o superior a 1 Hz como mínimo, y superior a 5 Hz preferiblemente. Además debe reportar la longitud y latitud de la aeronave, y si es posible también la altitud. El costo se debe mantener por debajo de los 300 USD, ya que debe existir margen para la incorporación de elementos adicionales que sirvan de soporte para su fijación en la aeronave.

4.2.1. Selección del Hardware

Tras una revisión inicial de proveedores, se han detectado fundamentalmente dos tipos de tecnología GNSS que cumplen con las restricciones de tamaño y masa para ser incorporados a la aeronave:

- Receptores convencionales: poseen una precisión horizontal en metros, requieren la incorporación de una antena y tienen un bajo costo.
- Receptores de Cinemática en Tiempo Real (RTK): poseen una precisión horizontal en centímetros, requieren el uso de antenas en un receptor estático y otro móvil, tienen un costo elevado.

Si bien la dimensionalidad de estos últimos es apta para el uso en aeronaves de tamaño reducido, de todos los proveedores consultados, ninguno cumple con los requerimientos de costo para el proyecto.

Se priorizan aquellos módulos que faciliten transmisión de datos y alimentación mediante USB, ya que de esta manera se puede incorporar con menor riesgo de comprometer la transmisión de datos al procesador Intel Atom x7-Z8750. Otro aspecto a considerar, es que los receptores GNSS esencialmente aportan información horizontal de la ubicación. Para obtener mediciones de altitud, es necesario agregar las lecturas de un barómetro. Si bien este último requisito no es indispensable para cumplir con las necesidades base del proyecto, podría ofrecer una ventaja considerable en la estimación de la posición.

Un último punto a tomar en cuenta es el tipo de antena que se emplea para captar las señales satelitales. Estas pueden ser convencionales o de cerámica, según se muestra en la Figura 4.5. Para el sistema diseñado, es preferible el uso de antenas de cerámica ya que poseen una estructura más compacta.

El Cuadro 4.1 muestra una comparativa entre distintos proveedores de módulos GNSS. La tercera y cuarta columna acotan la precisión horizontal y vertical. La quinta columna se refiere a la frecuencia de actualización.

De las opciones consideradas, se optó por emplear el módulo Matek AP M9N-F4, Figura 4.6a, ya que posee los mejores valores de operación en relación a su costo.

4.2.2. Acoplamiento Mecánico

Para poder fijar de forma segura este módulo a la aeronave, se diseñó un cobertor en base a las dimensiones reportadas en [49]. Dicho insumo se

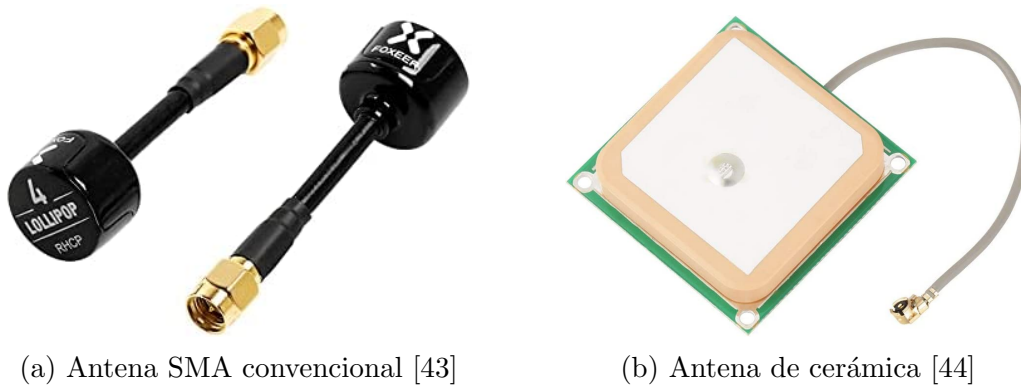
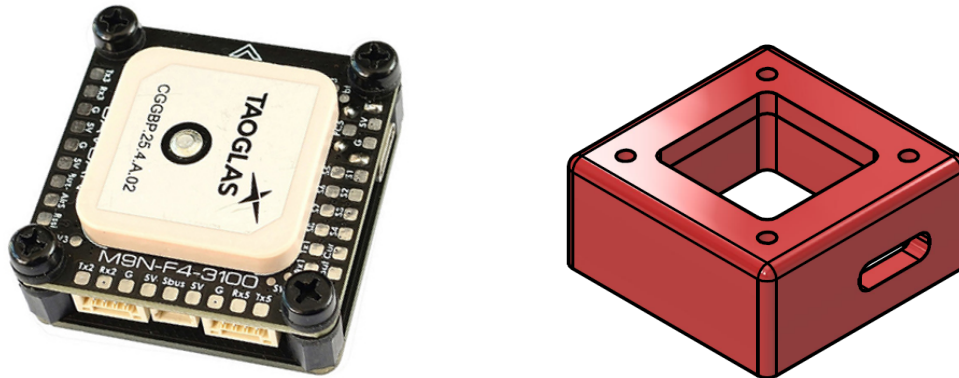


Figura 4.5: Tipos de antena para receptores GNSS

Cuadro 4.1: Comparativa entre módulos GNSS

Módulo	Costo (USD)	P.H. (m)	P.V. (m)	Act. (Hz)	USB	Cita
ELT0114	60.00	2.0	N/A	20	sí	[45]
Matek M8Q	44.80	2.5	0.5	18	no	[46]
Zubax GNSS 2	149.00	2.5	2.5	15	sí	[47]
UBX-M8030-KT	16.99	2.0	N/A	18	sí	[48]
Matek AP M9N-F4	101.72	2.0	0.5	25	sí	[49]

muestra en la Figura 4.6b. Este elemento tiene la función de evitar una exposición excesiva de la circuitería al medio de trabajo, al mismo tiempo que permite la captación de señales satelitales de forma ininterrumpida.



(a) Módulo GNSS modelo Matek AP M9N-F4 [49]

(b) Cobertor diseñado para el módulo GNSS

Figura 4.6: Hardware y fijación mecánica para la captación de datos geospaciales

La acotación completa del cobertor se indica en el plano respectivo, en el Anexo B.1. Como parte del entregable relacionado a este objetivo, en el Apéndice A.1 se detallan las instrucciones de ensamblaje y la lista de partes completa.

El método de construcción designado para el cobertor, es por medio de manufactura aditiva. Se ha seleccionado como material designado el plástico ABS ya que posee una densidad de $1.05 \text{ g}\cdot\text{cm}^{-3}$, lo que lo convierte en un material ligero frente a otros polímeros para manufactura aditiva convencional [50]. El sólido resultante del cobertor fue optimizado para su manufactura con el modelo de impresora Makerbot Replicator 2X disponible en la compañía, según se muestra en la Figura 4.7.

Restricciones de Carga

Sobre el estrés que ejercen los nuevos componentes en las aeronaves, se debe mencionar que el rendimiento energético se ve impactado por la carga que se transporta. Si bien no existe un requerimiento específico de la masa

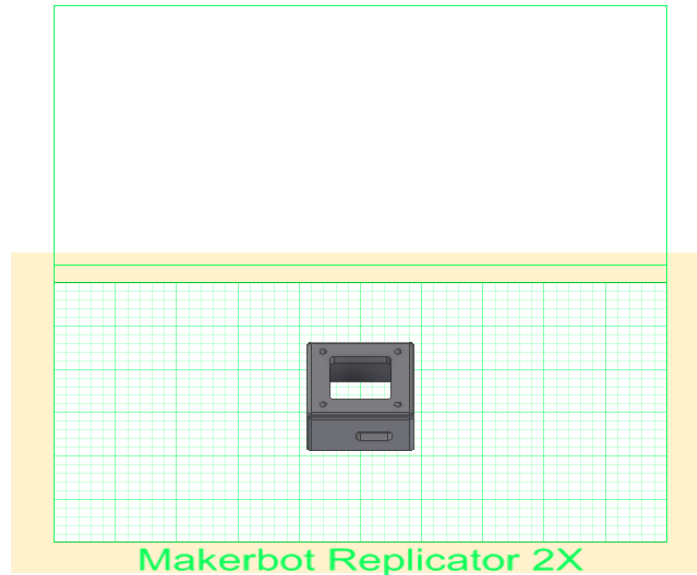


Figura 4.7: Configuración para manufactura con Makerbot Replicator 2X

máxima que se puede agregar al sistema, se realizó una comprobación de los distintos componentes mencionados en el Apéndice A.1, para verificar que el impacto total al sistema sea reducido. Los detalles se indican en el Cuadro 4.2.

Cuadro 4.2: Masa unitaria de los elementos físicos del sistema diseñado

Elemento	Especificación	Cantidad	Masa (g)
Módulo GNSS	M9N-F4	1	23.7
Perno Cabeza Hueca	M3 x 0.5 x 25mm	4	1.7
Tuerca	M3 x 0.5 mm	4	0.4
Cobertor	N/A	1	15.8
Cable Micro-USB	0.5 ft	1	10.0

El cálculo de la masa total añadida al sistema, da como resultado 57.9 gramos. Comparativamente, la masa reportada del sensor Intel RealSense T265 en el Anexo B.2 es de 60 gramos, por lo que la implementación física del sistema diseñado no sólo cumple con todos los requerimientos impuestos en las especificaciones objetivo, sino que también se encuentra dentro de un

margen más que razonable para las capacidades operativas de carga en la aeronave.

4.3. Fusión de Señales

Tras el ejercicio de la metodología de trabajo en el Capítulo 3, se decidió realizar la fusión de los distintos sensores en concordancia con el método expuesto por Qin et al. [32]. En esta sección se profundiza en los detalles conceptuales del método y la implementación.

El método realiza un tratamiento de las señales dependiendo de su clasificación:

- Lecturas locales, que se caracterizan por no estar globalmente referenciadas, por lo que es necesario definir un sistema de referencia que generalmente coincide con la primer lectura del sensor. Por lo general estas lecturas se actualizan a una alta frecuencia. La información proveniente de cámaras, giroscopios, acelerómetros y láseres generalmente cae dentro de esta categoría.
- Lecturas globales, donde el sistema de referencia es la superficie terrestre y se mantiene fijo. Las lecturas están libres de deriva, pero generalmente son ruidosas. La tasa de actualización suele disminuir respecto a los sensores locales. Dentro de este apartado típicamente se incluyen los barómetros, magnetómetros y dispositivos de navegación por satélite.

4.3.1. Estimación Local

Para la estimación local, dentro del método en estudio se puede emplear cualquier algoritmo de odometría visual inercial (VIO) siempre que este tenga por salida una estimación local con seis grados de libertad, correspondiendo a tres coordenadas de localización y tres ángulos que indiquen la orientación.

En la presente implementación esto se logra con la extracción de características en la imagen y posteriormente determinar su profundidad. El RealSense T265 incorpora un juego de cámaras estéreo con eje ópticos paralelos, por lo que la profundidad de los objetos en la imagen se puede estimar de forma geométrica al conocer los parámetros extrínsecos de la cámara. Una representación de este método se muestra en la Figura 4.8, con la relación por triángulos semejantes:

$$\frac{d}{b} = \frac{f}{z} \quad (4.4)$$

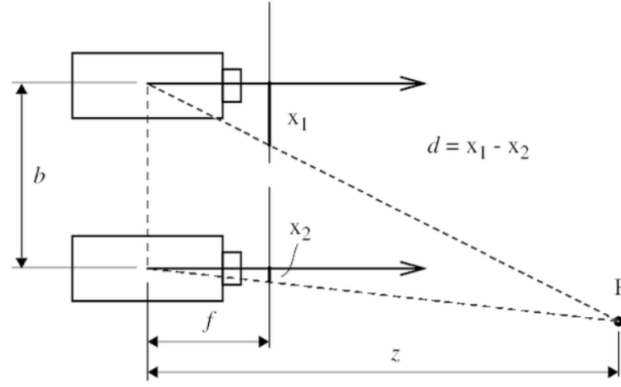


Figura 4.8: Determinación de profundidad por medio de triángulos semejantes [51]

Los objetos detectados en la imagen se parametrizan por el inverso de su profundidad, λ , la primera vez que son captados por la cámara. Por otra parte, el estado \mathbf{x} del IMU en el instante k se puede representar matemáticamente cómo:

$$\mathbf{x}_k = [\mathbf{p}_k^l, \mathbf{v}_k^l, \mathbf{q}_k^l, \mathbf{b}_a, \mathbf{b}_g] \quad (4.5)$$

Usando el sistema de coordenadas local l , los términos \mathbf{p}_k^l , \mathbf{v}_k^l y \mathbf{q}_k^l hacen referencia a la posición, velocidad y orientación percibida por el IMU incorporado a la aeronave. Por otra parte, \mathbf{b}_a y \mathbf{b}_g se refieren al sesgo del giroscopio y del acelerómetro respectivamente. De esta forma, el estado completo \mathcal{X} del sistema IMU-Cámaras se define para una ventana temporal de longitud n cómo:

$$\mathcal{X}_l = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \lambda_0, \lambda_1, \dots, \lambda_n] \quad (4.6)$$

En base a esta estructura, la estimación local de la pose se puede formular cómo un problema no lineal de mínimos cuadrados. El desarrollo completo se puede consultar en [52]. El resultado principal del desarrollo antes mencionado, es que se puede obtener una estimación local de la posición y orientación de la aeronave (o cualquier otro cuerpo) con seis grados de libertad.

4.3.2. Estimación Global

El método elegido tiene como objetivo fundamental dar una ubicación con alta exactitud, precisa y referenciada globalmente. Esto se logra optimizando las restricciones mutuas que, en el caso del sistema diseñado, imponen los factores mostrados en la Figura 4.9. El diagrama representa que cada estimación genera un factor local VIO que es condicionante del estado siguiente y se añaden los factores de sensores globalmente referenciados. La definición matemática de los factores de VIO, barómetro y GNSS se realiza en el Apéndice A.2.

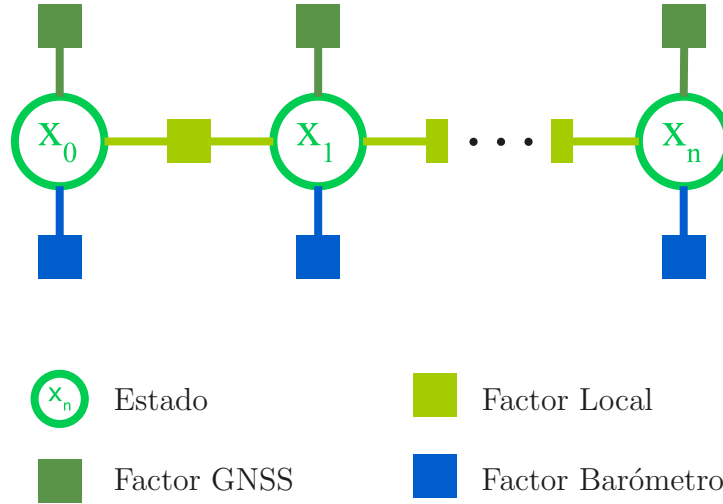


Figura 4.9: Representación gráfica de los factores de localización

La importancia de la Figura 4.9 radica en señalar que los estados de la aeronave representan un sistema de grafos y su optimización es un problema de estimación por máxima verosimilitud (MLE). La MLE consiste en la distribución de probabilidad conjunta de las poses del vehículo en una ventana de tiempo. Las variables son las poses globales de todos los nodos:

$$\mathcal{X} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\} \quad (4.7)$$

Dónde se tiene:

$$\mathbf{x}_i = \{\mathbf{p}_i^w, \mathbf{q}_i^w\} \quad (4.8)$$

Para la posición \mathbf{p} y orientación \mathbf{q} en el sistema de coordenadas global w . Bajo el supuesto que las probabilidades de todas las mediciones \mathbf{z} son independientes, el problema se formula cómo:

$$\mathcal{X}^* = \operatorname{argmax}_{\mathcal{X}} \prod_{t=0}^n \prod_{k \in \mathbf{S}} p(\mathbf{z}_t^k | \mathcal{X}), \quad (4.9)$$

Dónde \mathbf{S} es el conjunto de mediciones que incluye tanto sensores locales como globales. Asumiendo que la incertidumbre de las mediciones siguen una distribución gaussiana, el problema se puede reformular como una optimización no lineal de mínimos cuadrados. El desarrollo completo se puede consultar en [32].

4.3.3. Implementación Programada

Si bien los autores del método de estimación global ofrecen una implementación de código abierto basada en ROS, existen dos limitaciones principales para su uso en el sistema diseñado:

- El framework destinado para el manejo de las comunicaciones es Intel Edge Insights, no ROS.
- De forma pública, no existe ningún precedente de uso en el simulador Gazebo.

Por las razones anteriores, esta implementación tuvo que ser modificada para cumplir con los requerimientos del proyecto. El código resultante es demasiado extenso para ser discutido en su totalidad en el presente informe, por lo que en esta sección se incluyen únicamente los aspectos más relevantes. La implementación completa puede ser consultada en el repositorio adjunto en el Apéndice A.4, junto con una guía de los distintos módulos que lo componen.

Modelado de Sensores

Similar a cómo se procedió en la Sección 4.1, los nuevos sensores incorporados al sistema fueron configurados. Para el caso de las cámaras estéreo se tomó en cuenta la distancia ente los ejes ópticos según se acota en el Anexo B.2. También se extrajo de la misma hoja de datos, los valores para el tamaño del sensor y la cantidad de cuadros por segundo.

Lo parámetros de distorsión del lente, según el modelo *pinhole*, no son anotados por el fabricante. En todo caso, conocer dichos valores no aportaría mayor realismo a la simulación, ya que el algoritmo de localización tiene cómo entradas estos coeficientes para corregir la imagen. Al conocer el valor exacto, se realizaría una corrección perfecta, lo que es equivalente a tener una imagen sin ningún tipo de distorsión. Es importante recalcar que no conocer estos valores de antemano no sería un impedimento para el uso del sistema diseñado en un contexto real, ya que dichos parámetros se pueden estimar con alta exactitud siguiendo el método de calibración propuesto en [53] o cualquier otro método similar. Para efectos de la simulación, todos los parámetros de distorsión fueron ajustados en cero.

Adicionalmente, RotorS incluye como parámetro de entrada para las cámaras, el ángulo de visión horizontal h_{FOV} . En la hoja de datos del Anexo B.2 se reporta el ángulo de visión diagonal θ , según se indica en la Figura 4.10. Tras una manipulación geométrica básica, se determinó que el ángulo de visión horizontal es de 156.77° . Los principales parámetros de modelado para las cámaras estereo son:

Código 4.2: Parámetros de simulación de las cámaras estereo

```

1 <!-- macro to add stereo camera -->
2 <xacro:vi_sensor_stereo_camera_macro
3   ...
4   distortionK1="0.0"
5   distortionK2="0.0"
6   distortionK3="0.0"
7   distortionT1="0.0"
8   distortionT2="0.0"
9   baseline_y="0.064"
10  frame_rate="30.0"
11  image_width="848"
12  image_height="800"
13  horizontal_fov="2.973818069"
14  ...
15 </xacro:vi_sensor_stereo_camera_macro>

```

En relación al módulo Matek AP M9N-F4, se consideraron los valores de ruido indicados en las hojas de datos del GNSS y el barómetro, Anexos B.4 y B.5 respectivamente.

La precisión del GNSS se acota en 2.0 m CEP. Este tipo de indicación, comúnmente utilizada en aplicaciones militares, quiere decir que un 50 % de las lecturas arrojadas por el sensor, caerán dentro de un círculo de radio 2.0

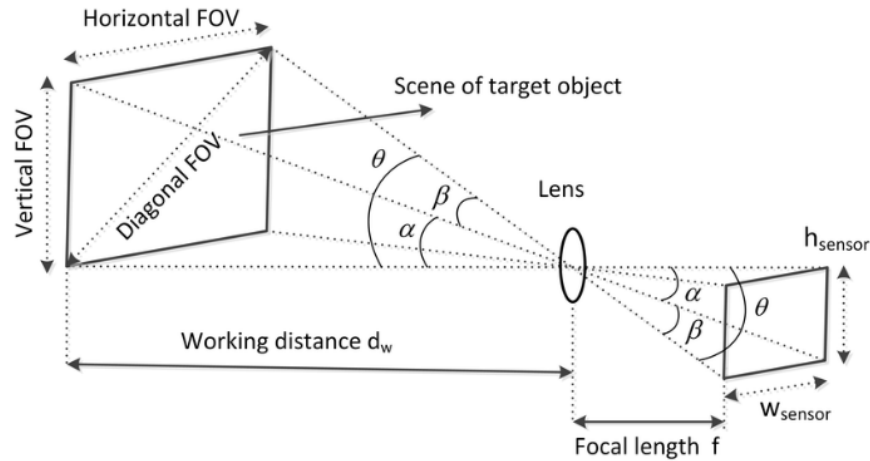


Figura 4.10: Relación de los ángulos de visión

metros centrado en la ubicación real. Para poder simular el ruido, se debe acotar la desviación estándar horizontal. La relación entre ambos conceptos sigue:

$$\sigma = \frac{\text{CEP}}{\sqrt{2 \ln(2)}} \approx 1.6986 \text{ m} \quad (4.10)$$

Para la desviación estándar vertical, se reporta el valor directamente de la hoja de datos. Se incluye la tasa de refrescamiento impuesta por el módulo GNSS, ya que es la menor entre ambos sensores. Finalmente se indica la masa del Matek AP M9N-F4 tomando en cuenta el cálculo presentado en la Sección 4.2.2. Los parámetros relevantes de simulación para el módulo siguen:

Código 4.3: Parámetros de simulación del módulo GNSS

```

1 <!-- Matek AP M9N-F4 -->
2 <xacro:gps_plugin_macro
3   ...
4   mass_gps_sensor="0.0579"
5   horizontal_pos_std_dev="1.6986"
6   vertical_pos_std_dev="0.5"
7   update_rate="25"
8   ...
9 </xacro:gps_plugin_macro>

```

Parámetros de Entrada

Existen múltiples parámetros de entrada para el algoritmo de localización. Algunos están sujetos a experimentación para lograr el mejor desempeño posible: dicha tarea corresponde al Capítulo 5. En este apartado se discuten aquellos parámetros de entrada que son consecuencia del hardware empleado y por tanto, son fijos.

Se comenzará por mencionar los parámetros intrínsecos de las cámaras. Si bien algunos de estos ya se han discutido, el sistema requiere también una estimación de la matriz de la cámara, según se definió en la Sección 2.5. Las coordenadas del centro óptico, al tratarse de un modelo *pinhole* ideal, se ubican justo al centro de la imagen. Con un ancho en píxeles w y un alto h se tiene:

$$c_x = \frac{w}{2} \quad (4.11)$$

$$c_y = \frac{h}{2} \quad (4.12)$$

Por otra parte, la distancia focal se puede computar fácilmente a partir del ángulo de visión horizontal cómo:

$$f = \frac{w}{2 \cdot \tan(\frac{h_{FOV}}{2})} \quad (4.13)$$

Cuando se tienen píxeles perfectamente cuadrados, las componentes f_x y f_y son iguales, por lo tanto:

$$f = f_x = f_y \quad (4.14)$$

De esta manera la configuración de ambas cámaras se muestra en el Código 4.4. En una aplicación real, estos parámetros se determinan empíricamente siguiendo el método propuesto en [53].

Código 4.4: Parámetros intrínsecos de las cámaras

```

1 model_type: PINHOLE
2 camera_name: camera
3 image_width: 848
4 image_height: 800
5 distortion_parameters:
6     k1: 0.0
7     ...

```

```

8 projection_parameters:
9   fx: 87.15
10  fy: 87.15
11  cx: 424.0
12  cy: 400.0

```

Otro aspecto de suma importancia es la correcta determinación de los parámetros extrínsecos en el sistema IMU-Cámaras. Estos se pueden estimar en aplicaciones reales mediante algún método de calibración o midiendo las distancias entre componentes con ayuda de las hojas de datos. Para la simulación se conocen los valores exactos, la convención utilizada es ingresar las matrices de transformación de los ejes ópticos de cada cámara, desde la perspectiva del IMU. Para ejemplificar este concepto, en la Figura 4.11 se puede apreciar la posición relativa de las tramas del IMU y el eje óptico de la cámara izquierda. Las direcciones x , y y z se indican en rojo, verde y azul respectivamente. En el Código 4.5 se indica la transformación correspondiente.

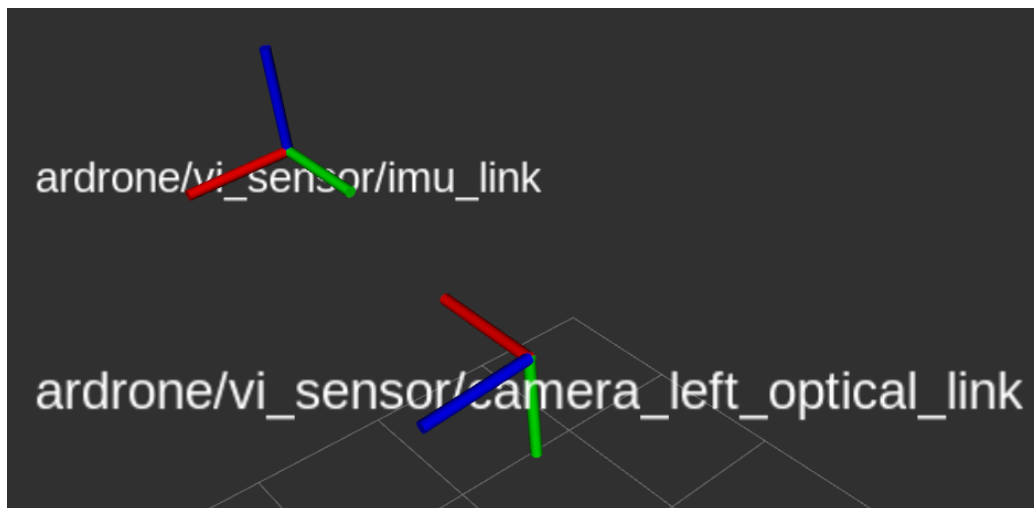


Figura 4.11: Tramas de referencia

Código 4.5: Parámetros extrínsecos

```

1 # left camera to IMU
2 body_T_cam0: !!opencv-matrix
3   rows: 4
4   cols: 4

```

```
5 dt: d
6 data: [ 0, 0, 1, 0.0,
7         -1, 0, 0, 0.032,
8         0, -1, 0, -0.005,
9         0, 0, 0, 1 ]
```

4.4. Escenario de Pruebas

Tras emplear la metodología de diseño, se concluyó que Blender era la mejor plataforma para la creación del escenario de validaciones. En esta sección se describe el flujo de trabajo seguido para la creación de un bosque que refleja una situación de riesgo por incendio.

Las dimensiones requeridas por el cliente fueron de 10 km^2 , por lo que el escenario base es de 100 m de lado. Se empleó la herramienta de generación de terreno de Blender, que permite la creación de un terreno irregular en función de una serie de parámetros de ruido. La malla resultante se muestra en la Figura 4.12.

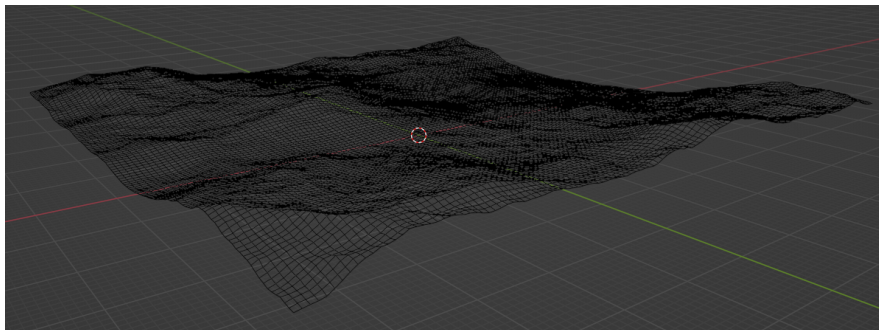


Figura 4.12: Estructura base del terreno

Debido a que este objeto se configura en un espacio tridimensional, para aplicar una textura al modelo se debe realizar el proceso de mapeo UV, que consiste en proyectar una imagen bidimensional a la superficie de un modelo 3D. Blender facilita realizar este proceso de forma automática al analizar la geometría del cuerpo de interés. En las Figuras 4.13a y 4.13b se muestra el desdoblamiento de la textura y su respectiva aplicación. Se debe aclarar que la textura fue diseñada manualmente a partir de la combinación de otras texturas más simples. La intención del diseño fue representar una sección de bosque que ha sido afectada por un incendio reciente.

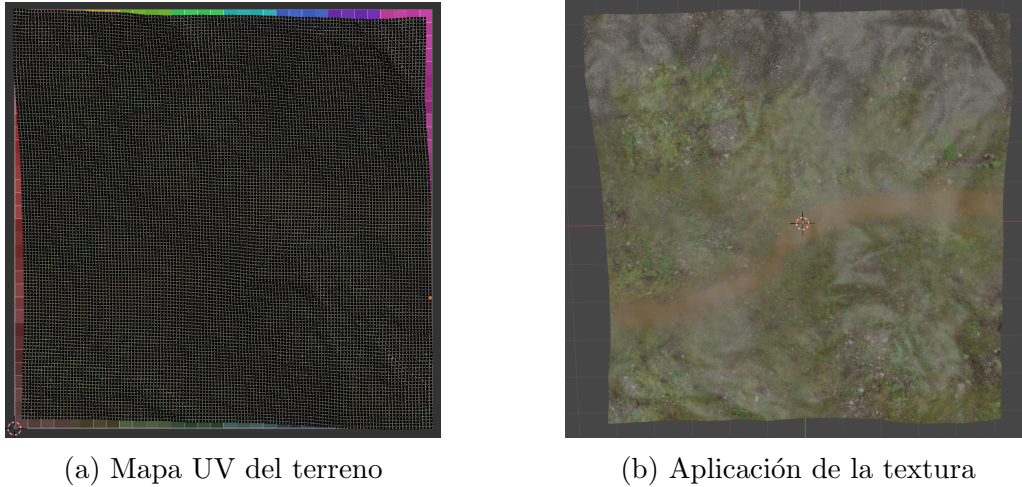


Figura 4.13: Diseño visual del terreno

El siguiente paso fue popular la escena con los modelos 3D requeridos. Todos los modelos empleados son de uso libre y no requieren atribución. Algunas texturas fueron modificadas en función de la intencionalidad de la escena. Un concepto importante que se tomó en cuenta durante esta etapa fue la del poligonaje de los modelos incorporados, que se puede entender como la resolución de un modelo en tres dimensiones. En la primera iteración del diseño, se emplearon modelos de árboles donde cada rama y hoja formaba un cuerpo individual. Dichos modelos resultaron particularmente inadecuados para la simulación de Gazebo, ya que el factor de tiempo real, según se define en la Ecuación 4.15, cayó hasta 0.68.

$$\text{factor de tiempo real} = \frac{\text{tiempo transcurrido en simulación}}{\text{tiempo transcurrido en la realidad}} \quad (4.15)$$

La solución para dicho inconveniente radica en emplear modelos con una geometría sencilla, pero que emplean texturas con una componente alpha. Tradicionalmente, el color de un píxel puede ser representado por un vector RGB, para las componentes rojo, verde y azul. Esta descripción puede ser ampliada agregando un canal adicional, alpha, que representa la opacidad del píxel. Un valor $\alpha = 1$ se considera un píxel completamente opaco, mientras que un valor $\alpha = 0$ vuelve al píxel completamente transparente. Para ejemplificar de mejor manera este concepto, en la Figura 4.14 se muestra el

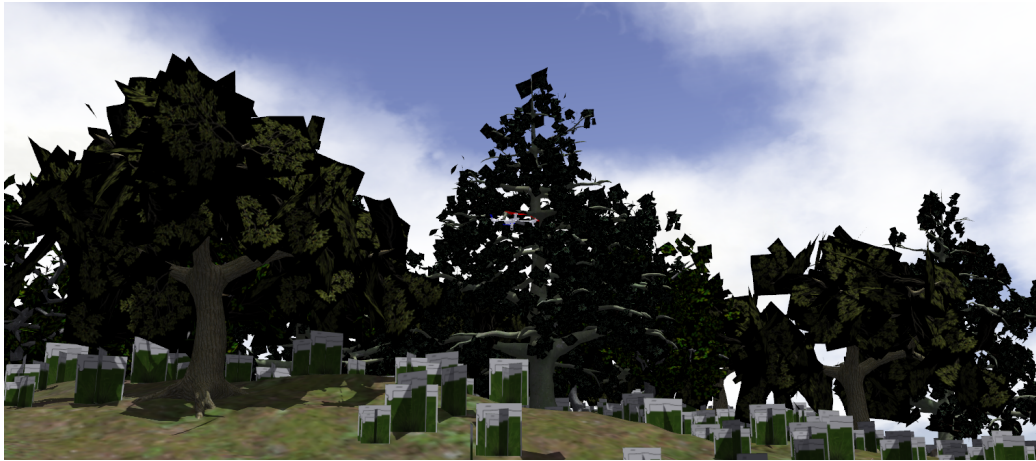
modelo de césped utilizado en el escenario. Como puede apreciarse, el modelo está compuesto únicamente por 6 mallas rectangulares que se intersecan entre sí; no obstante, el modelo aparenta tener una geometría más compleja por el uso de texturas con un canal alpha.



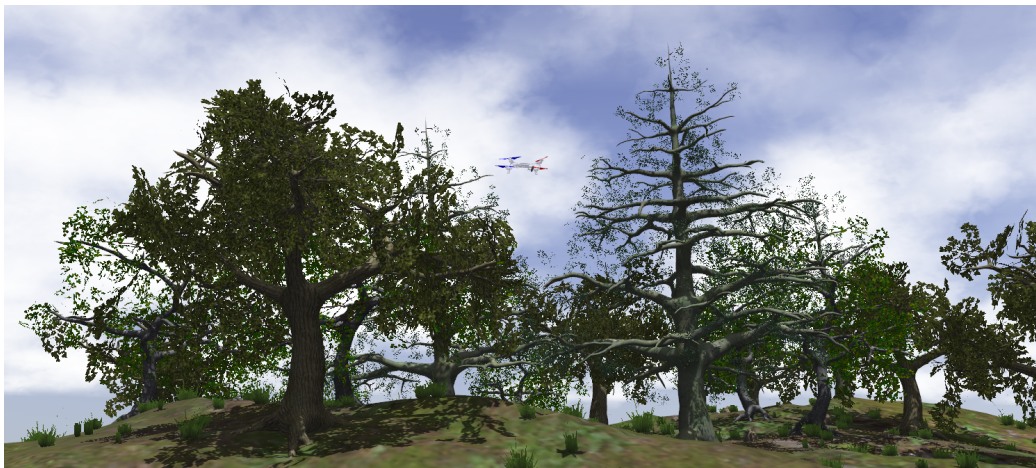
Figura 4.14: Modelo de pasto utilizado

Se emplearon modelos de bajo poligonaje para la inclusión de rocas, dos torres de transmisión de potencia y tres especies de árboles, con una variante de texturas para simular los efectos de un incendio. Además se añadió vegetación empleando el concepto del canal alpha para las texturas.

Cómo comentario final, se debe mencionar que Gazebo no soporta de manera nativa modelos que usen texturas con transparencia. Esta se visualiza con un color negro y la sombra proyectada por el modelo corresponde a la de toda la geometría, no únicamente a las partes que se supone son visibles. Por esta razón se trabajó un script de Ogre, que es el motor de renderizado sobre el que opera Gazebo [54], para corregir este inconveniente. La comparativa entre el antes y el después de la aplicación de dicho script se muestra en la Figura 4.15. El factor de tiempo real del mapa resultante es de 1.00, lo que confirma que el uso de modelos con bajo poligonaje aumenta considerablemente el rendimiento.



(a) Antes



(b) Después

Figura 4.15: Corrección del renderizado de texturas

CAPÍTULO 5

RESULTADOS Y ANÁLISIS

5.1. Prueba de Concepto

En este apartado se indican las pruebas realizadas para la verificación del concepto de solución elegido. Como se comentó en el Capítulo 4, el algoritmo de fusión de señales tiene distintos parámetros de entrada, algunos se definen de forma directa ya que son dependientes del hardware utilizado. Los restantes constituyen factores para la experimentación y determinarán, en última instancia, las especificaciones finales del sistema diseñado. Estos factores se dividen en categorías:

- Condiciones de uso:
 - Trayectoria: un algoritmo de odometría visual, evidentemente se puede ver afectado por el entorno en el que se utiliza. El despliegue de la aeronave en un espacio que tenga pocas características identificables, podría repercutir en el desempeño.
- Rastreo de características en la imagen:
 - Máxima cantidad de características: durante la ejecución se detectan una serie de puntos sobre la imagen que corresponden a los elementos distintivos. Un ejemplo se muestra en la Figura 5.1. El sistema rastrea el movimiento de todos estos puntos para aproximar el cambio en la posición.

- Distancia mínima entre características: como el nombre sugiere, es la distancia mínima que debe existir entre dos características detectadas.
- Flujo retroactivo: la revisión de fotogramas anteriores para mejorar la capacidad de rastreo.
- Parámetros de optimización:
 - Cantidad de iteraciones: restringe la cantidad de iteraciones en los métodos numéricos de optimización para garantizar un desempeño en tiempo real.
 - Tiempo de iteración: tiempo máximo que puede tardar una iteración en el proceso de optimización.
 - Paralaje de fotograma: controla el impacto del efecto de paralaje, que ocurre con la diferencia entre las perspectivas de las cámaras.
- Parámetros de sincronización:
 - Estimación de retardo: el algoritmo intentará estimar el desfase temporal entre los datos que vienen desde las cámaras respecto a la unidad de medición inercial.

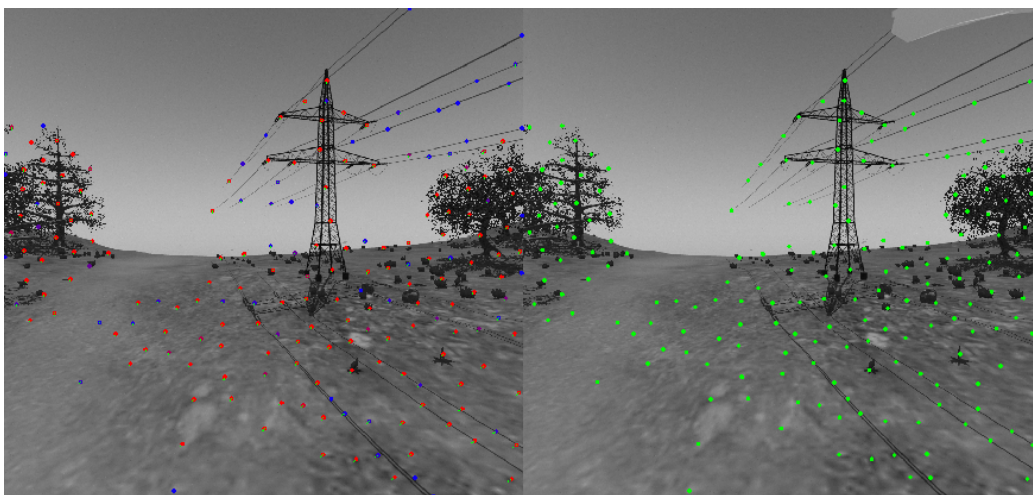


Figura 5.1: Características detectadas en la imagen

Debido a que existen ocho factores, un análisis factorial completo para estimar la importancia de todos los factores primarios y sus interacciones, con dos niveles, sería necesario ejecutar 256 pruebas, una para cada tratamiento:

$$2^8 = 256 \quad (5.1)$$

Debido a que la cantidad de tratamientos es demasiado extensa, se propone una estrategia inicial para determinar los factores importantes.

5.1.1. Diseño de Cribado

La primera fase de experimentación consiste en un diseño de cribado para discernir cuales son los factores con mayor impacto en el desempeño del sistema. Para esta fase inicial se determina únicamente la importancia de los efectos principales y todas las interacciones entre dos factores. Se emplean dos niveles por factor. A excepción de las trayectorias, los valores de cada nivel se han seleccionado en base a desviaciones respecto a la configuración por defecto de cada parámetro de entrada. En el Cuadro 5.1, se muestran los niveles para la fase inicial de cribado. La ruta A corresponde a una trayectoria con mayor acceso a elementos distintivos capturados por las cámaras; la ruta B se ejecuta reduciendo esta disponibilidad.

Cuadro 5.1: Factores para el diseño de cribado

Factor	Tipo	Unidades	Límite Inferior	Límite Superior
Trayectoria	Catagórico	N/A	B	A
Cantidad C.	Discreto	N/A	150	200
Distancia C.	Discreto	Píxeles	10	30
Retroactivo	Catagórico	N/A	No	Sí
Cantidad I.	Discreto	N/A	6	12
Tiempo I.	Continuo	Segundos	0.02	0.10
Paralaje	Discreto	Píxeles	10	20
Sincronización	Catagórico	N/A	No	Sí

Para este diseño, se empleó un criterio de optimización D, que tiene cómo propósito minimizar el determinante de la matriz de covarianza de las estimaciones del coeficiente del modelo. Este criterio es especialmente aconsejado en diseños de cribado [55]. De esta manera, el total de pruebas necesarias para

CAPÍTULO 5. RESULTADOS Y ANÁLISIS

estimar la importancia de los 8 factores principales, y las 28 interacciones de segundo nivel, se reduce a 37 pruebas.

Para repetir de forma exacta las condiciones entre experimentos, los datos de cada sensor en las trayectorias seguidas fueron almacenados para su posterior reproducción y uso en el sistema. La grabación de las rutas A y B se puede consultar en el Apéndice A.3, junto con todos los datos extraídos en cada experimento. En el Cuadro 5.2 se resumen los resultados obtenidos.

Cuadro 5.2: Resultados del diseño de criba

Tray.	Can. C.	Dis. C. (px)	Ret.	Can. I.	Tie. I. (s)	Par. (px)	Sinc.	Error (m)
A	200	30	Sí	12	0.1	10	Sí	0.26394
B	150	10	No	12	0.1	10	No	0.36029
B	200	10	Sí	12	0.1	20	Sí	0.3471
A	200	10	No	12	0.1	20	No	0.32526
A	200	30	No	12	0.02	10	Sí	0.2591
A	150	30	Sí	6	0.1	10	Sí	0.27056
A	150	10	No	6	0.02	20	Sí	0.34731
B	150	30	No	12	0.02	10	No	0.31171
B	150	30	No	12	0.1	20	Sí	0.34316
B	150	10	No	6	0.1	20	No	0.36895
B	200	30	Sí	12	0.1	20	No	0.34884
B	150	10	Sí	12	0.02	20	No	0.32517
B	200	10	Sí	6	0.02	10	No	0.31946
A	200	10	Sí	6	0.02	20	No	0.3207
A	200	10	Sí	12	0.02	10	No	0.3199
A	200	10	No	6	0.02	10	No	0.28828
A	150	30	No	12	0.1	10	No	0.26345
A	150	10	Sí	6	0.02	10	No	0.28423
B	200	30	Sí	6	0.1	10	Sí	0.29418
A	200	30	Sí	12	0.02	20	Sí	0.27907
B	200	30	No	6	0.02	20	Sí	0.32344
B	150	30	Sí	6	0.02	20	Sí	0.28746
B	200	10	No	6	0.1	10	Sí	0.34176
B	150	10	Sí	6	0.1	10	Sí	0.33831
A	150	10	No	12	0.1	10	Sí	0.29897

A	150	30	No	6	0.02	20	No	0.30242
A	150	10	Sí	12	0.1	20	No	0.31368
B	200	30	No	6	0.1	10	No	0.33668
B	150	30	Sí	6	0.1	10	No	0.30634
B	150	10	No	12	0.02	10	Sí	0.32064
A	200	10	Sí	6	0.02	10	Sí	0.28628
A	200	30	No	6	0.1	20	Sí	0.26051
A	150	30	Sí	12	0.02	10	Sí	0.26531
A	200	10	Sí	6	0.1	10	No	0.30504
A	200	30	Sí	6	0.02	10	No	0.26897
B	150	30	No	6	0.02	10	Sí	0.25621
B	200	30	Sí	12	0.02	10	Sí	0.28087

Tras remover de forma jerárquica los factores no significativos, en la Figura 5.2 se muestran aquellos que prevalecen, cuatro factores principales y una interacción de segundo nivel. Si bien el factor de tiempo de iteración no es significativo por sí mismo, su interacción con el tipo de trayectoria restringe su consideración.

Fuente	ValorLog		ValorP
Trayectoria	6.564		0.00000
Distancia C.(10,30)	6.528		0.00000
Paralaje(10,20)	3.711		0.00019
Trayectoria*Tiempo I.	2.782		0.00165
Tiempo I.(0.02,0.1)	1.922		0.01197 ^

Figura 5.2: Factores significativos del diseño de criba

5.2. Diseño de Superficie de Respuesta

Con la ayuda del diseño de cribado se han obtenido los efectos de importancia que influyen en el error de estimación. De esta manera se puede proceder a un estudio más detallado con el objetivo de mejorar las estimaciones. En el análisis preliminar, los valores ideales de optimización se muestran en la Figura 5.3

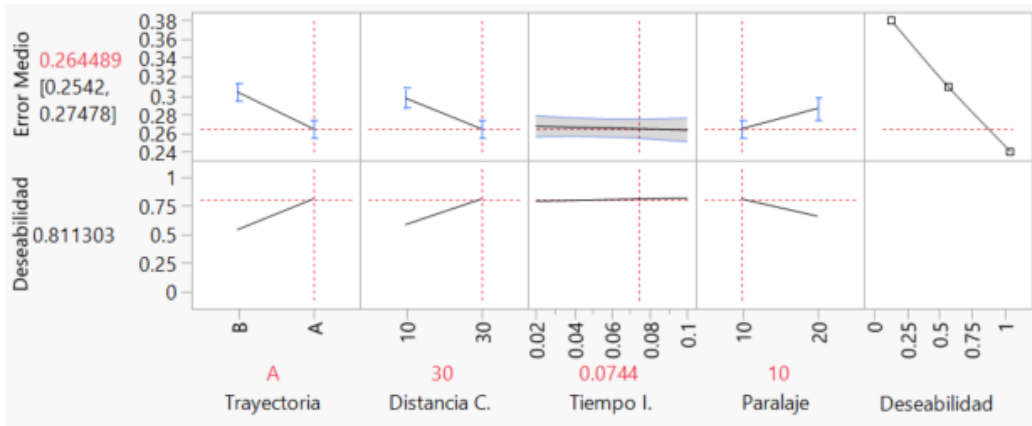


Figura 5.3: Optimización para el diseño de cribado

Para estimar los efectos de orden cuadrático que podrían ejercer influencia en la respuesta del sistema, se eligió emplear un diseño central compuesto. Este tipo de diseños son ampliamente utilizados en los análisis de superficie de respuesta, ya que emplean puntos centrales y axiales para estimar eficientemente términos segundo orden y modelar una variable de respuesta con curvatura [56].

Debido a la necesidad de incorporar puntos centrales al nuevo diseño, los niveles se reestructuraron de la siguiente manera para explorar de forma más amplia el espacio de solución:

- El límite superior para la distancia entre características se eleva a 50, por lo que el anterior límite superior ahora es el punto central.
- Los valores para el tiempo de iteración se mantienen, ya que según se muestra en la figura 5.3, el valor óptimo se encuentra en el medio.
- El límite inferior del paralelo se reduce a 2, por lo que el límite central es ahora 11.

En el Cuadro 5.3 se resumen los niveles de experimentación.

Para esta segunda fase, se emplea un criterio de optimización I, que es particularmente aconsejado en experimentos cuya intención es crear un modelo para predecir la respuesta del sistema, y cuando se desea conocer las condiciones óptimas de operación [55]. Bajo este enfoque, y empleando sólo

Cuadro 5.3: Factores para el diseño de superficie de respuesta

Factor	Tipo	Unidades	L. Inferior	V. Central	L. Superior
Distancia C.	Discreto	Píxeles	10	30	50
Tiempo I.	Continuo	Segundos	0.02	0.06	0.10
Paralaje	Discreto	Píxeles	2	11	20

la ruta A, es necesario realizar 16 pruebas para determinar los factores principales, las interacciones de segundo nivel y los términos cuadráticos. En el Cuadro 5.4 se resumen los resultados obtenidos.

Cuadro 5.4: Resultados del diseño de superficie de respuesta

Patrón	Distancia C. (px)	Tiempo I. (s)	Paralaje (px)	Error (m)
0 0 +	30	0.06	20	0.29576
0 0 -	30	0.06	2	0.28776
- 0 0	10	0.06	11	0.31394
+ + +	50	0.1	20	0.30037
+ 0 0	50	0.06	11	0.28188
0 + 0	30	0.1	11	0.28504
0 - 0	30	0.02	11	0.29372
+ + -	50	0.1	2	0.29566
- + -	10	0.1	2	0.30153
+ - +	50	0.02	20	0.29594
0 0 0	30	0.06	11	0.28569
0 0 0	30	0.06	11	0.28521
- - +	10	0.02	20	0.33329
- - -	10	0.02	2	0.29001
+ - -	50	0.02	2	0.29256
- + +	10	0.1	20	0.33187

Tras una reducción jerárquica de los términos no significativos, el resultado del modelo se muestra en la Figura 5.4. Cómo era de esperar, el factor de tiempo de iteración desaparece ya que estaba vinculado con el tipo de trayectoria. La implicación evidente es que mientras menos elementos característicos estén presentes en la imagen, el sistema se puede beneficiar de un mayor tiempo de iteración del algoritmo.

Fuente	ValorLog		ValorP
Distancia C.(10,50)	3.778		0.00017
Distancia C.*Distancia C.	3.299		0.00050
Paralaje(2,20)	3.255		0.00056
Distancia C.*Paralaje	2.619		0.00240

Figura 5.4: Factores significativos del diseño de superficie de respuesta

Sobre el ajuste del modelo, se puede analizar el gráfico de error de estimación real respecto al predicho, que se muestra en la Figura 5.5. Cómo puede observarse, el error cuadrático medio es bastante reducido, y el intervalo de confianza denotado por la líneas rojas, es estrecho.

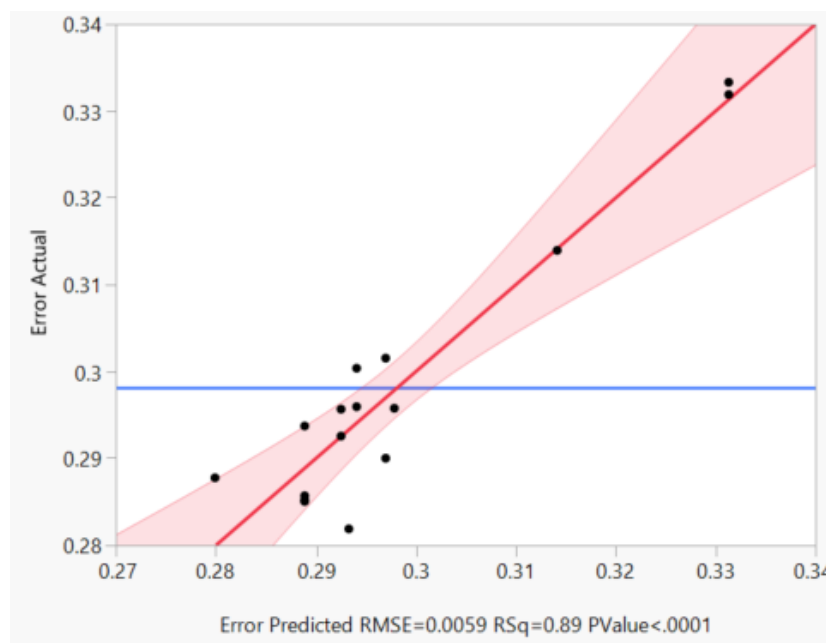


Figura 5.5: Error real respecto a error predicho

Además, en la Figura 5.6, el gráfico de residuales (la diferencia entre el valor predicho y el valor experimental del error) bajo la distribución t-student muestra que no existe ninguna violación al modelo ni valores atípicos, ya que todos los puntos se encuentran dentro de la región delimitada por las líneas rojas.

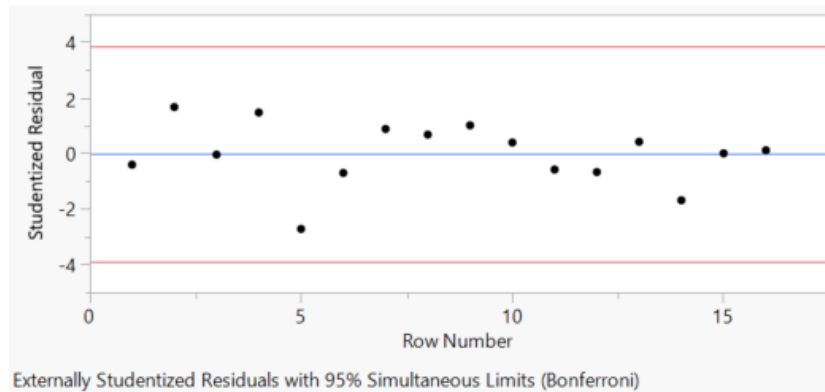


Figura 5.6: Residuales bajo la distribución t-student

Los factores de paralaje y distancia entre características forman una superficie que tiene por respuesta el error de estimación. Dicha superficie se puede observar en la Figura 5.7. De esta manera se comprueba que la evaluación de términos cuadráticos permitió desarrollar un modelo más preciso para realizar la optimización de los parámetros de entrada al algoritmo.

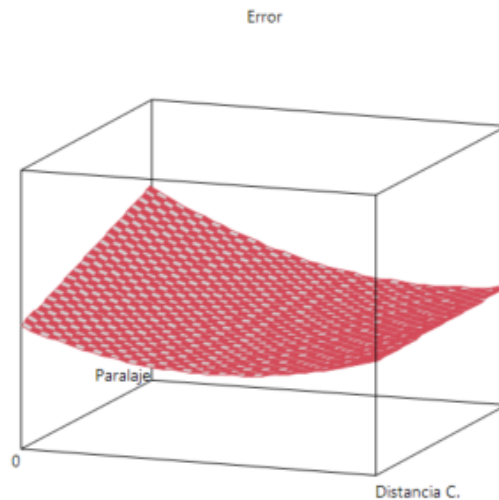


Figura 5.7: Superficie formada por la interacción entre el paralaje y la distancia entre características

Finalmente, en la Figura 5.8 se muestra la optimización de factores realizada por el modelo. Se debe notar que para el caso de la distancia entre

características, al tratarse de una variable discreta, se debe redondear al entero más próximo.

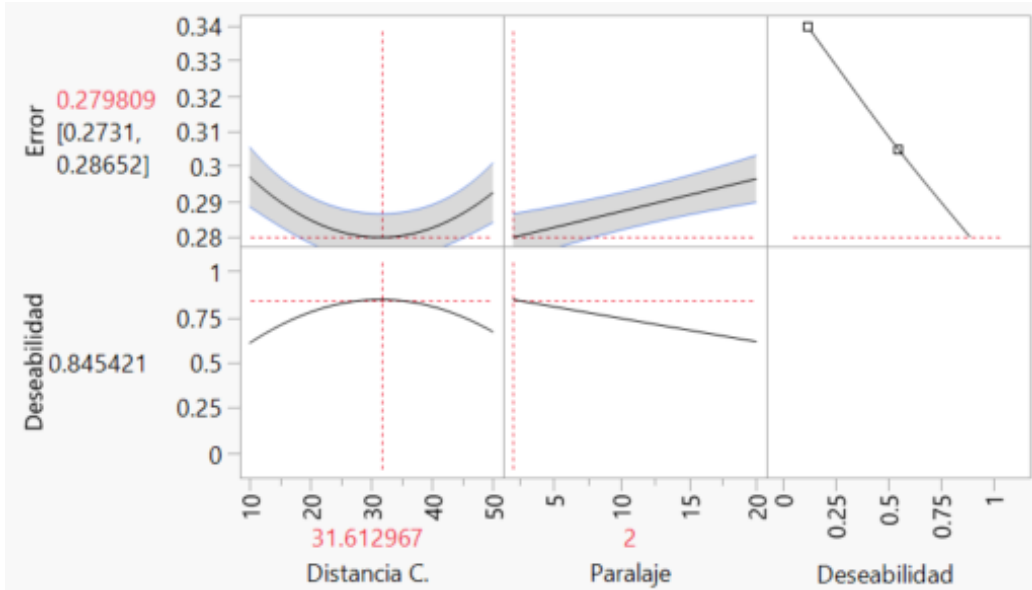


Figura 5.8: Optimización para el diseño de superficie de respuesta

La acotación final del error medio de localización para el sistema diseñado es:

$$\text{error}_{SD} = 0.27981 \pm 0.00671 \text{ m} \quad (5.2)$$

Tras repetir el experimento con los valores óptimos en la ruta A, se confirma que el valor de error se encuentra dentro del rango predicho por el modelo. Una visualización de la ruta estimada y la real se muestra en la Figura 5.9.

5.3. Experimentos Complementarios

5.3.1. Corrección de la Deriva

En este experimento se piloteó la aeronave de forma aleatoria durante 10 minutos. El propósito fundamental es verificar que, en base al hardware incluido en la aeronave, el fenómeno de la deriva se corrige de forma

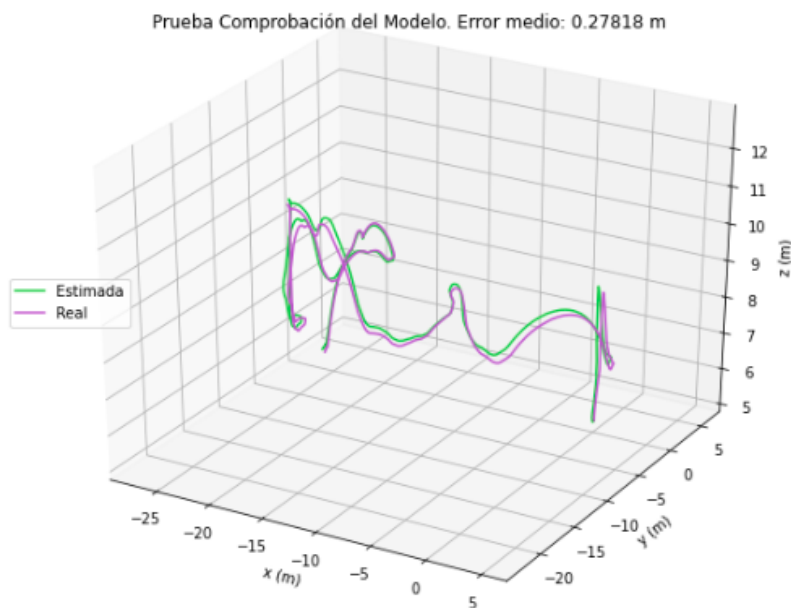


Figura 5.9: Comprobación del error medio predicho por el modelo con los parámetros optimizados

satisfactoria. Se procuró volar la aeronave bajo las condiciones ideales de características identificables. El error medio obtenido es de 0.28438 metros. La trayectoria seguida se muestra en la Figura 5.10.

5.3.2. Condiciones de Poca Luz

Cómo ya se comprobó en la Sección 5.1.1, debido a que el sistema emplea la información captada por las cámaras para ejecutar la localización, una falta de características visibles en la imagen puede comprometer el desempeño. Se empleó el escenario de validaciones en condiciones nocturnas, según se muestra en la Figura 5.11 para determinar el detrimento en la estimación. Los resultados obtenidos se muestran en la Figura 5.12.

Considerando el error medio de 0.3209 metros obtenido, es posible afirmar que operar la aeronave en condiciones de poca luz tiene un efecto comparable a su uso en entornos con características poco destacables.

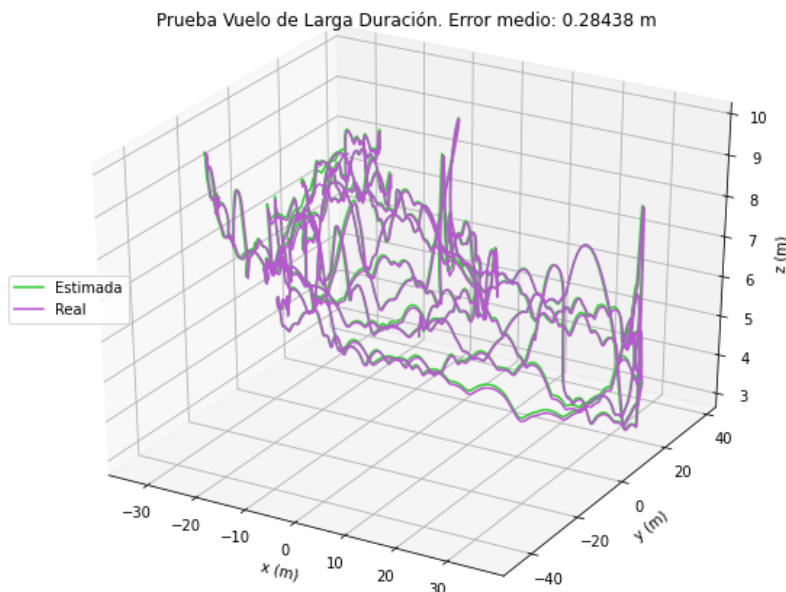


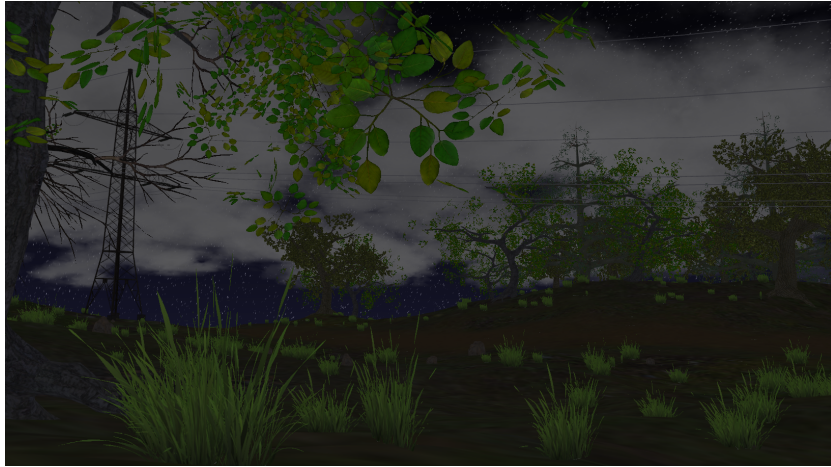
Figura 5.10: Desempeño del sistema a largo plazo

5.3.3. Navegación sin Datos Geoespaciales

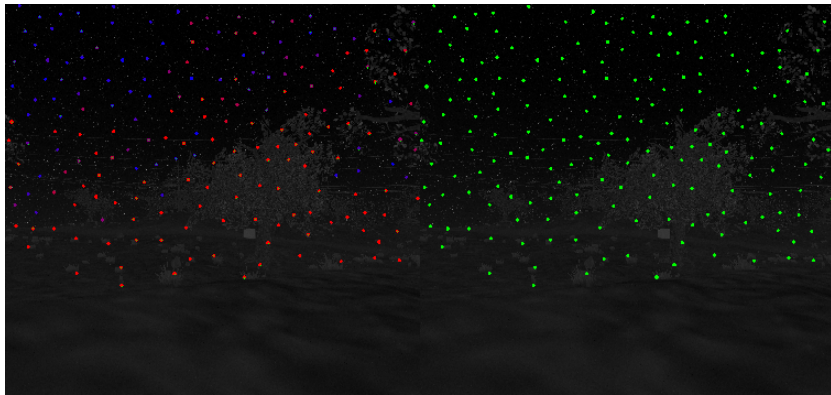
La recolección de datos geoespaciales por parte de la aeronave juega un rol fundamental para corregir la trayectoria estimada. Si esta es incapaz de obtener información georeferenciada, el sistema diseñado estimará la posición únicamente a partir de los sensores locales. El hecho de involucrar las cámaras disponibles en la aeronave, fue una decisión de diseño alentada por la necesidad de obtener una estimación más robusta, por lo que es importante comprobar el desempeño bajo estas circunstancias.

Los datos de la Figura 5.13 corresponden al desempeño de la aeronave sin señal del módulo GNSS.

Aunque en principio podría parecer que la omisión de los datos geoespaciales no tiene un efecto considerable en el desempeño, se debe recordar que cualquier aplicación que sólo utilice información local, adolecerá los efectos de la deriva. Por esta razón, se analizó la evolución del error en el tiempo, para obtener un estimado del desempeño esperable en condiciones donde la señal del GNSS es débil. Además se realiza una comparativa con el estado actual del sistema de localización, empleando los datos de la Sección 4.1. Los



(a) Perspectiva general



(b) Perspectiva de la aeronave

Figura 5.11: Condiciones experimentales de poca luz

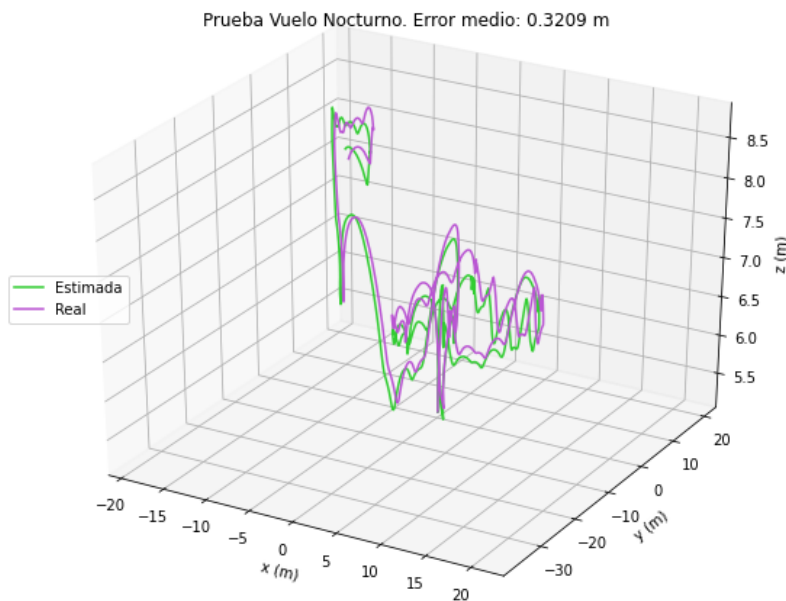


Figura 5.12: Desempeño del sistema en condiciones de poca luz

resultados obtenidos se muestran en la Figura 5.14.

Es posible afirmar que el error de estimación no sólo muestra un comportamiento más estable, sino también que su proyección lineal tiene una pendiente menor en contraste con el estado actual del sistema de localización.

5.3.4. Navegación con el Módulo Matek AP M9N-F4

Las opciones comerciales consultadas en el Cuadro 4.1 corresponden a módulos de localización dentro del rango de costos presupuestado. Dado que los datos que brindan están georeferenciados, se pueden emplear en cualquier contexto siempre y cuando se tenga señales satelitales.

Se plantea un experimento en la ruta A empleando las características del módulo Matek AP M9N-F4, que fue la opción elegida en el proyecto. Se omite el procesamiento y fusión con los datos del IMU o las cámaras. Los resultados obtenidos se muestran en la Figura 5.15.

Cómo se puede observar, si bien este módulo permite realizar localización libre de deriva, el error medio de estimación pasa a estar en el rango de 2.24

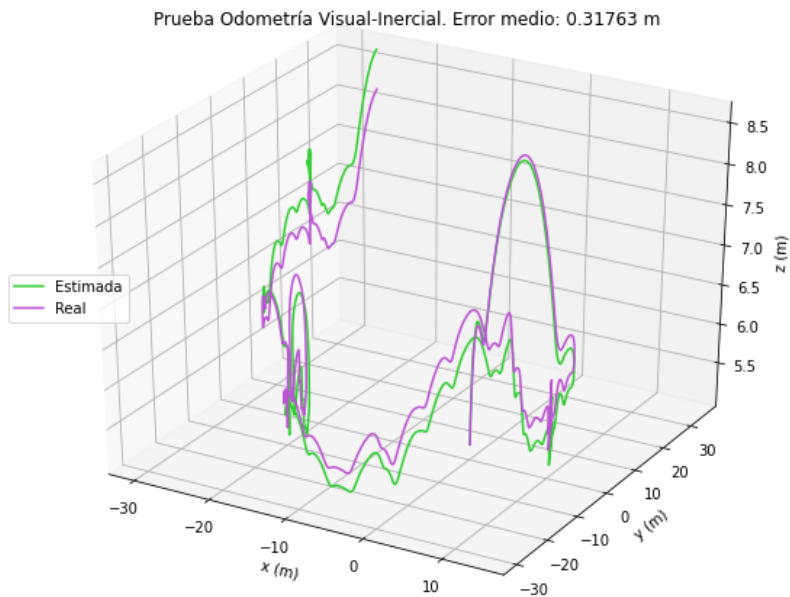


Figura 5.13: Desempeño del sistema sin señal del módulo GNSS

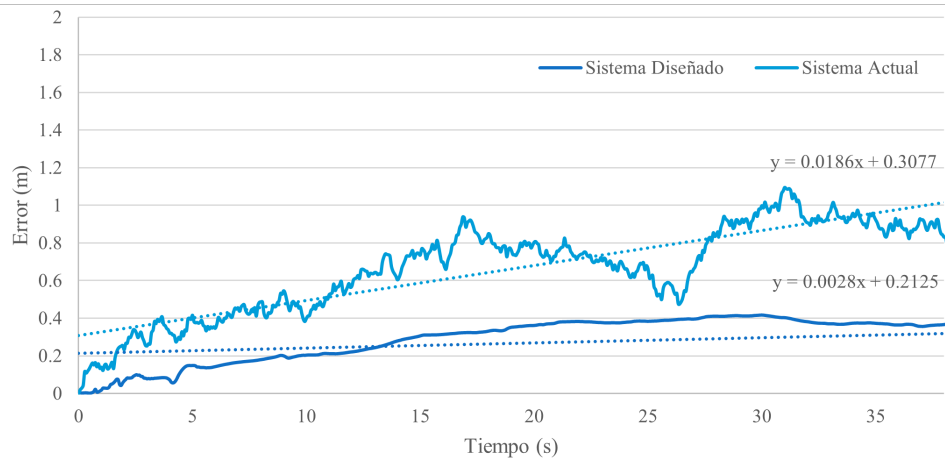


Figura 5.14: Comparativa entre estimación con sensores locales

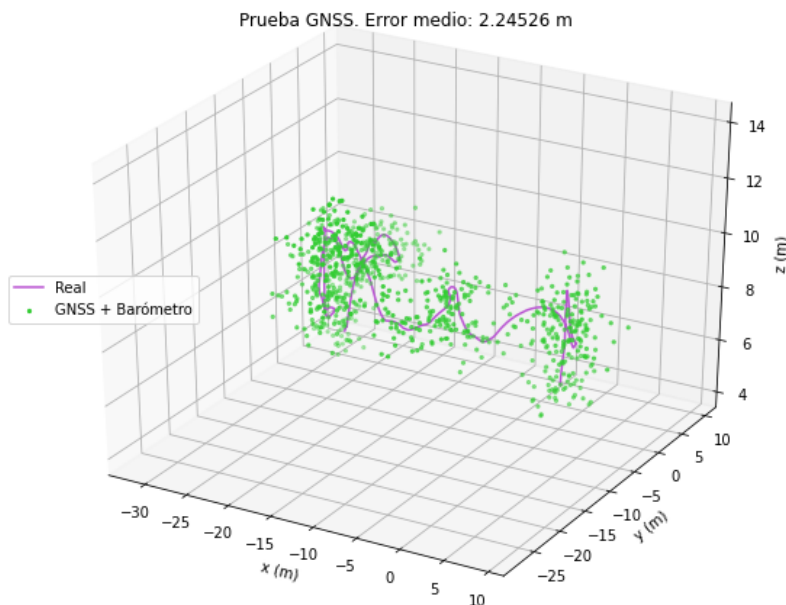


Figura 5.15: Desempeño del sistema empleando datos geospaciales

metros, en contraste con el sistema diseñado que se ubica en 28.0 centímetros.

5.4. Análisis Económico

Es evidente que el proyecto se ha realizado dentro de un contexto de investigación, por lo que ofrecer una métrica de retorno de inversión no es factible. No obstante, es posible establecer distintas métricas auxiliares para evaluar el impacto económico del proyecto.

5.4.1. Coste de Implementación

Cómo valor ideal para el coste de implementación, se fijó la cifra en 300 USD. En el Cuadro 5.5 se resumen todos los insumos necesarios para la construcción física del sistema diseñado. Para el caso del cobertor, se tomó en cuenta la cantidad de filamento ABS necesario para la impresión, la cifra indicada se refiere a la fracción de un kilogramo requerida.

De esta manera, el coste de total de la implementación es de 109.17 USD,

Cuadro 5.5: Coste de los elementos físicos del sistema diseñado

Elemento	Cantidad	Coste Unitario (USD)	Cita
Módulo GNSS M9N-F4	1	101.72	[49]
Perno M3 x 0.5 x 25 mm	4	0.0872	[57]
Tuerca M3 x 0.5 mm	4	0.0262	[58]
Cobertor	0.0158	19.70	[59]
Cable Micro-USB 0.5 ft	1	6.69	[60]

lo que representa un ahorro de 190.83 USD por unidad construida frente al presupuesto inicial. No se están contemplando los costos asociados a la mano de obra, el transporte de la materia prima o el hardware ya existente en la aeronave que fue aprovechado, ya que este tipo de gastos no fueron incluidos en la formulación del presupuesto inicial. Con una reducción del 63.61% del presupuesto previsto, el sistema diseñado deja un amplio margen para la implementación de otras mejoras en la aeronave empleando los recursos sobrantes.

Adicionalmente, cómo se evidencia en el Apéndice A.1, el proceso de ensamblaje es relativamente sencillo y no requiere capital humano especializado, por lo que una vez que se tienen todos los insumos, el proceso de construcción no incurre en gastos adicionales.

5.4.2. Comparativa con Soluciones Comerciales

Existen distintas soluciones comerciales que bajo el amparo de hardware especializado, pueden ser utilizadas para ejecutar una localización precisa. Si bien por sus dimensiones o cualidades prácticas algunas no serían aptas para sus uso en las aeronaves de HRC, es menester realizar una comparación de su rendimiento en relación al costo, respecto al sistema diseñado. Para dicho fin se propone el empleo de la siguiente métrica:

$$\text{rendimiento} = \frac{100}{\text{costo} \cdot \text{error de estimación}} \quad (5.3)$$

De esta manera, para aquellas aplicaciones dónde el producto del error de estimación y el costo sea reducido, el rendimiento será alto. En el cálculo de esta métrica para el sistema diseñado, sí se incluye el costo del módulo RealSense T265, ya que es necesario para su funcionamiento. El rendimiento sigue:

$$\text{rendimiento}_{SD} = \frac{100}{(109.17 + 325.46) \text{ USD} \cdot 0.28 \text{ m}} \approx 0.82 \text{ USD}^{-1} \text{ m}^{-1} \quad (5.4)$$

El módulo GNSS Matek AP M9N-F4 usado en el proyecto, fue la mejor opción comercial (dentro del presupuesto previsto) de todas las alternativas consultadas en el Cuadro 4.1. Como comparativa, si se hubiera empleado únicamente este módulo para estimar la posición, sin realizar ningún procesamiento ni fusión de datos, el rendimiento sería:

$$\text{rendimiento}_{M9N-F4} = \frac{100}{(109.17) \text{ USD} \cdot 2.24 \text{ m}} \approx 0.41 \text{ USD}^{-1} \text{ m}^{-1} \quad (5.5)$$

Este cálculo se repitió con varias soluciones comerciales de mayor precisión, que ofrecen datos del error de localización en tres dimensiones. Los resultados se resumen en el Cuadro 5.6.

Cuadro 5.6: Rendimiento de sistemas de localización comerciales

Solución	Costo (USD)	Error (m)	Rendimiento (USD ⁻¹ m ⁻¹)	Cita
Sistema Diseñado	434.63	0.27	0.82	-
Matek AP M9N-F4	109.17	2.24	0.41	[49]
D-RTK 2 + Matrice 300 RTK	3600.00	0.1	0.28	[61]
Juniper Geode GNS3	4845.00	0.08	0.26	[62]
N3 IMU GNSS Receiver	10500.00	0.01	0.95	[63]
Novatel CPT7	11200.00	0.01	0.89	[64]

Cómo puede observarse, si bien todos los sistemas comerciales por encima de los 1,000 USD tienen un mejor desempeño de localización absoluto, la relación costo-desempeño es bastante superior en el sistema diseñado, y se mantiene competitivo contra las alternativas que superan los 10,000 USD.

Previsiblemente dentro del espacio de otros trabajos académicos pueden existir soluciones de localización que, con una cuidadosa selección de hardware y software, igualen o superen el rendimiento del sistema diseñado. No

obstante, se puede afirmar con certeza que desde un punto de vista económico, el proyecto se justifica al no existir soluciones comerciales dentro del rango presupuestado que ofrezcan un mejor desempeño de localización, tanto en términos absolutos como en la relación costo-desempeño.

5.4.3. Entorno de Simulación

Dada la naturaleza visual de la solución implementada, un entorno de simulación abundante en características fue indispensable para la verificación del proyecto. Más allá de este uso aislado, este escenario puede ser usado en una gran cantidad de experimentos distintos, e incluso servir de base para futuras ampliaciones.

Esta opcionalidad, permite al departamento de HRC avanzar en sus investigaciones en entornos naturales que modelen situaciones de riesgo, lo que en última instancia da pie a la generación de artículos científicos y publicaciones: ambos elementos son determinantes en el financiamiento que recibe el departamento.

Si bien no es posible poner una cifra concreta al impacto económico en referencia a la reutilización del mapa, todos los archivos requeridos para su generación están a disposición del departamento, por lo que en el futuro se podrán agregar funcionalidades según sea conveniente. Estos archivos pueden ser consultados en el Apéndice A.3.

5.5. Especificaciones Finales

En apego a la metodología de trabajo, se indican las especificaciones finales del sistema diseñado. Se puede observar que en referencia al Cuadro 3.3, todas las especificaciones se han cumplido de manera ideal a excepción del error medio de localización, que se ha cumplido marginalmente.

Cuadro 5.7: Especificaciones finales

ID	Métrica	Unidades	Valor Final
1	Error medio de localización.	m	0.27981 ± 0.00671
2	Información geoespacial recolectada.	lista	latitud, longitud, altitud
3	Frecuencia de actualización.	Hz	25
4	Porcentaje de integración con IMU.	%	100
5	Disponibilidad de verificación en Gazebo.	binaria	disponible
6	Factor de tiempo real.	%	100
7	Semejanza a situación de riesgo por incendio.	lista	árboles quemados, marcas del terreno, torres de alta tensión
8	Semejanza a entorno natural.	lista	árboles, césped, rocas, terreno irregular
9	Grado de compatibilidad con el procesador Intel Atom.	%	100
10	Grado de compatibilidad con Intel EI.	%	100
11	Coste económico de implementación.	USD	109.17

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Tras finalizar el proyecto, el departamento de HRC cuenta con una propuesta sólida y de bajo costo para el sistema de localización de las aeronaves autónomas que desarrolla. Las principales conclusiones son:

- Se realizó un estudio del estado del arte en cuanto a tecnologías y sistemas de localización en tres dimensiones, lo que garantizó una selección fundamentada de los métodos y soluciones más adecuados para resolver el problema planteado, en concordancia con la metodología de diseño.
- Se diagnosticó con un 95 % de confianza estadística el estado actual del sistema de localización, dónde se logró una caracterización relevante como insumo para el proceso de diseño. Además permitió establecer la validez del sistema diseñado, dónde se demostró una reducción del 58.3 % en el error medio de estimación para una misma ventana temporal.
- Se establecieron los componentes mecánicos y de hardware requeridos para la captación de datos georeferenciados de latitud, longitud y altitud por parte de la aeronave. Con una tasa de recolección de 25 Hz,

estos datos son una pieza clave para corregir el problema de deriva que adolece el sistema actualmente operante.

- Se confeccionó un sistema de software para la fusión de señales de sensores locales y globales. Dicho sistema permitió lograr una estimación precisa de la ubicación de la aeronave y al ser compatible con Intel Edge Insights, la estimación puede integrarse al 100 % con otros sistemas presentes en la misma, lo que garantiza una operación robusta ante fallos de comunicación.
- Se verificó el desempeño del sistema diseñado mediante la elaboración de un entorno de simulación basado en una situación de riesgo real. Dicho entorno es multipropósito y permitirá al departamento realizar experimentos de diversa índole en el futuro. Se demostró que el nivel de confianza de 95 % utilizado en la verificación, fue capaz de predecir de forma correcta el desempeño de la localización en distintas condiciones operativas.
- Se demostró que el sistema diseñado es económicamente competitivo, empleando la relación entre los recursos monetarios invertidos, el desempeño de localización logrado, y comparándolo contra múltiples alternativas comerciales. Los principales hallazgos apuntan a un desempeño superior en términos absolutos para los sistemas comerciales dentro del presupuesto para el proyecto, y una valoración competitiva en la métrica costo-desempeño contra las soluciones comerciales que superan el presupuesto.

6.2. Recomendaciones

Debido a que sólo se concluyó con la primera parte del ciclo de diseño de HRC, existen múltiples recomendaciones a tener en cuenta durante la etapa de construcción física:

- Si bien la mayoría de parámetros ya se encuentran configurados para un rendimiento óptimo con el hardware presente en las aeronaves, se debe mencionar la notable excepción de los parámetros intrínsecos de las cámaras, mencionados en la Sección 4.3.3. Se debe someter a las cámaras a un proceso de calibración para estimar de forma apropiada estos parámetros.

CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

- Como se comentó en la Sección 4.1, no fue posible incluir algunos valores de ruido del IMU ya que no están disponibles por el fabricante. Además, pueden existir algunos fenómenos físicos no contemplados en el simulador que puedan perjudicar el rendimiento, por lo que un montaje experimental de validación es necesario para dar un valor al desempeño del sistema en la realidad.
- En la Sección 5.3 se exploraron algunos factores operativos que pueden incidir en el error de localización. Si bien los entornos reales suelen ser abundantes en características visuales, es de vital importancia que se asegure una adecuada captación de las señales satelitales para el módulo GNSS. El ensamblaje físico se debe realizar evitando cualquier obstrucción en la señal.

BIBLIOGRAFÍA

- [1] Intel Corporation, “Intel’s vision for the future of technology.” intel.la. <https://www.intel.la/content/www/xl/es/company-overview/company-overview.html> (accesado: Jun. 24, 2022).
- [2] Intel Corporation, “Intel Labs Research Areas: Inventing World-Changing Technologies.” intel.com. <https://www.intel.com/content/www/us/en/research/blogs/research-areas.html> (accesado: Jun. 24, 2022).
- [3] Intel Corporation, “Using AI technologies to amplify human potential.” intel.com. <https://www.intel.com/content/www/us/en/company-overview/wonderful/human-ai-technology-research.html> (accesado: Jun. 27, 2022).
- [4] K. Bipin, *Robot Operating System Cookbook*. Birmingham, United Kingdom: Packt Publishing, 2018.
- [5] Open Robotics, “Features and Benefits.” gazebosim.org. <https://gazebosim.org/features> (accesado: Oct. 09, 2022).
- [6] R. Toledo-Moreo, J. M. Armingol, M. Clavijo, A. de la Escalera, J. del Ser, F. Jiménez, B. Musleh, J. E. Naranjo, I. I. Olabarrieta, and J. Sánchez-Cubillo, “Chapter 4 - positioning and digital maps,” in *Intelligent Vehicles* (F. Jiménez, ed.), pp. 141–174, Butterworth-Heinemann, 2018.

-
- [7] A. Civita, S. Fiori, and G. Romani, “A mobile acquisition system and a method for hips sway fluency assessment,” *Information*, vol. 9, p. 321, 12 2018.
- [8] F. E. Casado, D. Lema, R. Iglesias, C. V. Regueiro, and S. Barro, “Concept drift detection and adaptation for robotics and mobile devices in federated and continual settings,” in *Advances in Physical Agents II* (L. M. Bergasa, M. Ocaña, R. Barea, E. López-Guillén, and P. Revenga, eds.), (Cham), pp. 79–93, Springer International Publishing, 2021.
- [9] OpenCV, “Camera Calibration.” `docs.opencv.org`. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (accesado: Oct. 09, 2022).
- [10] A. Hahn, *Operational Technology and Information Technology in Industrial Control Systems*, pp. 51–68. Cham: Springer International Publishing, 2016.
- [11] K. Ulrich and S. Eppinger, *Diseño y desarrollo de productos*. Ciudad de México: McGraw-Hill, 2013.
- [12] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, 2014.
- [13] C. Kerl, J. Sturm, and D. Cremers, “Dense visual slam for rgb-d cameras,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2100–2106, 2013.
- [14] Intel Corporation, “Intel RealSense Technology.” `intel.com`. <https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html> (accesado: Ago. 02, 2022).
- [15] J. Zhang and S. Singh, “Loam : Lidar odometry and mapping in real-time,” *Robotics: Science and Systems Conference (RSS)*, pp. 109–111, 01 2014.
- [16] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, “Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2017.

-
- [17] Open Robotics, “Make a model.” `classic.gazebosim.org`. https://classic.gazebosim.org/tutorials?tut=build_model (accesado: Ago. 21, 2022).
- [18] Blender Foundation, “About.” `blender.org`. <https://www.blender.org/about/> (accesado: Ago. 21, 2022).
- [19] Trimble, “Help Center.” `help.sketchup.com`. <https://help.sketchup.com/en/sketchup/sketchup> (accesado: Ago. 21, 2022).
- [20] National Aeronautics and Space Administration, “What is Remote Sensing?.” `earthdata.nasa.gov`. <https://www.earthdata.nasa.gov/learn/backgrounders/remote-sensing> (accesado: Ago. 27, 2022).
- [21] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, “Accurate object localization in remote sensing images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2486–2498, 2017.
- [22] United States Geological Survey, “What is a geographic information system (GIS)?.” `usgs.gov`. <https://www.usgs.gov/faqs/what-geographic-information-system-gis> (accesado: Ago. 27, 2022).
- [23] D. Larnaout, S. Bourgeois, V. Gay-Bellile, and M. Dhome, “Towards bundle adjustment with gis constraints for online geo-localization of a vehicle in urban center,” in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 348–355, 2012.
- [24] European Union Agency for the Space Programme, “What is GNSS?.” `euspa.europa.eu`. <https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss> (accesado: Ago. 27, 2022).
- [25] B. Zhou and F. Wang, “Method and system for training unmanned aerial vehicle control model based on artificial intelligence,” Oct. 19 2021. US Patent 11,150,655.
- [26] M. Gong and W. Fan, “Method, apparatus and system of providing communication coverage to an unmanned aerial vehicle,” July 31 2018. US Patent 10,038,492.

-
- [27] K. Abuelsaad, G. J. Boss, K. C. McConnell, and S. B. McElligott, “Unmanned aerial vehicle for hazard detection,” Oct. 11 2016. US Patent 9,463,875.
- [28] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, “Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects,” *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.
- [29] G. Wang, Y. Han, J. Chen, S. Wang, Z. Zhang, N. Du, and Y. Zheng, “A gnss/ins integrated navigation algorithm based on kalman filter,” *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 232–237, 2018. 6th IFAC Conference on Bio-Robotics BIOROBOTICS 2018.
- [30] P. S. Madhukar and L. Prasad, “State estimation using extended kalman filter and unscented kalman filter,” in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pp. 1–4, 2020.
- [31] J. Pan, X. Yang, H. Cai, and B. Mu, “Image noise smoothing using a modified kalman filter,” *Neurocomputing*, vol. 173, pp. 1625–1629, 2016.
- [32] T. Qin, S. Cao, J. Pan, and S. Shen, “A general optimization-based framework for global pose estimation with multiple sensors,” 2019.
- [33] H. fa Dai, H. wei Bian, R. ying Wang, and H. Ma, “An ins/gnss integrated navigation in gnss denied environment using recurrent neural network,” *Defence Technology*, vol. 16, no. 2, pp. 334–340, 2020.
- [34] H.-U. Kim and T.-S. Bae, “Deep learning-based gnss network-based real-time kinematic improvement for autonomous ground vehicle navigation,” *Journal of Sensors*, vol. 2019, pp. 1–8, 03 2019.
- [35] Intel Corporation, “Edge Insights for Industrial.” intel.com. <https://www.intel.com/content/www/us/en/develop/documentation/edge-insights-industrial-doc/top/how-it-works.html> (accesado: Ago. 05, 2022).
- [36] Intel Corporation, “Audi Precision Manufacturing.” intel.com. <https://www.intel.com/content/www/us/en/customer-spotlight/stories/audi-automated-factory.html> (accesado: Ago. 05, 2022).

-
- [37] Intel Corporation, “Helping PCB Manufacturers Become More Efficient.” intel.com. <https://www.intel.com/content/www/us/en/edge-computing/resources/edge-software-hub-axiomtek-brief.html> (accesado: Ago. 05, 2022).
- [38] Intel Corporation, “Microvision Partners on Pharma Manufacturing.” intel.com. <https://www.intel.com/content/www/us/en/edge-computing/resources/edge-software-hub-microsivion-pharma-solution.html> (accesado: Ago. 05, 2022).
- [39] Intel Corporation, “How-to: Getting IMU data from D435i and T265.” intelrealsense.com. <https://www.intelrealsense.com/how-to-getting-imu-data-from-d435i-and-t265/> (accesado: Sep. 13, 2022).
- [40] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. Cham: Springer International Publishing, 2016.
- [41] Open Robotics, “SDFormat Specification.” sdformat.org. http://sdformat.org/spec?ver=1.7&elem=sensor#sensor_imu (accesado: Sep. 13, 2022).
- [42] L. Campos-Macías, R. Aldana-López, R. de la Guardia, J. I. Parra-Vilchis, and D. Gómez-Gutiérrez, “Autonomous navigation of mavs in unknown cluttered environments,” *Journal of Field Robotics*, vol. 38, no. 2, pp. 307–326, 2021.
- [43] Amazon, “Foxeer 2pcs FPV Antenna Lollipop 5.8G 2.6dBi Super Mini RHCP Antenna SMA Male for RC Drone FPV Quadcopeter Multicopter TX/RX (Black).” amazon.com. <https://www.amazon.com/Foxeer-Antenna-Lollipop-Quadcopeter-Multicopter/dp/B07D12XS6K> (accesado: Oct. 16, 2022).
- [44] Amazon, “DollaTek 28dB High Gain GPS Active Antenna Ceramic Patch Internal Module Navigation Plug with Cable.” amazon.de. <https://www.amazon.de/-/en/DollaTek-Antenna-Ceramic-Internal-Navigation/dp/B07MHGPT8L> (accesado: Oct. 16, 2022).

-
- [45] GNSS OEM, “NEO-M9N four GNSS USB receiver for Stratux ADS-B Aviation Receiver.” [gnss.store. https://gnss.store/neo-m9n-gnss-modules/131-elt0114.html](https://gnss.store/neo-m9n-gnss-modules/131-elt0114.html) (accesado: Oct. 16, 2022).
- [46] Flying Tech, “Matek M8Q-CAN - GPS, Compass Barometer Module (UAVCAN).” [flyingtech.co.uk. https://www.flyingtech.co.uk/electronics/matek-m8q-can-gps-compass-barometer-module-uavcan](https://www.flyingtech.co.uk/electronics/matek-m8q-can-gps-compass-barometer-module-uavcan) (accesado: Oct. 16, 2022).
- [47] Titan Elite, “Zubax GNSS 2 Receiver, GPS + GLONASS + Galileo with Ublox MAX M8 (M8Q) Engine.” [titaneliteinc.com. https://shop.titaneliteinc.com/index.php?route=product/product&product_id=987](https://shop.titaneliteinc.com/index.php?route=product/product&product_id=987) (accesado: Oct. 16, 2022).
- [48] Amazon, “Stemedu USB GPS Module GLONASS Beidou UBX-M8030-KT GPS Dongle.” [amazon.com. https://www.amazon.com/-/es/Stemedu-UBX-M8030-KT-ceramica-integrada-Arduino/dp/B07ZPM6HB5](https://www.amazon.com/-/es/Stemedu-UBX-M8030-KT-ceramica-integrada-Arduino/dp/B07ZPM6HB5) (accesado: Oct. 16, 2022).
- [49] RC Innovations, “Matek GPS GNSS Module M9N F4 3100 Compass CAN - UAVCAN.” [rc-innovations.es. https://rc-innovations.es/en/shop/matek-m9n-f4-3100-compass-uavcan-msp-gnss-module-gps-glonass-galileo-beidou#attr=5225](https://rc-innovations.es/en/shop/matek-m9n-f4-3100-compass-uavcan-msp-gnss-module-gps-glonass-galileo-beidou#attr=5225) (accesado: Oct. 16, 2022).
- [50] British Plastic Federation, “Acrylonitrile Butadiene Styrene (ABS) and Other Specialist Styrenics.” [bpf.co.uk. https://www.bpf.co.uk/plastipedia/polymers/ABS_and_Other_Specialist_Styrenics.aspx](https://www.bpf.co.uk/plastipedia/polymers/ABS_and_Other_Specialist_Styrenics.aspx) (accesado: Oct. 17, 2022).
- [51] D. Tan, “Depth Estimation: Basics and Intuition.” [towardsdatascience.com. https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1](https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1) (accesado: Oct. 18, 2022).
- [52] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [53] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4304–4311, 2016.

-
- [54] O. Team, “Showcase.” ogre3d.org. <https://www.ogre3d.org/showcase> (accesado: Oct. 24, 2022).
- [55] J. S. Discovery, “Optimality Criteria.” jmp.com. <https://www.jmp.com/support/help/en/16.2/index.shtml#page/jmp/optimality-criteria.shtml> (accesado: Oct. 22, 2022).
- [56] M. Inc, “¿Qué son los diseños de superficie de respuesta, los diseños centrales compuestos y los diseños de Box-Behnken?.” support.minitab.com. <https://support.minitab.com/es-mx/minitab/20/help-and-how-to/statistical-modeling/doe/supporting-topics/response-surface-designs/response-surface-central-composite-and-box-behnken-designs/> (accesado: Oct. 23, 2022).
- [57] McMaster-Carr, “18-8 Stainless Steel Socket Head Screw M3 x 0.5 mm Thread, 25 mm Long.” mcmaster.com. <https://www.mcmaster.com/91292A020/> (accesado: Oct. 17, 2022).
- [58] McMaster-Carr, “Steel Hex Nut Medium-Strength, Class 8, M3 x 0.5 mm Thread.” mcmaster.com. <https://www.mcmaster.com/90592A085/> (accesado: Oct. 17, 2022).
- [59] Amazon, “ABS 3D Printer Filament, 1.75mm, Red, 1 kg Spool.” amazon.com. <https://www.amazon.com/AmazonBasics-Printer-Filament-1-75mm-Spool/dp/B07Y3PHD2C> (accesado: Oct. 26, 2022).
- [60] Amazon, “Short Micro USB Cable.” amazon.com. <https://www.amazon.com/CableCreation-Charger-Compatible-Chromecast-Android/dp/B013G4EAEI> (accesado: Oct. 17, 2022).
- [61] SZ DJI Technology, “D-RTK 2 High Precision GNSS Mobile Stationl.” dji.com. <https://www.dji.com/d-rtk-2> (accesado: Oct. 26, 2022).
- [62] Juniper Systems, “Juniper Systems Geode GNS3 Real-Time GNSS Multi-Frequency Receiverl.” allterracentral.com. <https://allterracentral.com/juniper-systems-geode-gns3-real-time-gnss-receiver-multi-frequency-antenna-model-1.html> (accesado: Oct. 26, 2022).

- [63] ComNav Technology, “N3 IMU GNSS Receiver.” [comnavtech.com. https://www.comnavtech.com/new_N3.html](https://www.comnavtech.com/new_N3.html) (accesado: Oct. 26, 2022).
- [64] NovAtel, “Dual-antenna CPT7 enclosure with GNSS+INS technology.” [novatel.com. https://novatel.com/products/gnss-inertial-navigation-systems/combined-systems/cpt7](https://novatel.com/products/gnss-inertial-navigation-systems/combined-systems/cpt7) (accesado: Oct. 26, 2022).
- [65] Intel Corporation, “Tracking Camera T265 / T261 Datasheet.” [dev.intelrealsense.com. https://dev.intelrealsense.com/docs/tracking-camera-t265-datasheet](https://dev.intelrealsense.com/docs/tracking-camera-t265-datasheet) (accesado: Oct. 14, 2022).
- [66] Bosch, “IMU BMI055.” [bosch-sensortec.com. https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi055/](https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi055/) (accesado: Oct. 14, 2022).
- [67] Ublox, “NEO-M9N module.” [u-blox.com. https://www.u-blox.com/en/product/neo-m9n-module](https://www.u-blox.com/en/product/neo-m9n-module) (accesado: Oct. 19, 2022).
- [68] Infineon, “DPS310.” [infineon.com. https://www.infineon.com/cms/en/product/sensor/pressure-sensors/pressure-sensors-for-iot/dps310/](https://www.infineon.com/cms/en/product/sensor/pressure-sensors/pressure-sensors-for-iot/dps310/) (accesado: Oct. 19, 2022).

APÉNDICE A

MATERIAL COMPLEMENTARIO

A.1. Incorporación del Módulo GNSS

Este apartado contiene las instrucciones para habilitar la captación de datos geoespaciales en los prototipos de aeronaves autónomas desarrollados por el departamento de Human Robot Collaboration. Las especificaciones finales del sistema son:

- Precisión horizontal de 2.0 metros CEP.
- Precisión vertical de ± 0.5 metros.
- Tasa de actualización de 25 hercios.

En el Cuadro A.1 se encuentra la lista de partes requerida para ensamblar el sistema. Refiérase a las Figuras A.1-A.7 para ejecutar el ensamblaje de forma secuencial.

APÉNDICE A. MATERIAL COMPLEMENTARIO

Cuadro A.1: Lista de partes para ensamble del módulo GNSS

Elemento	Especificación	Cantidad	Proveedor
Módulo GNSS	M9N-F4	1	[49]
Perno Cabeza Hueca	M3 x 0.5 x 25mm	4	[57]
Tuerca	M3 x 0.5 mm	4	[58]
Cobertor	N/A	1	Anexo B.1
Cable Micro-USB	0.5 ft	1	[60]

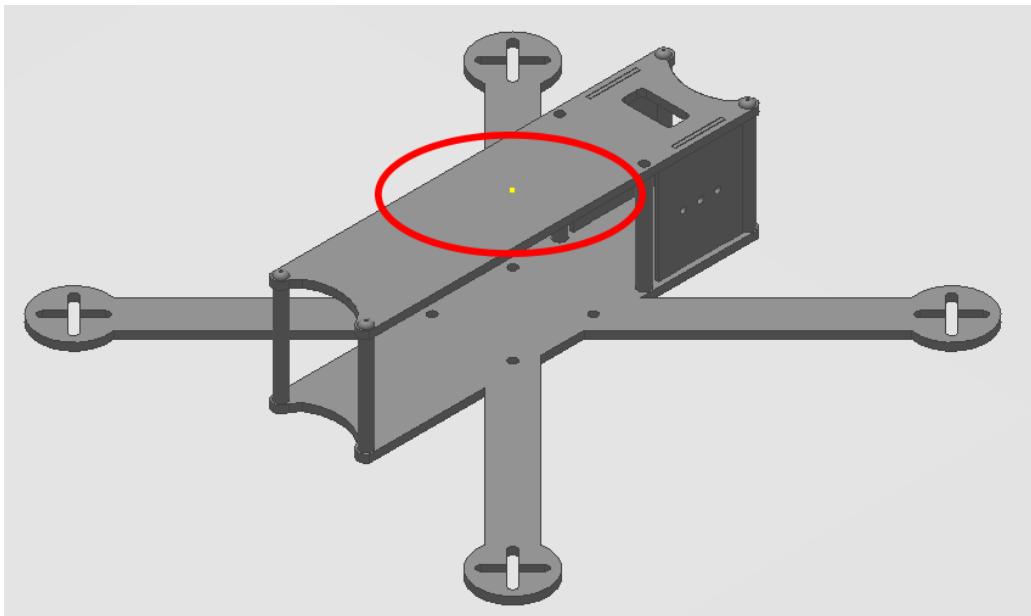


Figura A.1: Identifique la zona designada para colocar el módulo GNSS.

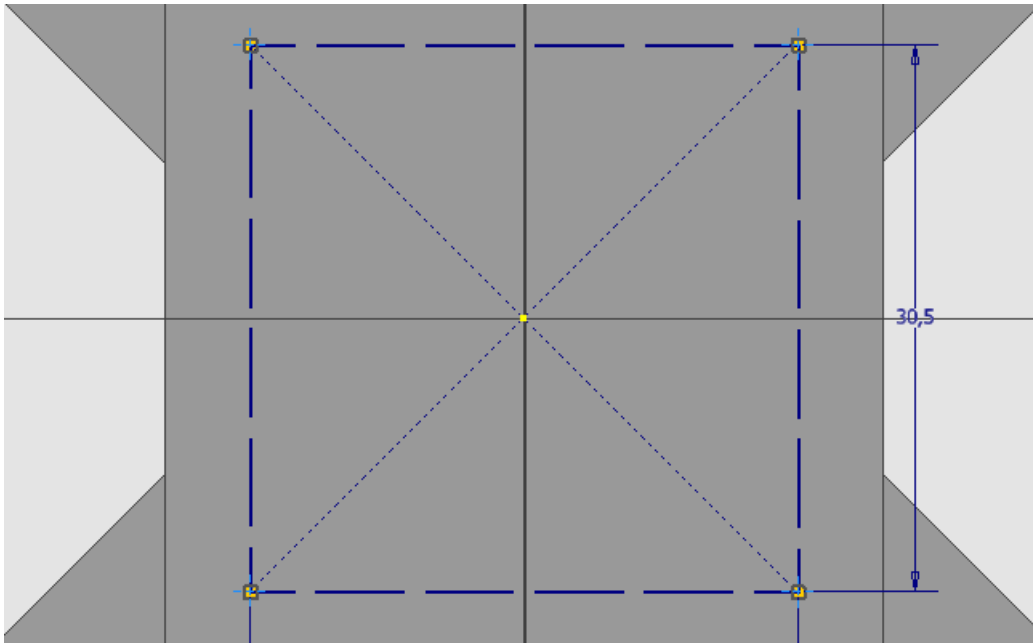


Figura A.2: Marque cuatro puntos sobre la superficie, formando un cuadrado de 30.5 mm de lado.

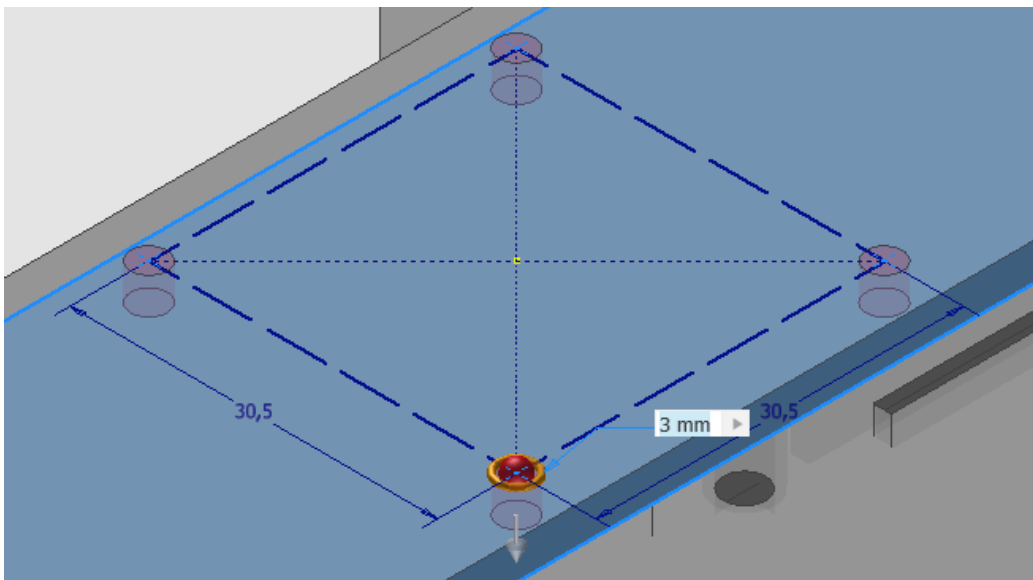


Figura A.3: Realice cuatro agujeros sobre los puntos marcados, con un diámetro de 3 mm.

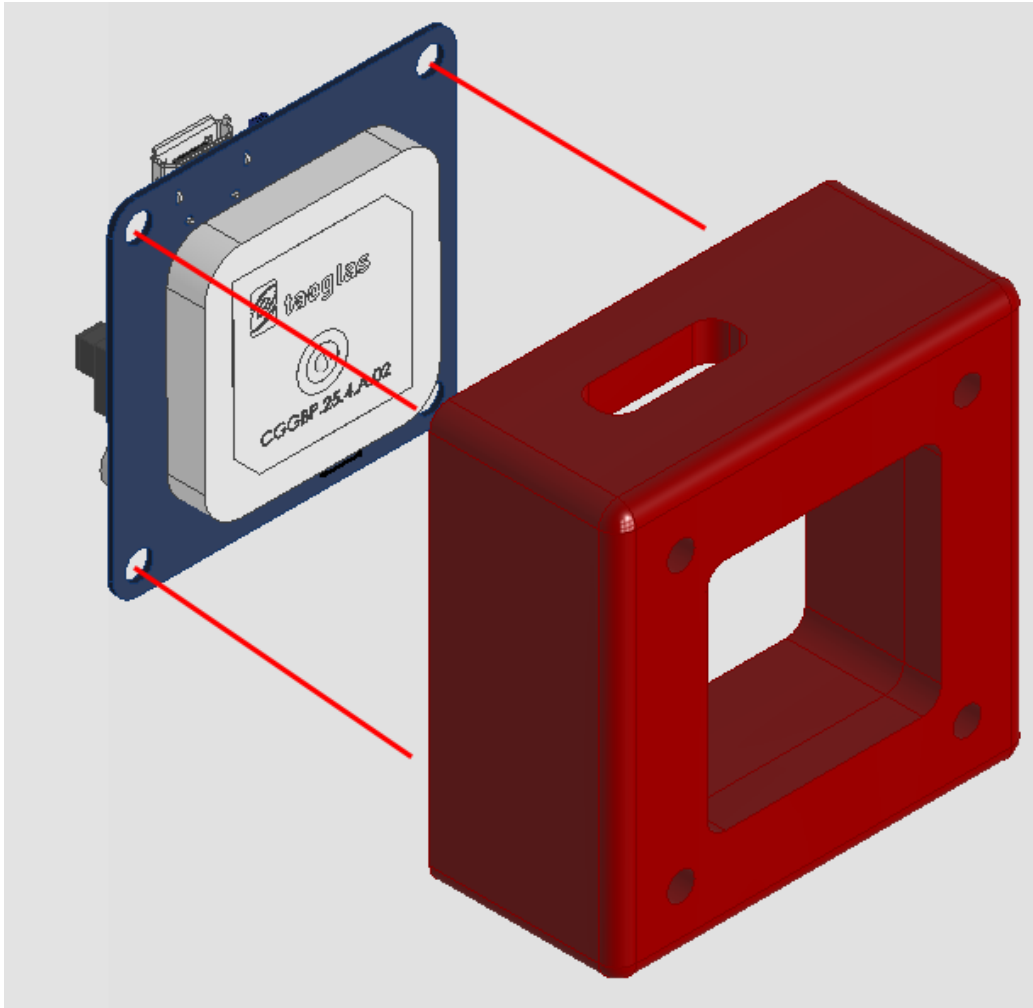


Figura A.4: Introduzca el módulo M9N-F4 en el cobertor.



Figura A.5: Asegúrese que el conector micro-USB se encuentra alineado con la abertura del cobertor.

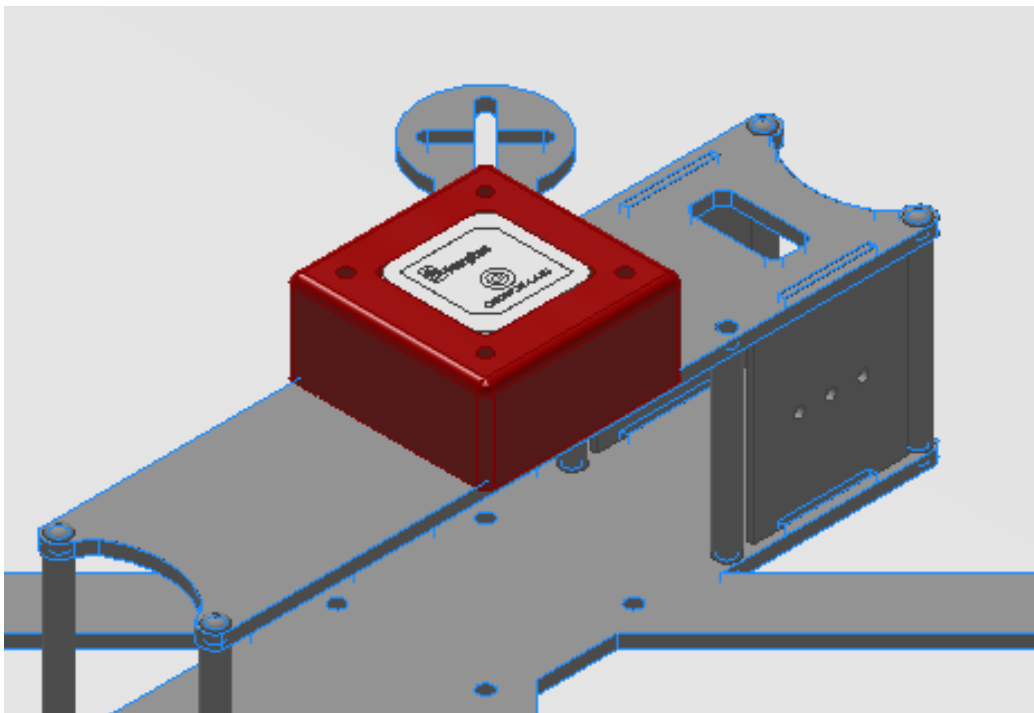


Figura A.6: Coloque el ensamble sobre la superficie de la aeronave, de forma que todos los agujeros queden alineados.

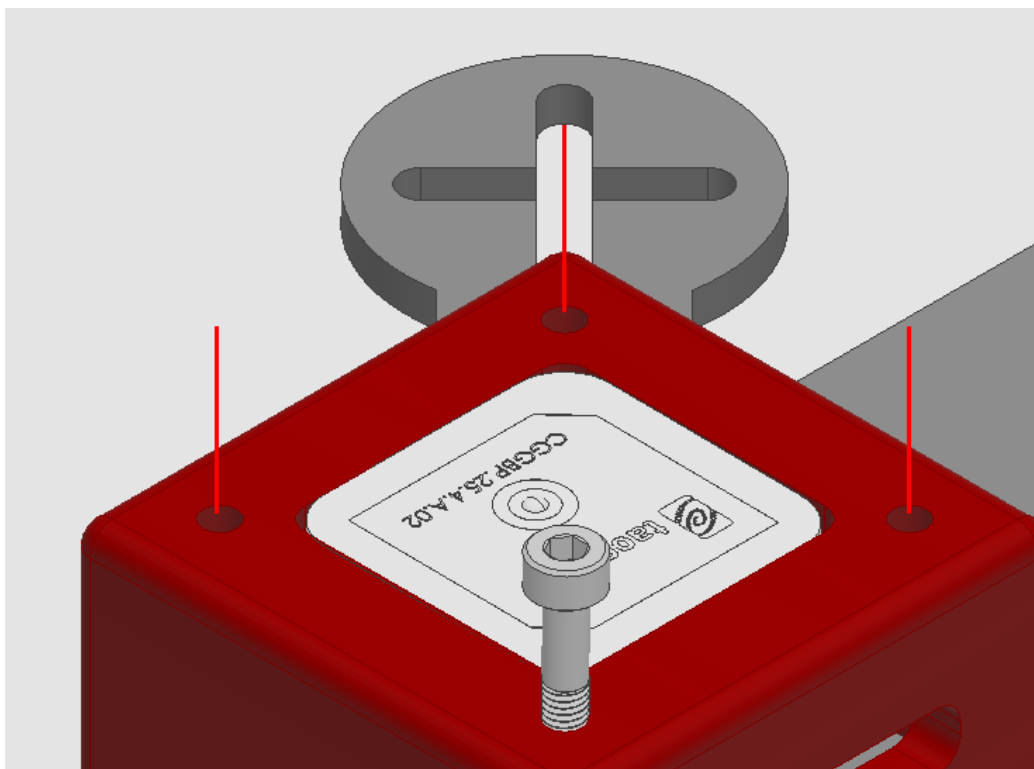


Figura A.7: Finalmente, coloque los cuatro pernos de cabeza hueca en los agujeros, generando un ajuste firme con las tuercas. Conecte el módulo al procesador empleando el cable micro-USB.

A.2. Definición de Factores de Optimización

A.2.1. IMU-Cámaras

Partiendo de la definición general de estado \mathcal{X} , realizada en la Ecuación 4.8, se toma ventaja de que la estimación con sensores locales es bastante precisa en regiones pequeñas. Considerando dos instantes de tiempo seguidos, $t - 1$ y t , el factor local para una medición \mathbf{z} y una estimación h se deriva cómo:

$$\begin{aligned} \mathbf{z}_t^l - h_t^l(\mathcal{X}) &= \mathbf{z}_t^l - h_t^l(\mathbf{x}_{t-1}, \mathbf{x}_t) \\ &= \begin{bmatrix} \mathbf{q}_{t-1}^{l-1} (\mathbf{p}_t^l - \mathbf{p}_{t-1}^l) \\ \mathbf{q}_{t-1}^{l-1} \mathbf{q}_t^l \end{bmatrix} \ominus \begin{bmatrix} \mathbf{q}_{t-1}^{w-1} (\mathbf{p}_t^w - \mathbf{p}_{t-1}^w) \\ \mathbf{q}_{t-1}^{w-1} \mathbf{q}_t^w \end{bmatrix} \end{aligned} \quad (\text{A.1})$$

Los términos \mathbf{q} , \mathbf{p} corresponden a la orientación y la posición en el sistema de referencia local l . Por otra parte, \ominus es la operación de resta en el estado del error en cuaterniones.

A.2.2. GNSS

Las mediciones sin procesar de un módulo GNSS son longitud latitud y altitud, que son distintas a las coordenadas x , y y z que se utilizan para representar la localización. No obstante, esta información se puede convertir bajo el sistema local este, norte y arriba. La primera medición del módulo se toma como el punto de origen, por lo que el factor se deriva:

$$\mathbf{z}_t^{GNSS} - h_t^{GNSS}(\mathcal{X}) = \mathbf{z}_t^{GNSS} - h_t^{GNSS}(\mathbf{x}_t) = \mathbf{p}_t^{GNSS} - \mathbf{p}_t^w \quad (\text{A.2})$$

Las mediciones del módulo GNSS restringen directamente la posición de cada nodo. La covarianza se determina en base a la cantidad de satélites disponibles cuando se realizó la medición.

A.2.3. Barómetro

El barómetro mide la presión de aire, por lo que se asume que esta es constante para una determinada altitud en un periodo de tiempo. Bajo este supuesto, la presión puede ser convertida a coordenadas de altitud de forma

lineal. Igual que en el caso anterior, se toma la primera medición cómo el origen de altitud. El factor bajo el sistema de referencia m sigue:

$$\mathbf{z}_t^m - h_t^m(\mathcal{X}) = \mathbf{z}_t^m - h_t^m(\mathbf{x}_t) = z_t^m - z_t \quad (\text{A.3})$$

Debido a que las mediciones de este sensor son ruidosas, la varianza se calcula tomando varias mediciones en un corto periodo de tiempo.

A.3. Repositorio de Documentación

En la Figura A.8 se muestra la organización del repositorio con material complementario, generado durante el desarrollo del proyecto.

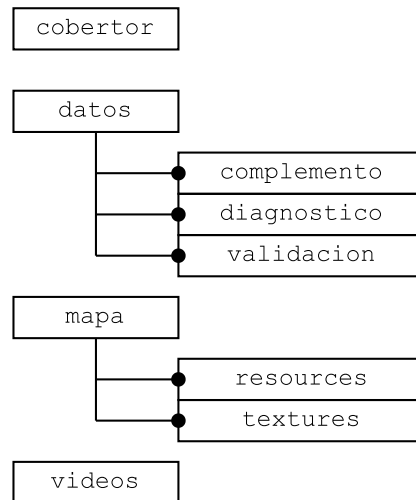


Figura A.8: Repositorio de material complementario

El contenido puede ser accedido en <https://github.com/MiguelSolisSegura/material-complementario> e incluye:

- *cobertor*: archivos del sólido del cobertor, en formato editable y para impresión.
- *datos*: datos de los vuelos realizados en el simulador, incluyendo los experimentos complementarios, de diagnóstico y de validación.
- *mapa*: archivos del escenario de simulación generado, donde se incluyen los sólidos y las texturas empleadas.
- *videos*: grabaciones de los principales vuelos de diagnóstico y validación.

A.4. Repositorio de Implementación

La implementación programada se puede consultar en <https://github.com/MiguelSolisSegura/implementacion-programada>. Esta consta de dos directorios principales, según se muestra en la Figura A.9.

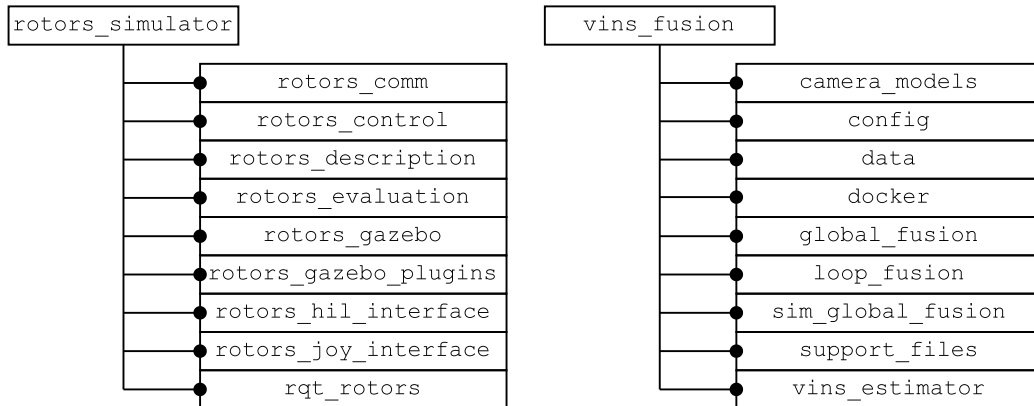


Figura A.9: Repositorio de implementación programada

El directorio *rotors.simulator* contiene los archivos necesarios para la simulación del sistema en Gazebo. El contenido se subdivide en:

- *rotors_comm*: definición de mensajes y servicios específicos.
- *rotors_control*: controladores para realizar maniobras con las aeronaves en el espacio simulado.
- *rotors_description*: archivos de configuración, en los que se incluye los sensores utilizados en la aeronave y las condiciones de simulación.
- *rotors_evaluation*: archivos de python para la evaluación de métricas experimentales.
- *rotors_gazebo*: complementos para la simulación en Gazebo, se incluye el escenario de simulación realizado en el proyecto.
- *rotors_gazebo_plugins*: definición del comportamiento y físicas de los sensores utilizados.
- *rotors_hil_interface*: elementos de apoyo para la simulación de hardware real.

- *rotors_joy_interface*: nodo para el control de la aeronave mediante un joystick externo.
- *rqt_rotors*: complementos de interfaz gráfica de usuario.

Por otra parte, el directorio *vins_simulator* incluye las modificaciones pertinentes a partir de la implementación propuesta por Qin et al. [32] para la fusión de señales. Los directorios contienen:

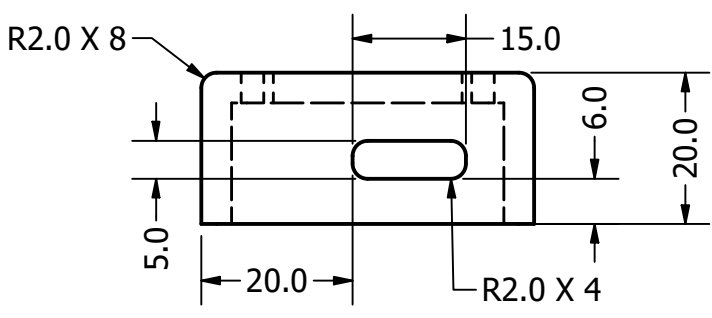
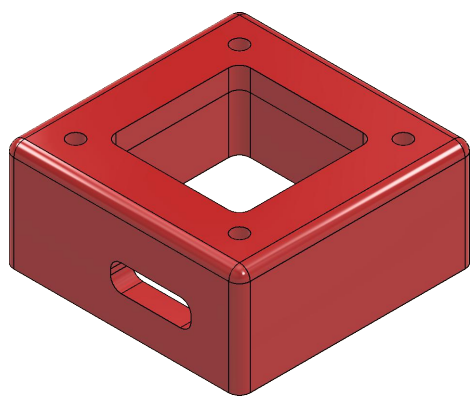
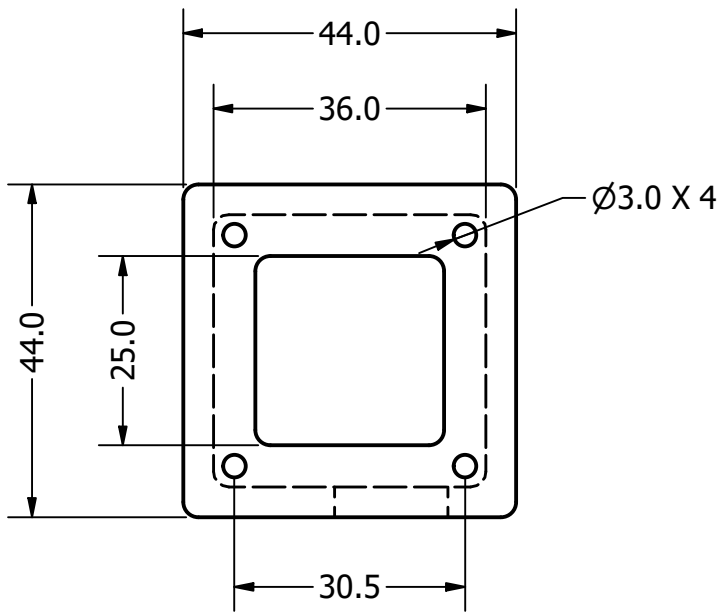
- *camera_models*: herramientas para la calibración de una cámara real.
- *config*: configuración del hardware utilizado, en la que se incluye los parámetros intrínsecos y extrínsecos de la unidad de medición inercial y las cámaras de simulación.
- *data*: directorio dónde se almacenan los datos de los experimentos, una vez que se culmina la ejecución del sistema.
- *docker*: utilidades para el despliegue en un contenedor de software.
- *global_fusion*: nodo que realiza la fusión de sensores globales y la odometría visual inercial.
- *loop_fusion*: nodo para el cierre de bucles en la trayectoria seguida.
- *simulation_global_fusion*: nodo que realiza la fusión de sensores globales y la odometría visual inercial, con funcionalidades ampliadas para su compatibilidad con Gazebo.
- *support_files*: archivos de documentación adicional.
- *vins_estimator*: nodo que ejecuta el proceso de odometría visual inercial.

ANEXO B

DOCUMENTACIÓN ADICIONAL

B.1. Plano del Cobertor para el Módulo GNSS

Consultar siguiente página.



Dibuja Miguel Solís	17/10/2022	Intel Corporation		
Revisa Rafael de la Guardia				
Control		Título		
Manufactura		Cobertor para el Módulo GNSS		
Aprueba		Tamaño	Número	Rev.
Indicación Makerbot Replicator 2X		Carta	N/A	1
Resolución 0.1 milímetros		Escala	Unidades	Hoja
Archivo cobertor.stl		1 : 1	milímetros	1 de 1

B.2. Hoja de Datos de la Cámara de Rastreo RealSense T265

Dimension	Min	Nominal	Max	Unit
Width	107.85	108.00	108.15	mm
Height	24.35	24.50	24.65	mm
Depth	12.35	12.50 </td <td>12.65</td> <td>mm</td>	12.65	mm
Flatness Tolerance	-	0.15	-	mm
Weight	57	60	63	gr

Figura B.1: Características mecánicas [65]

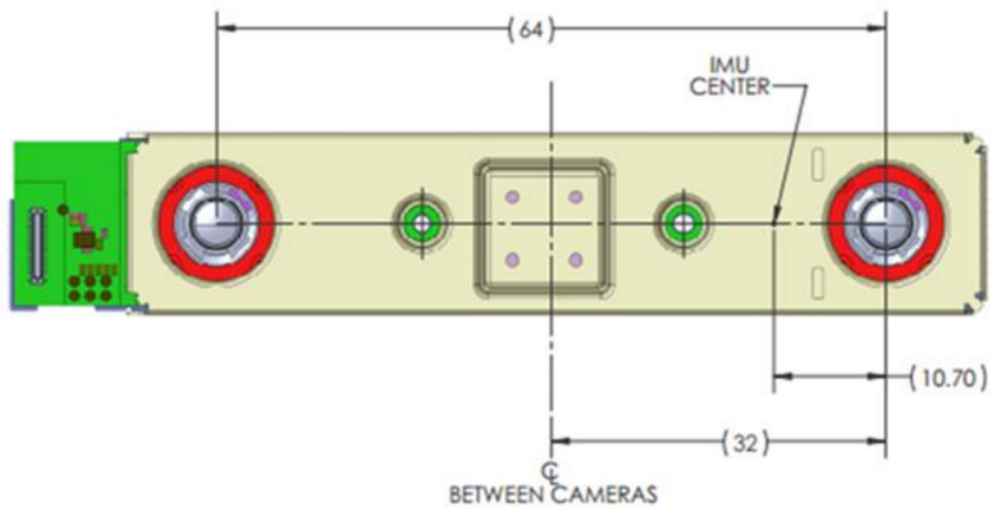


Figura B.2: Disposición de las cámaras [65]

ANEXO B. DOCUMENTACIÓN ADICIONAL

Parameter	Camera Sensor Properties
Active Pixels	848 X 800
Sensor Aspect Ratio	1.06
Format	8bit, 10-bit RAW
Filter Type	IR Cut Filter
Focus	Fixed
Shutter Type	Global Shutter
Signal Interface	MIPI CSI-2, 2 X Lanes

Figura B.3: Propiedades del sensor de la cámara [65]

Imaging Module	Intel® RealSense™ Tracking Module T261
Baseline (mm)	64±0.15
Left/Right Fisheye Imagers	OV9282
Shutter Type	Global
Fisheye FOV (degrees)	D:173
Module Dimensions (mm)	X=93.35 (+0.15 -0.25) Y=17.60±0.15 Z=7.13±0.30

Figura B.4: Propiedades ópticas [65]

B.3. Hoja de Datos del IMU BMI055

OPERATING CONDITIONS GYROSCOPE						
Parameter	Symbol	Condition	Min	Typ	Max	Unit
Range	R_{FS125}	Selectable via serial digital interface		125		°/s
	R_{FS250}			250		°/s
	R_{FS500}			500		°/s
	R_{FS1000}			1,000		°/s
	R_{FS2000}			2,000		°/s
Supply Current in Normal Mode	I_{DD}	see ²		5		mA
Supply Current in Fast Power-up Mode	I_{DDfpm}	see ²		2.5		mA
Supply Current in Suspend Mode	I_{DDsum}	see ² , digital and analog (only IF active)		25		μA
Supply Current in Deep Suspend Mode	I_{DDsum}	see ²		<5		μA
Start-up time	t_{su}	to ±1% of final value; from power-off		30		ms
Wake-up time	t_{wusm}	From suspend- and deep suspend-modes		30		ms
Wake-up time	t_{wufpm}	From fast power-up mode		10		ms
Non-volatile memory (NVM) write-cycles	n_{NVM}				15	cycles

Figura B.5: Características eléctricas del giroscopio [66]

ANEXO B. DOCUMENTACIÓN ADICIONAL

OUTPUT SIGNAL ACCELEROMETER						
Parameter	Symbol	Condition	Min	Typ	Max	Units
Sensitivity	S _{2g}	g _{FS2g} , T _A =25°C		1024		LSB/g
	S _{4g}	g _{FS4g} , T _A =25°C		512		LSB/g
	S _{8g}	g _{FS8g} , T _A =25°C		256		LSB/g
	S _{16g}	g _{FS16g} , T _A =25°C		128		LSB/g
Sensitivity Temperature Drift	TCS	g _{FS2g} , Nominal V _{DD} supplies		±0.02		%/K
Sensitivity Supply Volt. Drift	S _{VDD}	g _{FS2g} , T _A =25°C, V _{DD min} ≤ V _{DD} ≤ V _{DD max}		0.05		%/V
Zero-g Offset (x,z)	Off _{x,z}	g _{FS2g} , T _A =25°C, nominal V _{DD} supplies, over life- time		±70		mg
Zero-g Offset (y)	Off _y	g _{FS2g} , T _A =25°C, nominal V _{DD} supplies, over life- time		±70		mg
Zero-g Offset Temperature Drift	TCO	g _{FS2g} , Nominal V _{DD} supplies		±1		mg/K
Zero-g Offset Supply Volt. Drift	Off _{VDD}	g _{FS2g} , T _A =25°C, V _{DD min} ≤ V _{DD} ≤ V _{DD max}		0.5		mg/V
Bandwidth	bw ₈	2 nd order filter, bandwidth programmable		8		Hz
	bw ₁₆			16		Hz
	bw ₃₁			31		Hz
	bw ₆₃			63		Hz
	bw ₁₂₅			125		Hz
	bw ₂₅₀			250		Hz
	bw ₅₀₀			500		Hz
	bw ₁₀₀₀			1,000		Hz
Nonlinearity	NL	best fit straight line, g _{FS2g}		±0.5		%FS
Output Noise Density	n _{rms}	g _{FS2g} , T _A =25°C Nominal V _{DD} supplies Normal mode		150		µg/√Hz
Temperature Sensor Measurement Range	T _S		-40		85	°C
Temperature Sensor Slope	dT _S			0.5		K/LSB
Temperature Sensor Offset	OT _S			±2		K

Figura B.6: Características eléctricas del acelerómetro [66]

B.4. Hoja de Datos del GNSS NEO-M9N-00B

Parameter	Specification					
Receiver type	Multi-constellation GNSS standard precision receiver					
Accuracy of time pulse signal	RMS 30 ns 99% 60 ns					
Frequency of time pulse signal	0.25 Hz to 10 MHz (configurable)					
Operational limits ¹	Dynamics ≤ 4 g Altitude 80,000 m Velocity 500 m/s					
Velocity accuracy ²	0.05 m/s					
Dynamic heading accuracy ²	0.3 deg					
GNSS	GPS+GLO+GAL+BDS	GPS+GLO+GAL	GPS+GLO	GPS+BDS	GPS+GAL	
Acquisition ³	Cold start	24 s	25 s	26 s	28 s	29 s
	Hot start	2 s	2 s	2 s	2 s	2 s
	Aided start ⁴	2 s	2 s	2 s	2 s	2 s
Nav. update rate	PVT	25 Hz	25 Hz	25 Hz	25 Hz	25 Hz

Figura B.7: Desempeño general del GNSS [67]

B.5. Hoja de Datos del Barómetro DPS310

Features

- **Operation range:** Pressure: 300 –1200 hPa. Temperature: -40 – 85 °C.
- **Pressure sensor precision:** ± 0.002 hPa (or ± 0.02 m) (high precision mode).
- **Relative accuracy:** ± 0.06 hPa (or ± 0.5 m)
- **Absolute accuracy:** ± 1 hPa (or ± 8 m)
- **Temperature accuracy:** ± 0.5 °C.
- **Pressure temperature sensitivity:** 0.5Pa/K
- **Measurement time: Typical:** 27.6 ms for standard mode (16x). Minimum: 3.6 ms for low precision mode.
- **Average current consumption:** 1.7 μ A for Pressure Measurement, 1.5uA for Temperature measurement @1Hz sampling rate, Standby: 0.5 μ A.
- **Supply voltage:** VDDIO: 1.2 – 3.6 V, VDD: 1.7 – 3.6 V.
- **Operating modes:** Command (manual), Background (automatic), and Standby.
- **Calibration:** Individually calibrated with coefficients for measurement correction.
- **FIFO:** Stores up to 32 pressure or temperature measurements.
- **Interface:** I²C and SPI (both with optional interrupt)
- **Package dimensions:** 8-pin LGA, 2.0 mm x 2.5 mm x 1.0 mm.
- **Green Product (RoHS) Compliant**

Figura B.8: Características del barómetro [68]