

SISTEMA DE *SENSADO REMOTO* PARA LA AGRICULTURA DE PRECISIÓN



LISETH VIVIANA CAMPO ARCOS

Tesis de Maestría en Ingeniería Telemática

Director:

Juan Carlos Corrales Muñoz
Doctor en Ciencias de Computación

**Universidad del Cauca
Facultad de ingeniería en Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de investigación e-@mbiente
Popayán, 2017**

LISETH VIVIANA CAMPO ARCOS

**SISTEMA DE *SENSADO REMOTO* PARA LA
AGRICULTURA DE PRECISIÓN**

**Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de**

Magister en:
Ingeniería Telemática

Director:
PhD. Juan Carlos Corrales Muñoz

**Popayán
2017**

***A Dios, quien en su infinita misericordia me dio la vida
A mi hijo Samuel, quien es el regalo más hermoso que haya podido recibir
A mi familia, por su apoyo incondicional***

Resumen Estructurado

Antecedentes: La agricultura de precisión ha transformado durante más de 30 años los métodos de producción con el fin de gestionar de manera eficiente los recursos en el campo agrícola. El método comúnmente utilizado para las prácticas de agricultura de precisión, es el *sensado remoto*, que actualmente se apoya en vehículos aéreos no tripulados. Sin embargo, los vehículos aéreos no tripulados o *drones* son comercializados con un diseño estándar, que implica que estas aeronaves no se ajusten a entornos cambiantes como los cultivos agrícolas. De esta manera se torna necesario el diseño de alternativas de *sensado remoto* basado en vehículos aéreos no tripulados que superen las actuales falencias y contribuyan a los esquemas de toma de decisiones sobre los recursos de los cultivos.

Objetivo: Implementar un sistema de *sensado remoto* para agricultura de precisión soportado en un robot aéreo multirroto de bajo costo (*dron*).

Métodos: Se propone un sistema de *sensado remoto* para la recolección de información de cultivos agrícolas basado en una plataforma aérea de bajo costo, denominada AR Drone. La propuesta está dividida en dos módulos de desarrollo fundamentales: un módulo de posicionamiento basado en filtro de Kalman que se adapte a condiciones ambientales no controladas, como los cultivos agrícolas; y un módulo de planificación de ruta que permita realizar cobertura completa de un área específica mediante algoritmos basados en descomposición de área aproximada.

Resultados: El presente proyecto declara como resultados un conjunto de módulos de desarrollo para navegación y control del cuadricóptero AR Drone soportados en el sistema operativo ROS, un conjunto de algoritmos de planificación de cobertura para ambientes abiertos y una estrategia de *sensado remoto* utilizando una plataforma de bajo costo denominada AR Drone para la obtención de información confiable para la agricultura de precisión.

Conclusiones: La validación de los módulos desarrollados en este trabajo de investigación arroja que el método que gestiona de manera óptima el posicionamiento de un cuadricóptero de bajo costo utilizando GPS para ambientes abiertos, es el filtro de Kalman. Además, las pruebas denotan que los algoritmos de planificación de ruta de cobertura tienen una eficiencia cercana al 90%, sin embargo, cumplen su función de acuerdo a factores externos. Los módulos planteados anteriormente marcan una diferencia entre los sistemas de teledetección conocidos comercialmente, ya que no solo gestionan la captura de datos, sino también mejoran el rendimiento de la plataforma considerando el viento, obstáculos y el sensor portado.

Palabras Clave: Ar Drone, posicionamiento, planificación de ruta de cobertura, ROS, *sensado remoto*.

Structured Abstract

Background: Precision agriculture has transformed production methods for more than 30 years to efficiently manage resources in the agriculture. The commonly used method for precision farming practices is the remote sensing, which currently relies on unmanned aerial vehicles to firm up the agricultural sector. However, unmanned aerial vehicles or *drones* are marketed with a standard design, which implies that these aircraft do not adjust to changing environments such as agricultural crops. In this way, it is necessary to design remote sensing alternatives based on unmanned aerial vehicles that overcome the current shortcomings and contribute to decision-making schemes on crop resources.

Objective: Implement a remote sensing system for precision agriculture supported on an aerial robot inexpensive (dron).

Methods: A remote sensing system is proposed for the collection of information from agricultural crop based on a low cost aerial platform, called AR Drone. The proposal is divided into two fundamental development modules: a Kalman filter-based positioning module that adapts to uncontrolled environmental conditions, such as agricultural crops; And a path planning module that allows complete coverage of a specific area using algorithms based on approximate area decomposition and considering the aerial platform resources.

Results: The present project states as results a set of development modules for navigation and control of the AR Drone quadcopter supported in the operating system ROS; a set of coverage planning algorithms for open environments, and a remote sensing strategy using a low-cost platform, called AR Drone, to obtain reliable information for precision agriculture.

Conclusions: The validation of the modules developed in this research reveals that the method that optimally manages the positioning of a low-cost quadcopter using GPS for open environments is the Kalman filter. In addition, the tests denote that the algorithms of coverage path planning have an efficiency near 90%, nevertheless they fulfill their function according to external factors. The above-mentioned modules make a difference between commercially known remote sensing systems, as the prototype not only manages data capture, but also improves the performance of the platform considering the wind, obstacles and sensor.

Keywords: Ar Drone, positioning, coverage path planning, ROS, remote sensing.

Contenido

LISTA DE FIGURAS	VII
LISTA DE TABLAS	IX
INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DE PROBLEMA	1
1.2 ESCENARIO DE MOTIVACIÓN	2
1.3 OBJETIVOS	3
1.3.1 <i>Objetivo General</i>	3
1.3.2 <i>Objetivos Específicos</i>	3
1.4 CONTRIBUCIONES	3
1.5 CONTENIDO DE MONOGRAFÍA.....	4
ESTADO ACTUAL DE CONOCIMIENTO	5
2.1 CONCEPTOS GENERALES	5
2.1.1 <i>Agricultura de precisión</i>	5
2.1.2 <i>Sensado remoto</i>	6
2.1.3 <i>Vehículos aéreos no tripulados</i>	7
2.2 TRABAJOS RELACIONADOS	9
2.2.1 <i>Cuadricópteros para Investigación</i>	11
2.2.2 <i>Posicionamiento de drones en ambientes externos</i>	12
2.2.3 <i>Planificadores de Ruta de Cobertura</i>	13
2.2.4 <i>Sistemas de Sensado remoto con Drones</i>	15
2.2.5 <i>Brechas de Investigación</i>	16
2.3 PLATAFORMA AÉREA PARA SENSADO REMOTO	17
2.3.1 <i>Plataforma Aérea: AR Drone</i>	18
2.3.2 <i>Procesamiento en vuelo: Raspberry Pi 2 modelo B</i>	19
2.3.3 <i>ROS</i>	20
2.4 RESUMEN.....	21
POSICIONAMIENTO DEL DRONE EN AMBIENTES ABIERTOS.....	23
3.1 NAVEGACIÓN DE UN ROBOT AÉREO	23
3.1.1 <i>Sistemas de Referencia para un Robot Aéreo</i>	24
3.1.2 <i>Modelo Físico-Matemático de un Cuadricóptero</i>	26
3.1.3 <i>Odometría en ROS</i>	30
3.2 SELECCIÓN DE LA ESTRATEGIA DE POSICIONAMIENTO	31
3.2.1 <i>Navegación del Ar Drone</i>	31
3.2.2 <i>Navegación por Visión</i>	32
3.2.3 <i>Navegación por GPS</i>	32
3.2.4 <i>Comparación de sistemas de navegación</i>	33
3.2.5 <i>Fusión Sensorial: Filtro de Kalman (FK)</i>	34
3.3 DISEÑO DE LA ESTRATEGIA DE POSICIONAMIENTO	38
3.4 RESUMEN.....	39
PLANIFICACIÓN DE VUELO PARA SENSADO REMOTO DE CULTIVOS	41
4.1 ALGORITMOS DE COBERTURA GLOBAL PARA ROBOTS MÓVILES.....	41

4.1.1 Modelo del Mundo para Algoritmos de Cobertura Global.....	42
4.2 ALGORITMO DE COBERTURA BASADO EN DIJKSTRA	44
4.3 ALGORITMO DE COBERTURA CON FRENTE DE ONDA.....	47
4.4 ALGORITMO DE COBERTURA EN ESPIRALES	50
4.5 COMPARACIÓN DE ALGORITMOS DE COBERTURA	53
4.6 PLAN DE VUELO PARA SENSADO REMOTO	55
4.6.1 Proceso de captura automática de imágenes	55
4.6.2 Punto de Partida de Ruta de Cobertura.....	58
4.7 RESUMEN.....	59
EXPERIMENTACIÓN Y EVALUACIÓN.....	61
5.1 PROTOTIPO	61
5.1.1 Arquitectura de Control para AR Drone	62
5.1.2 Fusión Sensorial.....	63
5.1.3 Planificación de Cobertura.....	68
5.2 PLAN DE PRUEBAS	73
5.2.3 Escenarios de pruebas	74
5.2.4 Pruebas.....	75
5.2.3 Métricas de Evaluación.....	76
5.3 RESULTADOS Y ANÁLISIS	77
5.3.1 P1.1: Configuración de Filtros de Kalman	77
5.3.2 P1.2: Comparación de filtros FKE y FKU.....	83
5.3.2 P1.3: Posicionamiento con FKE	88
5.3.3 P2.1: Planeación de Vuelo.....	90
5.3.4 P2.2: Rutas de Cobertura con Algoritmos Heurísticos.....	97
5.3.5 P2.3: Mosaicos finales del Sensado remoto con las rutas de cobertura.....	102
5.4 RESUMEN.....	104
CONCLUSIONES Y TRABAJOS FUTUROS.....	105
6.1 CONCLUSIONES.....	105
6.2 TRABAJOS FUTUROS.....	106
BIBLIOGRAFÍA	109

Lista de Figuras

Figura 1. Ciclo de la Agricultura de Precisión.....	6
Figura 2. Revisión sistemática por base bibliográfica y pregunta de investigación ..	11
Figura 3. Componentes de AR Drone [95].....	18
Figura 4.Arquitectura de Control AR Drone [96]	19
Figura 5. Módulo de procesamiento en vuelo a) Raspberry Pi 2 B y b) Montaje en la plataforma aérea	20
Figura 6. Ejemplo de implementación de nodos en ROS.....	21
Figura 7. Sistemas de referencia para la navegación aérea	25
Figura 8. Sistemas de referencia para los sensores adheridos al cuerpo.....	26
Figura 9. Dinámica de un Cuadricóptero.....	28
Figura 10. Odometría básica de AR Drone.....	30
Figura 11. Estrategia de posicionamiento para AR Drone	39
Figura 12. Representación del área de cobertura como un grafo tipo rejilla.....	43
Figura 13. Configuración de búsqueda de vecinos a) Von Neumann y b) Moore.....	43
Figura 14. Procesamiento del modelo del mundo	44
Figura 15. Ruta de cobertura basada en Dijkstra	47
Figura 16. Propagación de frente de onda en un grafo	48
Figura 17. Ruta de cobertura basada en Transformada de Distancia	49
Figura 18. Casos de estudio para construir espirales en una región simple.....	51
Figura 19. Ruta de cobertura basada en Espirales	52
Figura 20. Ruta de cobertura basada en Dijkstra	53
Figura 21. Ruta de cobertura basada en Propagación de frente de onda.....	53
Figura 22. Ruta de cobertura basada en Espirales	54
Figura 23. Altura de Vuelo	56
Figura 24. Distancia entre puntos de paso	57
Figura 25. Procedimiento para ejecución de sensado remoto	62
Figura 26. Arquitectura de control para un sistema de sensado remoto basado en AR Drone	63
Figura 27. Grafo en ROS para estrategia de posicionamiento.....	65
Figura 28. Funcionalidades principales del código fuente desarrolladas	69
Figura 29. Escenario 1- CDU Universidad del Cauca - Popayán	74
Figura 30. Escenario 2- Cultivo de pasto gramillón – Vereda Guacas, Popayán.....	75
Figura 31. Estado estacionario de la IMU (a) Mediciones del acelerómetro y (b) Mediciones del giróscopo	78
Figura 32. Estado estacionario de la odometría básica de AR Drone (a) Posición x, (b) Posición y (c) Posición z	79
Figura 33. Estado estacionario del receptor GPS (a) Latitud y (b) Longitud.....	79
Figura 34. Odometría Básica (azul) y ruta esperada (negro)	81
Figura 35. Fusión IMU y Odometría.....	81

Figura 36. Fusión completa con GPS	82
Figura 37. Simulación de una ruta rectangular con FKE (a) vista en Rviz, (b) ruta en el plano y (c) ruta en altura	85
Figura 38. Simulación de una ruta rectangular con FKU (a) vista en Rviz, (b) ruta en el plano y (c) ruta en altura	86
Figura 39. Ruta rectangular experimental FKE y FKU en el plano	87
Figura 40. Ruta rectangular experimental FKE y FKU en altura	87
Figura 41. Ruta rectangular en el plano con FKE optimizado	89
Figura 42. Ruta rectangular en altura con FKE optimizado	89
Figura 43. Ruta circular en el plano con FKE optimizado	90
Figura 44. Deriva de AR Drone vs. Viento a Favor	91
Figura 45. Deriva de AR Drone vs. Viento en Contra	91
Figura 46. Deriva de AR Drone vs. Viento Trasversal	92
Figura 47. Vectores de Movimiento de AR Drone con respecto al viento	92
Figura 48. Rutas calculadas para un espacio libre de obstáculos (a) partiendo desde esquina Superior Izquierda (SI), (b) partiendo desde esquina Superior Derecha (SD), (c) partiendo desde esquina Inferior Izquierda (II) y (d) partiendo desde esquina Inferior Derecha (ID)	93
Figura 49. Rutas con viento del Norte	94
Figura 50. Rutas con viento del Sur	95
Figura 51. Rutas con viento del Este	95
Figura 52. Rutas con viento del Oeste	96
Figura 53. Rutas de Cobertura calculadas con DronePlanner (a) descoposicion de área en celdas, (b) ruta basada en propagación de frente de onda (c) ruta basada en Espirales y (d) ruta basada en Dijkstra	98
Figura 54. Parámetros resultantes calculados por DronePlaner para sensado remoto	99
Figura 55. Ruta de cobertura basada en la propagación de frente de onda	100
Figura 56. Ruta de cobertura basada en espirales	101
Figura 57. Ruta de cobertura basada en Dijkstra	101
Figura 58. Mosaico final con la ruta de cobertura basada en propagación de frente de onda	103
Figura 59. Mosaico final con la ruta de cobertura basada espirales	103
Figura 60. Mosaico final con la ruta de cobertura basada en Dijkstra	104

Lista de Tablas

Tabla 1. Clasificación de Vehículos Aéreos No Tripulados	8
Tabla 2. Brechas para cada enfoque de investigación	17
Tabla 3. Comparación de sistemas de navegación.....	33
Tabla 4. Comparación de características computacionales de los algoritmos de ruta de cobertura	54
Tabla 5. Nodos ROS en la optimización del posicionamiento	65
Tabla 6. Plan de Pruebas	76
Tabla 7. Covarianzas asociadas a Q	80
Tabla 8. Covarianzas asociadas a R	80
Tabla 9. Variables de estado de cada fuente de información.....	80
Tabla 10. Precisión de los filtros y cambios en las matrices de ruido	82
Tabla 11. Covarianzas asociadas a Q.....	83
Tabla 12. Covarianzas asociadas a R.....	83
Tabla 13. Resultados de posición inicial vs. Tiempo de ruta	97
Tabla 14. Parámetros de evaluación de cobertura calculada con DronePlanner	99
Tabla 15. Parámetros de evaluación de las rutas de cobertura real	102

Introducción

La investigación para la implementación de un sistema de *sensado remoto* para la agricultura de precisión inicia con la identificación de los retos que existen con los denominados *drones* para llevarlos a las prácticas agrícolas. Estos retos enmarcan el planteamiento del problema, desde el cual se plantea una pregunta de investigación y por ende los objetivos con los cuales se va a responder la pregunta. Este capítulo, además describe las principales contribuciones de la investigación finalizada y guía al lector en el contenido de la monografía con un resumen de cada uno de los capítulos subsecuentes.

1.1 Planteamiento de problema

La agricultura ha llegado a ser uno de los recientes enfoques en investigación de los países latinoamericanos debido a las prospectivas planteadas en los planes de desarrollo económico, social y cultural que exige la dinámica actual del mundo, donde el manejo óptimo de los recursos naturales es una estrategia central [1]. Colombia por su parte ha identificado el potencial del sector agropecuario y le apuesta a incrementar los procesos de innovación en los sistemas productivos con miras a mejorar la calidad, reducir los costos de producción y elevar la producción agroalimentaria. Con la misma visión, el Cauca ha acogido estos planteamientos en su plan estratégico departamental de Ciencia, Tecnología e Innovación CONCIENCIA-Cauca, donde se declara como principal fuente económica a la agricultura. Estos escenarios han motivado a la concentración de esfuerzos para implementar global y localmente lo que hoy es conocido como agricultura de precisión. Esta concepción tecnológica sostiene la idea de hacer lo correcto en el lugar correcto y en el tiempo correcto; incluye todas las prácticas de producción que usen tecnologías de la información y comunicación que permitan decidir sobre la información obtenida y monitorizar regularmente variables en los cultivos. Las tecnologías que forman parte de la agricultura de precisión incluyen redes de sensores inalámbricos, sistemas de posicionamiento global y *sensado remoto*¹. El *sensado remoto* en particular tiene gran auge en la agricultura, ya que permite obtener información detallada del crecimiento, deterioro y evolución del cultivo, de una forma confiable y en localizaciones específicas, sin importar lo lejanas que puedan estar.

¹ El documento tomará el término *sensado remoto* para nombrar al método denominado en inglés *remote sensing* debido a que no fueron encontradas traducciones acordes a los propósitos del trabajo

En los últimos años la investigación en *sensado remoto* para agricultura se ha concentrado en el análisis de imágenes satelitales e imágenes aéreas desde aviones tripulados; y aunque los resultados proporcionan estudios altamente especializados del cultivo, estas alternativas presentan resolución espacial limitada, dependencia de condiciones atmosféricas y no están al alcance de los pequeños y medianos productores, ya sea por el costo de la información y su procesamiento o por la poca oferta de estos servicios. Estas restricciones del *sensado remoto* satelital y aéreo han sido superadas por robots aéreos de bajo costo, denominados drones, que acoplan cámaras de alta definición y múltiples sensores en plataformas pequeñas, permitiendo adquirir información en cualquier lugar y momento. Sin embargo, los estudios de *sensado remoto* para cultivos utilizando drones continúa en desarrollo, ya que, a pesar del bajo costo, alta definición de la información y fácil manejo, estos vehículos aéreos tienen poca autonomía y en caso de fallas es inevitable pérdidas de información, además de que la configuración de los vehículos aéreos comerciales no considera el entorno del área de cobertura.

De acuerdo con lo anterior y con los requerimientos para fortalecer un sector agrícola competitivo en la región, tales como la detección temprana de enfermedades, la monitorización constante de suelos y el registro permanente de la evolución de los cultivos, se plantea la presente propuesta de investigación la cual tiene su origen en la siguiente pregunta de investigación:

¿Cómo monitorizar un cultivo agrícola mediante un dron que permita obtener información confiable, considerando su plataforma y características del área de cobertura?

1.2 Escenario de motivación

La agricultura de precisión abrió nuevas alternativas para los productores con métodos de producción que optimizan los recursos, lo cual lleva a incrementar la sostenibilidad, rentabilidad y conservación del medio ambiente. Para lograr dicha meta se desarrollan tecnologías que permiten actividades como la detección de enfermedades y plagas, pronósticos de evolución de las plantas, mediciones de campo, monitorización constante de suelos, entre otras. El método más utilizado es el *sensado remoto*, que actualmente se apoya en vehículos aéreos no tripulados para minimizar la inversión en obtención y procesamiento de información, acercando competitivamente a medianos y grandes productores.

Sin embargo, los vehículos aéreos no tripulados o *drones* como plataforma de *sensado remoto*, están diseñados para entornos controlados, es decir, terrenos planos y condiciones atmosféricas estándar, por tal razón la sensibilidad de los sensores portados en el dron y la planificación de vuelo de estas aeronaves deben ser adaptadas a las prácticas agrícolas en cultivos del trópico, como es el caso de Colombia. De acuerdo a la problemática anterior se considera necesario implementar un sistema de *sensado remoto* basado en vehículos aéreos no tripulados de bajo costo, que precise su posicionamiento en ambientes externos y diseñe un plan de

vuelo óptimo, con el fin de obtener información confiable y completa de un cultivo, teniendo en cuenta las características particulares de los cultivos agrícolas.

1.3 Objetivos

1.3.1 Objetivo General

Implementar un sistema de *sensado remoto* para agricultura de precisión soportado en un robot aéreo multirroto de bajo costo (*drone*).

1.3.2 Objetivos Específicos

- Desarrollar una estrategia para el posicionamiento del *drone* en ambientes abiertos mediante la integración de sensores y unidades de procesamiento externo.
- Evaluar dos o más algoritmos de cobertura para generar planes de vuelo sujetos a las capacidades del cuadricóptero y las características del cultivo.
- Establecer un mecanismo en el *drone* para capturar imágenes de manera automática.
- Implementar y evaluar los mecanismos propuestos en la construcción de un sistema de *sensado remoto* para un cultivo agrícola de la región

1.4 Contribuciones

Las principales contribuciones de este proyecto de investigación son:

- Un conjunto de módulos de desarrollo para navegación y control del cuadricóptero AR Drone soportados en el sistema operativo ROS.
- Un conjunto de algoritmos de planificación de cobertura para ambientes abiertos, los cuales fueron evaluados durante la ejecución del proyecto.
- Una estrategia de *sensado remoto* utilizando una plataforma de bajo costo como AR Drone para la obtención información confiable.
- Un artículo expuesto en VII Congreso Iberoamericano de Telemática realizado los días 10,11 y 12 de junio de 2015 en Popayán Colombia, titulado *Remote Sensing for Agricultural Crops Based on a Low Cost Quadcopter*. Con indexación C por Colciencias, el cual presenta la arquitectura de desarrollo del proyecto declarado y primeras pruebas experimentales que lo soportan.
- Un artículo expuesto en Workshop of Physical Agents 2016 realizado los días 16 y 17 de junio de 2016 en Malaga España, titulado *An Aerial Autonomous Robot for Complete Coverage Outdoors*. El cual presenta resultados finales del trabajo de investigación.

- Un artículo sometido a evaluación en la revista Escuela de ingenierías de Antioquía, Medellín, titulado *Remote Sensing based on Quadrotors for Precision Agriculture: A review*. Con una indexación A2 por Colciencias, el cual presenta un mapeo y revisión sistemática de la bibliografía que soporta el desarrollo del sistema de *sensado remoto* propuesto.

1.5 Contenido de monografía

La monografía presentada en este documento se compone de seis capítulos presentados a continuación:

Capítulo 2. Estado actual del conocimiento

Describe de manera general las bases conceptuales, trabajos relacionados que aborda el proyecto de investigación, brechas de investigación encontradas en los trabajos relacionados y finalmente una descripción de la tecnología en la que se apoya la propuesta.

Capítulo 3. Posicionamiento del Drone en Ambientes Abiertos

Presenta los fundamentos para comprender la navegación de robots aéreos multirrotores en entornos abiertos, con la cual se establece una estrategia de posicionamiento desarrollada en el sistema operativo ROS y basada en filtros de Kalman extendidos.

Capítulo 4. Planificación de Vuelo para Sensado Remoto de Cultivos

Relata la estrategia de planificación de cobertura mediante algoritmos heurísticos y descomposición aproximada de área. La misión de *sensado remoto* está conformada por la ejecución de la ruta seleccionada y por la captura de fotografías automática considerando las características del AR Drone y la cámara.

Capítulo 5. Experimentación y Evaluación.

Expone el prototipo desarrollado, el proceso de pruebas y evaluación y el análisis de los resultados con respecto a los objetivos.

Capítulo 6. Conclusiones y trabajos futuros

Detalla las principales contribuciones obtenidas en la ejecución del proyecto y se plantean algunas ideas para el desarrollo de trabajos futuros.

Estado Actual del Conocimiento

El contexto de la presente investigación está enmarcado por el requerimiento inherente de la agricultura de precisión para adquirir información completa y confiable de los cultivos. La recolección de datos en la agricultura de precisión abarca un amplio abanico de métodos, sin embargo, es el *sensado remoto* la técnica mayormente utilizada, puesto que se puede monitorizar áreas extensas en detalle y no interviene con el medio ambiente. Dentro de las tecnologías utilizadas para *sensado remoto* se encuentran popularizados los satélites, los cuales catapultaron en su momento la agricultura de precisión. No obstante, es una alternativa costosa y dependiente de las condiciones atmosféricas; lo cual no sucede con los vehículos aéreos no tripulados o *drones*, ya que permiten desplegar un sistema de *sensado remoto* de bajo costo y óptimo en resolución espacial, temporal y espectral, sin importar las condiciones atmosféricas.

De acuerdo al contexto, se describe, el estado del arte encontrado para optimizar el posicionamiento de un robot aéreo multirroto, ejecutar misiones de cobertura aérea y realizar *sensado remoto* usando drones, los cuales abordan directamente cada objetivo planteado. El estudio de los trabajos relacionados concluye en la identificación de brechas de investigación, para establecer los mayores aportes del trabajo. El capítulo finaliza con la descripción del robot aéreo multirroto y las herramientas hardware y software seleccionadas para la implementación de un sistema de *sensado remoto* aplicado a la agricultura de precisión.

2.1 Conceptos generales

2.1.1 Agricultura de precisión

El surgimiento de la tendencia en agricultura de precisión fue necesario debido al crecimiento en la extensión de los cultivos y la producción intensiva del siglo XX, donde los tratamientos manuales y mecánicos implicaban gran cantidad de mano de obra, alta inversión en fertilizantes, abonos, pesticidas, agua, etc. y dispendiosos análisis de datos; con pocas posibilidades de ejecutar acciones efectivas para todo el cultivo. En este contexto era costoso realizar monitorización de la variabilidad espacial y temporal de los cultivos. Con el establecimiento de un sistema de posicionamiento global (GPS) basado en un grupo de satélites colocados en órbita por el Departamento de Defensa Estadounidense en la década de los setenta, fue posible recolectar información localizada del estado de los cultivos, consolidando de esta manera el concepto de agricultura de precisión [2]. El concepto apoya la idea de

hacer lo correcto, en el lugar correcto y en el tiempo correcto. Formalmente la agricultura de precisión es conocida como la gestión de la agricultura en sitios específicos de una manera automatizada, usando tecnologías de la información y la comunicación [3]. Aunque en la actualidad la agricultura de precisión incluye el aprovechamiento de las tecnologías emergentes y convergentes, como los sistemas micro electromecánicos, redes de sensores inalámbricos, sistemas de información geográfica, sistemas de posicionamiento global, control automático, *sensado remoto*, computación móvil y telecomunicaciones, con el propósito de obtener mayor información que le permita al agricultor registrar temporal y espacialmente los cultivos, tratar enfermedades en zonas puntuales y distribuir eficientemente los recursos, considerando la sostenibilidad del medio ambiente [4].

La ejecución de las técnicas de agricultura de precisión sucede en tres fases generales (figura 1): Recolección de los datos, que incluye el mapeo de la variabilidad del suelo, la planta y el medio ambiente; Interpretación de los datos mediante mapas de rendimiento, nutrientes e isosalinidad, que resulta en un esquema de toma de decisiones y la fase final de Aplicación que gestiona el riego y fertilizantes en las prácticas sobre el cultivo [2]. Por ejemplo, para la recolección de datos es frecuente el uso de tecnologías como las WSN y el *sensado remoto*. Las técnicas de *sensado remoto* permiten recolectar datos útiles en el proceso de la agricultura de precisión mediante sensores a distancia.

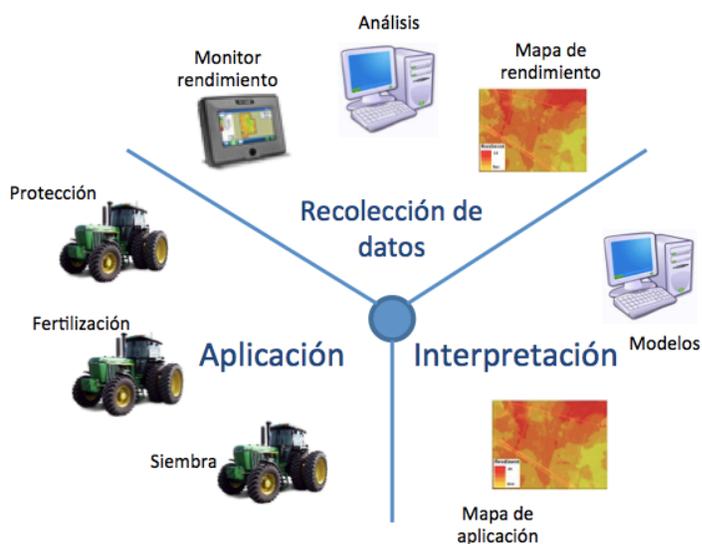


Figura 1. Ciclo de la Agricultura de Precisión

2.1.2 *Sensado remoto*

De acuerdo a la sociedad americana de meteorología *sensado remoto* es un método para obtener información de las propiedades de un objeto o fenómeno sin contacto directo con dispositivos tales como cámaras, láseres, receptores de radiofrecuencia, radares, sonares, magnetómetros o sensores que obtengan información mediante la medición de ondas electromagnéticas o acústicas emitidas o reflejadas por los

objetos o fenómenos [5]. Con base en esto, un sistema de *sensado remoto* puede ser: pasivo, cuando la información recolectada depende de una fuente externa que irradia el objetivo, por ejemplo el *sensado remoto* en el espectro visible, el sol es la fuente y las ondas reflejadas por el objetivo son obtenidas por los sensores, o activo cuando la información recolectada depende tanto de los sensores como de la fuente para obtener las ondas reflejadas, por ejemplo un radar transmite pulsos de microondas desde una antena transmisora al objetivo y recibe la respuesta en una antena receptora.

Existen tres tipos de plataformas en las que se fundamenta un sistema de *sensado remoto*: Plataforma de observación en la tierra, plataforma de observación aérea y plataforma de observación espacial. Para la primera opción los sensores deben colocarse alrededor de 15 metros por encima del suelo. Cuando la observación es aérea, las plataformas pueden ser aviones, drones o globos; los cuales soportan la fotogrametría (fotografía aérea), bajo estas circunstancias los sensores pueden estar a 100 m, o más, por encima del suelo. La última plataforma mencionada incluye satélites, situados y estabilizados en una órbita determinada [5]. La información recolectada por las plataformas aéreas y satelital generalmente queda almacenada como una imagen. Para tratar el *sensado remoto* basado en imágenes es necesario establecer la escala. Las imágenes de gran escala presentan mayor precisión que las imágenes de escala pequeña. La precisión de una imagen está determinada por las resoluciones: espacial, espectral y temporal. La primera está determinada por la distancia mínima entre dos objetos que un sensor puede distinguir. La segunda está determinada por el ancho de banda de los canales usados; a mayor ancho de banda, mejor reconocimiento de objetos. Por último, la temporal está relacionada con los intervalos de tiempo entre dos visitas sucesivas a un escenario particular [6].

Las resoluciones espaciales, espectrales y temporales determinan el campo de aplicación de las técnicas de *sensado remoto*. La mayoría de casos convergen en aplicaciones para medio ambiente y a la agricultura, ya que estos campos de aplicación requieren la adquisición, procesamiento e interpretación de datos de grandes aéreas de una manera precisa y efectiva. Las imágenes obtenidas describen parámetros como la cobertura del cultivo, la salud de las plantas y la humedad del suelo, los cuales son útiles en la construcción de mapas de estrés, fertilización y control de irrigación [7]. Para obtener dichos parámetros en los cultivos son analizadas las variaciones en el espectro electromagnético de los objetos o fenómenos, por ejemplo, para la caracterización de la vegetación se combinan las reflectancias del rojo e infrarrojo para generar índices de vegetación como: el índice de vegetación de diferencia normalizada (NDVI), índice de área de hoja (LAI) y fracción de radiación activa absorbida fotosintéticamente, los cuales permiten crear modelos de las condiciones del cultivo.

2.1.3 Vehículos aéreos no tripulados

Los vehículos aéreos no tripulados o *drones* son definidos de varias formas por los diferentes departamentos militares y de vigilancia en los Estados Unidos, Europa, Canadá y otros, determinando su concepción de acuerdo al enfoque, al diseño y funcionalidad. Sin embargo, todas concluyen en decir que es una aeronave

propulsada, sin piloto o pasajeros, que puede ser operada remota o autónomamente. Generalmente excluye globos, planeadores y cometas [8].

La popularidad de las aeronaves no tripuladas nace de la ingeniería militar de los Estados Unidos en la primera guerra mundial, cuando fue creado un torpedo aéreo (Hewitt) en 1917. Desde entonces surgieron esfuerzos por la construcción de vehículos no tripulados para portar bombas y seguir los movimientos del enemigo, llegando a obtener el JH-1, un UAV desarrollado por el laboratorio de investigación naval, pilotado remotamente a 11 millas de distancia [9] [10].

Clase	Categoría	Desarrollo	Altura de Operación	Radio de Misión	Plataforma de ejemplo
Clase I < 150 kg	Micro < 2 kg	Operaciones tácticas individuales o de pelotón	Por encima de 200 pies sobre el suelo	5 km	Black widow
	Mini 2-20 kg	Operaciones tácticas de sub-unidad (lanzamiento manual)	Por encima de 3000 pies sobre el suelo	25 km	Skylark
	Pequeño >20 kg	Operaciones tácticas de una unidad	Por encima de 5000 pies sobre el suelo	50 km	Hermes 90
Clase II 150-600 kg	Táctico	Formación táctica	Por encima de 10000 pies sobre el suelo	200 km	Aerostar
Clase III >600 kg	MALE	Operación local	Por encima de 45000 pies sobre el suelo	Sin limite	Predator A & B
	HALE	Estrategia nacional	Por encima de 45000 pies sobre el suelo	Sin limite	Global Hawck
	Ataque	Estrategia nacional	Por encima de 45000 pies sobre el suelo	Sin limite	

Tabla 1. Clasificación de Vehículos Aéreos No Tripulados

Los vehículos aéreos no tripulados están clasificados por la mayoría de los países que los acogen en sus reglamentos de aviación tanto para uso civil como uso militar. En las diferentes clasificaciones frecuentemente está considerado el peso, la velocidad, el área de operación y altura de vuelo, el enfoque depende de la organización administrativa de aviación de cada país. Por ejemplo, la tabla 1 reúne la gran mayoría de los criterios de clasificación internacionales y es propuesta por el Panel de vehículos aéreos no tripulados (JUAVP) y el Grupo de capacidades sobre vehículos aéreos no tripulados (JCGUAV) [11]. De esta clasificación, la categoría de Micro Vehículos Aéreos (MAV), encuentra un abanico de aplicaciones comerciales como control de tráfico vehicular [12], rescate [13] y labores agrícolas [14]. Algunas de estas aplicaciones cuentan con MAVs inspirados en el funcionamiento de los helicópteros, pero con múltiples rotores, denominados robots multirrotores, como los cuadricópteros, ya que presentan mejor estabilidad, modelado físico sencillo y fácil maniobrabilidad. Para propósitos de este estudio son de interés los trabajos realizados con cuadricópteros.

Las aeronaves multirrotor tienen la capacidad de despegar y aterrizar de manera vertical (VTOL), el control de la plataforma está basado en la variación de la velocidad de cuatro rotores adheridos a una cruz simétrica. Para evitar la inestabilidad, es necesario que la estructura sea lo suficientemente rígida, manteniendo un peso liviano. Un *drone* de este tipo está compuesto por un microcontrolador, controladores para los motores, sistema de navegación inercial, sistema de comunicación entre el *drone* y una estación en tierra y una fuente energética [15]. El sistema de navegación mide los movimientos inerciales del *drone* y está conformado por una unidad de movimiento inercial (*IMU*, por sus siglas en inglés), adicionalmente se utiliza un GPS (Sistema de posicionamiento global) para el posicionamiento georreferenciado y sensores de presión para medir la altitud [16]. Los movimientos relativos de un cuadricóptero están descritos matemáticamente por seis grados de libertad x , y , z , $roll$, $pitch$ y yaw ($roll$ hace referencia a la rotación alrededor del eje Y , $pitch$ a la rotación alrededor del eje X , y yaw a la rotación alrededor del eje Z). La combinación de estas variables permite la manipulación de las fuerzas en los rotores para obtener un torque balanceado. De esta manera, para mantener el *drone* suspendido, todos los rotores deben tener la misma potencia, y para desplazarse en el eje vertical hacia arriba o hacia abajo, todos los rotores reducen o aumentan la potencia. Para los demás movimientos (giros y movimientos hacia la derecha y hacia la izquierda, hacia delante y hacia atrás) la potencia es distribuida en los rotores de tal manera que genere un diferencial del torque total orientado hacia la dirección requerida [17].

2.2 Trabajos relacionados

La metodología para realizar la revisión bibliográfica está basada en un estudio de mapeo sistemático en ingeniería del software propuesto en [18]. Este método considera la variabilidad de los productos software y propone los pasos del proceso para la construcción de un mapeo sistemático, abarcando la definición de las preguntas de investigación, búsqueda de publicaciones relevantes, proyección de los hallazgos, revisión de resúmenes, extracción de datos y finalmente el mapeo.

En la primera etapa de la metodología fueron planteadas cuatro preguntas de investigación desde las cuales parte la revisión bibliográfica:

- ¿Qué cuadricópteros de desarrollo han sido utilizados para investigación?
- ¿Cuáles son las estrategias para el posicionamiento de cuadricópteros?
- ¿Cuáles son las estrategias de cobertura de grandes áreas con robots aéreos?
- ¿Qué sistemas de sensado remoto basados en robots aéreos multirrotor son encontrados para AP?

El siguiente paso es iniciar la búsqueda de las publicaciones, para lo cual se propone la construcción de cadenas de búsqueda basadas en las preguntas anteriores. Las cadenas de búsqueda utilizadas fueron construidas con las siguientes palabras claves: *agriculture, remote sensing, photogrammetry, UAV, quadcopter, quadrotor, aerial robot, coverage, path planning*.

Con las anteriores palabras claves fue obtenida una lista amplia de publicaciones que revelaron la importancia del tema y la manera contundente como ha sido abordado en investigaciones. Sin embargo, la metodología propone filtrar los hallazgos mediante criterios de inclusión y exclusión para obtener un conjunto certero de publicaciones que aporten a la construcción del mapeo. Para este paso se tienen los siguientes criterios:

- Fueron incluidos libros, artículos de revistas, reportes técnicos y trabajos finales de grado.
- El rango de años de publicación predominantemente esta entre 2007 y 2014.
- Fueron incluidas publicaciones de las siguientes bases de datos: Science Direct, IEEE Xplore, Springer y Google.
- No están incluidas publicaciones con vehículos aéreos de ala fija.

Continuando con la metodología, es propuesto un esquema de clasificación de la bibliografía resultante de los criterios de inclusión y exclusión. Este esquema es planteado para estructurar el contenido los trabajos relacionados: *Posicionamiento de drones en ambientes externos, planificadores de ruta de cobertura y sistemas de sensado remoto con drones*.

Con el esquema relacionado, es definida la etapa final de la construcción del mapeo sistemático en la cual es realizada una extracción de datos del anterior esquema de clasificación con todos los hallazgos presentados en diagramas de frecuencias correspondientes a cada tema. De esta manera es posible analizar aportes a la investigación y brechas en los trabajos revisados. La figura 1 evidencia la revisión por bases bibliográficas y preguntas de investigación identificadas con un color particular, para un total de 130 publicaciones evaluadas.

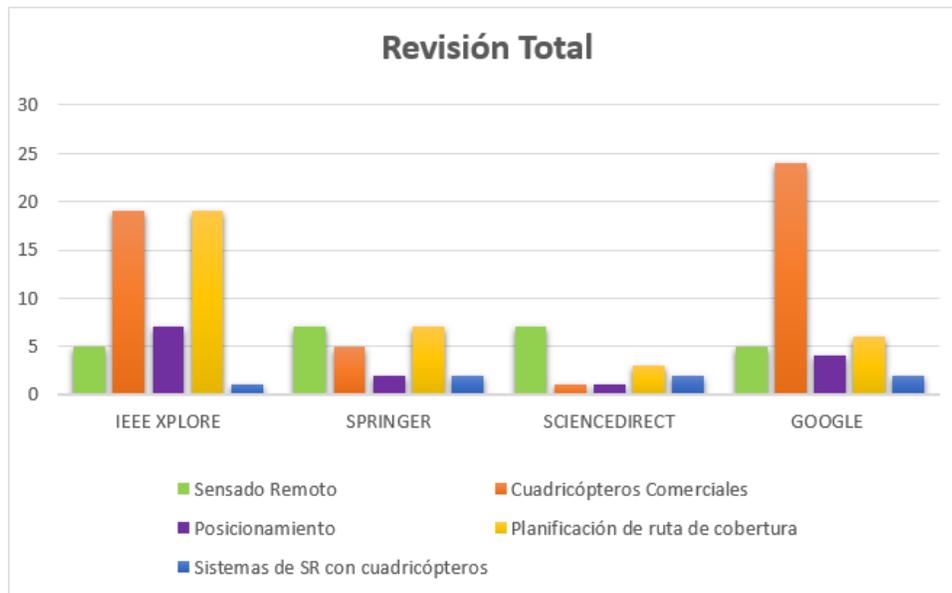


Figura 2. Revisión sistemática por base bibliográfica y pregunta de investigación

2.2.1 Cuadricópteros para Investigación

Los cuadricópteros representan una de las mejores opciones para los investigadores por su capacidad de carga y fácil manejo, a pesar de su elevado consumo de potencia [19]. En la literatura revisada son consideradas plataformas de desarrollo como el Arducopter para aplicaciones civiles como acompañamiento de personas o diseño de controladores. El atractivo de esta plataforma para la investigación es su autopiloto compatible con Arduino llamado Ardupilot [20] [21] [22].

El Draganflyer es una de las primeras plataformas comerciales que provee facilidades para incorporar desarrollos en investigación, tales como análisis de fenómenos naturales en periodos de tiempo [23] [24] [25]. Otro cuadricóptero incluido en investigaciones es el Mikrokopter con el que han sido realizados trabajos de mapeo en 3D, registro de parámetros por múltiples sensores portados en el MAV y otros trabajos para analizar y mejorar el funcionamiento de la plataforma [26] [27].

Dentro de las plataformas de fuente abierta, quizás inspiradas en el proyecto Arducopter han sido encontrados proyectos como AeroQuad y Parrot AR.Drone. El AeroQuad es una nueva alternativa con la cual ha sido posible integrar tecnologías como Smartphones y Kinect [28] [29]. Sin embargo, el Ar Drone ha sido masivamente utilizado desde su primera versión tanto por aficionados como por investigadores debido a que añade técnicas de visión y receptores GPS para el control del drone en cualquier ambiente [30]. Debido a que el Ar Drone es el resultado de un proyecto de carácter público, alrededor de éste han sido generados estudios sobre métodos alternativos para la navegación, estabilidad y autonomía. Sus aplicaciones incluyen seguimiento y detección de objetos en espacios abiertos, vigilancia de edificios, monitoreo ambiental y coordinación de enjambres [31] [32] [33] [34] [35] [36]. De

acuerdo a los trabajos de investigación, se observa una tendencia predilecta por el cuadricóptero Ar Drone.

2.2.2 Posicionamiento de *drones* en ambientes externos

El posicionamiento de un robot aéreo multirrotor toma como base la unidad de movimiento inercial, la cual está conformada por un grupo de giroscopios, magnetómetros y acelerómetros que miden el cambio en la posición, velocidad o aceleración determinados por los seis grados de libertad correspondientes. Estos valores son comparados con la posición previa para calcular la posición actual, a través de estrategias de control lineales como PID (Proporcional-Integral-Derivativo). El control PID es aplicable a las variables involucradas en el posicionamiento, es decir, a los ángulos roll y pitch [37] [38]; generalmente se utiliza un regulador LQR para filtrar los errores provocados por la inestabilidad del *drone* [39]. Aunque los resultados del posicionamiento con los métodos tradicionales son aceptables, es posible optimizarlo mediante técnicas como *backstepping*², garantizando la estabilidad asintótica y la robustez [40] [41].

Además del control que debe ser realizado en la posición y orientación de un cuadricóptero, existen diferentes sensores utilizados para la medición de la altitud (sonares, ultrasonido, barómetros, etc.) para los cuales también aplica el control PID [42]; aunque no es la única alternativa, el control robusto adaptativo y el control PID neuro-adaptativo permiten reducir significativamente los errores en la altura. Por otro lado, es posible encontrar combinaciones del controlador PID con el Filtro de Kalman Extendido (FKE), permitiendo la convergencia de los sensores y estimando el posicionamiento del *drone* eficientemente [43] [44].

Para que un *drone* navegue en entornos abiertos no basta con los sensores de la IMU y altímetros, es requerido un receptor GPS, el cual convierte señales de satélite en posición, velocidad y altura, necesita mínimo cuatro señales satelitales para determinar la posición con respecto a la tierra. Algunos trabajos desarrollan controladores PID combinado con FKE para dotar de precisión el posicionamiento en exteriores [45]. Ciertamente el posicionamiento basado en GPS implica errores cuando hay pérdida de señal, algunos investigadores promueven la idea de tener dos o tres sensores observables para diferentes satélites con el fin de precisar de mejor manera las mediciones. *Kis* y *Lantos* exponen la implementación de dos filtros de *Kalman* extendidos usando más de un receptor y sensores de navegación inercial para fijar el posicionamiento [46]. De acuerdo a lo anterior es relevante mencionar que el posicionamiento de cuadricópteros con GPS integrado dependen de la precisión de los receptores, generalmente con aciertos del 90% en la estimación de la longitud y latitud, y en la estimación de la altura un error de alrededor de 12 metros, por lo que no son recomendables si es requerida precisión en altitud. Si es considerada la altura como elemento clave en el posicionamiento en lugares abiertos, algunos autores combinan las fortalezas de este tipo de receptores (GPS) y los sensores de altura (barómetros o sensores de ultrasonido). Los resultados arrojan

² Es una técnica que es basada en una estructura recursiva adicional a los sistemas no lineales para mejorar su estabilidad.

una estimación confiable de la altura como lo presentan *Chee* y *Zhong*, quienes obtienen una desviación estándar de 0.07 metros en las medidas y una esfera de radio 5 metros para el posicionamiento con GPS [47]. La principal restricción de este tipo de posicionamiento es que los sensores basados en presión y ondas acústicas dependen de las condiciones atmosféricas, aunque su precisión suele ser menor a un metro.

Adicional a las técnicas anteriores para control de la navegación de los cuadricópteros, investigaciones plantean la navegación de un *drone* basada en visión, usando una o dos cámaras con alta resolución espacial mediante técnicas de localización y mapeo conocidas como SLAM [48] [49]. Además, es posible fusionar los receptores GPS con visión, usando una o dos cámaras de alta resolución espacial, sin embargo, la opción es cuestionada debido a que el procesamiento de las medidas del GPS en conjunto con imágenes suele ser costoso [50].

2.2.3 Planificadores de Ruta de Cobertura

La tarea de planificación de ruta para robots móviles como los vehículos aéreos multirrotores es diseñada mediante algoritmos que consideran el área de cobertura, obstáculos y sensores portados en la plataforma [51]. La determinación de la ruta de cobertura es posible enfocarla para entornos desconocidos con algoritmos basados en heurística y la aleatoriedad, en cuyo caso el robot es equipado con un conjunto de comportamientos básicos para luego tomar acciones más complicadas de manera jerárquica. Este tipo de algoritmos son la base de las capacidades de búsqueda de los robots móviles. Generalmente incluyen algoritmos bio-inspirados por ejemplo en el comportamiento de las hormigas [52] [53], en este tipo de algoritmos los robots dejan marcas a su paso como lo harían las hormigas con sus feromonas. En la misma tendencia de robots exploratorios se encuentran las redes neuronales que mediante el registro de la dinámica del entorno es posible representar el área de cobertura y calcular los movimientos [54] [55], en conjunto con los anteriores existen algoritmos evolutivos que otorgan mayor complejidad en la navegación de entornos desconocidos [56] [57].

Otro enfoque incluye algoritmos que consideran entornos conocidos con sus respectivos límites y obstáculos. Esta alternativa de planificación deliberativa considera factores del entorno y en ocasiones del robot, llevando a diseñar algoritmos que permitan optimizar los recursos. Dichos algoritmos generalmente dividen el área de cobertura ya sea por descomposición en rejillas (cuadrículas) o por descomposición exacta. Esta última técnica ha evolucionado con el enfoque *boustrophedon*, que permite minimizar los movimientos del robot [58]. De esta manera son identificados los puntos del área por los cuales puede pasar el robot relacionando el punto medio de las celdas como las posiciones de referencia, determinando la mejor ruta de cobertura considerando las restricciones. La descomposición de área basada en rejilla o descomposición aproximada como lo indica *Choset*, es combinada con algoritmos como por ejemplo el algoritmo de retroceso en espiral y sus modificaciones, el cual completa las rutas en espiral de manera que la cobertura sea completa, en el proceso las regiones no visitadas son marcadas y cubiertas por mecanismos de retroceso [59]. Por otro lado, la cobertura

resultante de técnicas de descomposición del aérea en rejillas iguales y la reducción del *problema del agente viajero* han aportado en la planificación de ruta para robots móviles bajo el agua como lo señala Morin et al, el algoritmo planteado toma los puntos de referencia de cada rejilla y las distancias entre ellas para calcular la mejor ruta que visite una sola vez cada celda y regrese al punto de partida [60]. Para complementar la revisión de esta categoría de algoritmos existe la técnica del *árbol de expansión mínima* que utiliza la descomposición aproximada. Este algoritmo es popular entre los generadores de rutas de cobertura y consiste en tomar las aristas del mapa de rejillas de área de cobertura para formar un árbol que será expandido de acuerdo a pesos propuestos previamente en el aérea [61]. También es posible encontrar trabajos con el denominado algoritmo de frente de onda (transformada de distancia), el cual asigna un cero (0) a la meta y un uno (1) al resto de celdas, a medida que es visitada una celda, un dos (2) es asignado, este proceso es repetitivo hasta que el frente de onda llegue al punto inicial [62].

Los trabajos revisados que usan la descomposición exacta de celdas, incluyen algoritmos como el árbol en expansión para generar las rutas de robots que realizan movimientos hacia adelante y hacia atrás [63]. La descomposición también es combinada con el concepto de entropía de rama del árbol, construido para ayudar, por ejemplo, a un vehículo autónomo subacuático para alcanzar sus objetivos [64]. Otras investigaciones retoman algoritmos genéticos y bis-inspirados para optimizar la distancia y el tiempo [65] [66].

En particular para la planificación de cobertura con robots aéreos multirrotores se retoma el método *boustrophedon*, mencionado anteriormente, adecuado para ambientes complejos; en estos estudios tratan el problema de planificación de cobertura con más de un vehículo, extendiendo las capacidades de vuelo y la obtención de información [67] y también consideran el método basado en rejillas que en conjunto con la estrategia de búsqueda con grafos [68] [69]. Una estrategia que llama la atención es el uso de algoritmos probabilísticos que dependiendo de los sensores de la plataforma es posible determinar la mejor ruta considerando ponderaciones de diferentes rutas posibles [70]. También se encuentran trabajos basados en el algoritmo de expansión de árbol rápida aleatoria que permite la generación de rutas en tiempo real [71] [72]. De manera más compleja es posible encontrar la combinación entre un comportamiento estratégico que incluyen un conocimiento previo del área para generar trayectorias mediante algoritmo de frente de onda y un comportamiento reactivo que utiliza estrategias como ocupación tridimensional de rejillas bayesianas, con el propósito de equilibrar en una sola plataforma las dos tendencias [73]. Uno de los enfoques más recientes para planificación de ruta de un robot aéreo tipo cuadricóptero es el uso de algoritmos de búsqueda heurística de grafos tales como el *A o *D. A pesar que maneja alta complejidad, son considerados por que no requiere chequeos adicionales de colisiones durante la planificación, el reto más importante es la construcción previa del mapa de aérea, para lo cual es posible utilizar técnicas como el estado de rejilla y búsqueda con grafos [74] [75].

La cooperación y la coordinación de *drones* es el foco de múltiples investigaciones, ya que es posible tener mayor cobertura y flexibilidad en la planificación de rutas. La mayoría de aplicaciones se orientan a la distribución de sub-áreas para cada *drone*,

de acuerdo a las estrategias mencionadas anteriormente [76]. Siguiendo esta línea Joao Valente describe en su tesis un desarrollo relevante, en el que incluye optimización de los algoritmos basados en descomposición aproximada con el objeto de construir mosaicos de cultivos. Además de describir el despliegue de tres *drones* en un cultivo, propone en su investigación un trabajo colaborativo con robots terrestres para complementar las tareas de monitorización [77]. La mayoría de trabajos recientes consideran la cobertura con grupos de *drones* para desarrollar estrategias inspiradas en la naturaleza, por ejemplo para cuadricópteros se encuentran investigaciones alrededor de comportamientos de enjambre de insectos, bandada de aves (líder y seguidores) o grupo de peces [78] [79] [80] [81] [82] [83].

2.2.4 Sistemas de *Sensado remoto con Drones*

Los drones fueron diseñados inicialmente para supervisión de campos en el ámbito militar, sin embargo, son las plataformas más utilizadas actualmente para aplicaciones de *sensado remoto*, ya que permiten construir mapas de cualquier objeto o fenómeno mediante cámaras de alta resolución, portadas en drones comercialmente disponibles; los cuales navegan basados en puntos georreferenciados predefinidos [84]. Entre las aplicaciones más nombradas que usan *sensado remoto* para detallar propiedades, se encuentran la arqueología [85], el control medio ambiental [86], topografía [87] y por su puesto la agricultura de precisión; para la cual se encuentran trabajos en gestión de viñedos, como el desarrollado en el centro de Italia con un Mikropkopter de seis rotores modificado y denominado VIPtero. El robot cuenta con 10 minutos de vuelo, es controlado con un transmisor 2.4 Ghz, la actualización de firmware es realizada mediante BlueTooth y la cámara multi-espectral portada en la plataforma es de 3.2 Megapíxeles con sensor CMOS. La principal motivación de este sistema es la obtención de una gran cantidad de índices de vegetación como NDVI, índice de vegetación ajustada al suelo, segmentación del follaje, entre otros. Estos índices se obtienen después de un procesamiento en las imágenes capturadas por el *drone*, mediante una ruta de cobertura pre programada a una altura fija de 150 metros. Los resultados de este trabajo señalan la dificultad de evaluar las imágenes tomadas y seleccionar aquellas que fueron afectadas por la distorsión de líneas o por vibraciones de la cámara y sistemas CMOS [88]. En otro estudio se utiliza un scanner de láser en vez de una cámara, debido a las limitaciones de carga del *drone*. El sistema calcula la altura de un cultivo de maíz mediante la aplicación de filtros en la plataforma y transformaciones posteriores. En este proyecto el *sensado remoto* utiliza un hexacóptero comercial llamado AscTec Firefly, el cual es operado a bajas alturas (1-2 metros) para aumentar la resolución espacial y obtiene las fotografías mediante un vuelo automático basado en GPS pre-programado sobre una tarjeta de procesamiento portada en la plataforma usando el sistema operativo para robots (ROS). Como aporte los investigadores de este trabajo combinan las medidas del escáner laser con un sensor de altura, para controlar de manera precisa la altura de vuelo del robot [89]. Con la misma configuración de seis rotores que las anteriores aplicaciones, se encuentran sistemas de *sensado remoto* para análisis en el espectro de color (RGB). Un trabajo utiliza un robot llamado FlightCopter que vuela sobre

cultivos de avena y arvejas ubicados en una granja al norte de Hesse en Alemania, a una altura de 30 metros, con tiempo de vuelo de aproximadamente 10 minutos y controlado remotamente. Este es utilizado para calcular el índice normalizado de diferencia verde-rojo y de esta manera relaciona la biomasa sobre el suelo y el índice del área de la hoja en toda la granja [90]. Otro trabajo representativo de los sistemas de *sensado remoto* basado en imágenes digitales es publicado por Gómez - Candón et al., en este estudio los autores tratan la precisión de las imágenes capturadas desde un cuadricóptero MD4-1000 el cual sigue una trayectoria programada con diferentes alturas, giros y puntos de paso. Los productos del sistema, son orto mosaicos de cultivos de trigo ubicados en Sevilla, España, en los cuales se identifica la maleza. Además del aporte a la agricultura de precisión los investigadores analizan la importancia de la altura fija de vuelo (30-100 metros) en el momento de obtener las imágenes que permita diferenciar la maleza del cultivo y para obtener la cantidad de imágenes adecuada para cubrir toda el área en un tiempo mínimo [91]. Con la misma plataforma aérea, Mesas y su equipo han evaluado la precisión del *sensado remoto* basado en el solapamiento de las imágenes capturadas de una cámara multiespectral TetraCam mini-MCA portada en la plataforma [92]. Para completar con el estudio del *sensado remoto* se encuentra un estudio relacionado desarrollado en Colombia para cultivos agrícolas, utilizando un cuadricóptero llamado Pelican y una cámara multiespectral Tetracam ADC Lite, el objetivo de este trabajo es establecer un método para realizar de manera adecuada el mosaico de las imágenes en los diferentes espectros [93].

Es de esperar que las nuevas tendencias de coordinación y cooperación de drones hagan parte de las estrategias para AP. Como ejemplo Joao Valente, ha desplegado tres cuadricópteros en una granja de España. Las plataformas utilizadas para este estudio son el AR 100 y Ast Tec Hummingbird, su altura, ruta y tiempo de vuelo son el resultado de un planificador de ruta Off line diseñado para gestionar la potencia, velocidad y revisitas a los puntos de paso programados de cada cuadricóptero [77]. Otro ejemplo de trabajo en equipo de cuadricópteros es el trabajo desarrollado por [94] en el que crean un método para determinar el número de vehículos que optimice el problema de cobertura, variando solo la altura de vuelo y considerando el problema de encontrar la ruta óptima, mediante grafos.

2.2.5 Brechas de Investigación

El análisis de los trabajos expuestos en la sección anterior ha permitido identificar las brechas o limitaciones en un campo de acción como la agricultura de precisión y el *sensado remoto* con cuadricópteros como se muestra en la tabla 2, este análisis permite definir los aportes en los cuales el presente trabajo de maestría esta soportado.

Enfoque de Trabajos relacionados	Brechas
Posicionamiento de <i>drones</i> en ambientes externos	<ul style="list-style-type: none"> • El alcance en desarrollo de los trabajos revisados solo llega a pruebas de laboratorio o simulaciones • El control de altitud del dron es realizado frecuentemente con sensores incluidos en el sistema de navegación inercial • Los análisis del comportamiento del dron en los hallazgos arroja errores significativos en la estabilidad de vuelo sobre un punto definido.
Planificadores de Ruta de Cobertura	<ul style="list-style-type: none"> • La mayoría de los algoritmos desarrollados en la literatura revisada no consideran la topografía del cultivo. • Los trabajos encontrados no asumen tiempo de vuelo de la plataforma como parte de la planificación. • Ningún trabajo revisado considera una relación entre la altura de vuelo adecuada para sensado remoto en cultivos.
Sistemas de <i>Sensado remoto con Drones</i>	<ul style="list-style-type: none"> • Las técnicas de sensado remoto con drones revisadas capturan imágenes secuencialmente sin prever precisión en ellas. • Todos los trabajos encontrados consideran el mecanismo de captura de la imagen remotamente desde una estación base.

Tabla 2. Brechas para cada enfoque de investigación

2.3 Plataforma Aérea para Sensado Remoto

El sistema de *sensado remoto* propuesto para el proyecto está conformado por un conjunto de componentes hardware y software. Los componentes principales son: el robot aéreo multirroto, que para el caso es el cuadricóptero AR Drone versión 2.0 de Parrot y el sistema de procesamiento para ejecutar la misión y la estrategia de posicionamiento, llamado RaspberryPi 2 modelo B.

2.3.1 Plataforma Aérea: AR Drone

Para el desarrollo del proyecto se seleccionó AR Drone 2.0 como plataforma aérea para el sistema de *sensado remoto* debido a su accesibilidad, tecnología, sensores y valor agregado en cuanto a estabilidad, dado que usa técnicas de visión para la estimación de la velocidad. AR Drone hace parte de los vehículos aéreos fabricados por la empresa Parrot SA que desde hace más de 5 años ha tenido gran auge en todo el mundo por su enfoque en realidad aumentada y sensores integrados. AR Drone es un robot aéreo de desarrollo con un control basado en la variación de la velocidad de cuatro rotores adheridos a una equis simétrica. Puede tener un vuelo continuo de 15 minutos y una capacidad de carga máxima de 100 gramos. Además, puede ser pilotado mediante una aplicación móvil o software de escritorio, utilizando una comunicación WiFi.

El equipo tiene la capacidad de realizar maniobras agresivas, con un sistema robusto a las numerosas perturbaciones del ambiente y de la práctica real. Cada uno de los cuatro motores tiene una tarjeta de control llamada BLCB, que se basa en un micro controlador ATMEGA8L que interrumpe la operación del motor en el caso de que la hélice encuentre un obstáculo. Los motores están adheridos a una equis reforzada con fibra de carbono y plástico. La estructura soporta una batería tipo LiPo (en inglés Lithium Polymer) de tres celdas que proporciona con 11.1 Voltios y 1500 mAh para versión Power. La capacidad de las baterías es de 10 a 15 minutos de vuelo.

La base de funcionamiento del AR Drone se soporta en dos tarjetas: la tarjeta principal que embebe un procesador de 32 bits ARM Cortex A8, un circuito integrado WiFi, una cámara orientada verticalmente y una cámara frontal. El sistema operativo es de tiempo real basado en Linux, mediante el cual se adquiere el flujo de datos y el video de las dos cámaras integradas. La cámara frontal tiene una resolución tipo HD de 1280 x 720 con frecuencia de 30 marcos por segundo (FPS) y la cámara vertical tiene una resolución tipo QVGA con una frecuencia de 60 marcos por segundo. La tarjeta de navegación se basa en un Pic de 16 bits, el cual sirve de interfaz para los acelerómetros de tres ejes (precisión +/-50 mg), giróscopos de 3 ejes (precisión 2.000°/s), Magnetómetro de 3 ejes (precisión 6°), Sensor de presión (precisión +/- 10 Pa - 80 cm a nivel del mar) y sensor de ultrasonido para altitud respecto al suelo (figura 3).



Figura 3. Componentes de AR Drone [95]

La arquitectura de control del AR Drone se basa en varios modos de operación como se observa en la tabla 2, los cuales son gestionados por una máquina de estados finita la cual recibe las órdenes del piloto (sea manual o automático). El control se realiza en dos lazos anidados, uno para el control de posición y orientación, y el otro para el control de la velocidad angular (figura 4). El primer lazo calcula la velocidad angular del punto fijo, utilizando control proporcional integral y el segundo lazo controla los motores con controladores proporcionales simples. En la arquitectura de control definida en la figura 4 para AR Drone el lazo que calcula la velocidad angular comprende los módulos naranjas, los módulos color azul son módulos de procesamiento para obtener dichas velocidades, los módulos grises son los sensores que contiene los datos a ser procesados y finalmente el lazo de control de motores es representado por el modulo color morado.

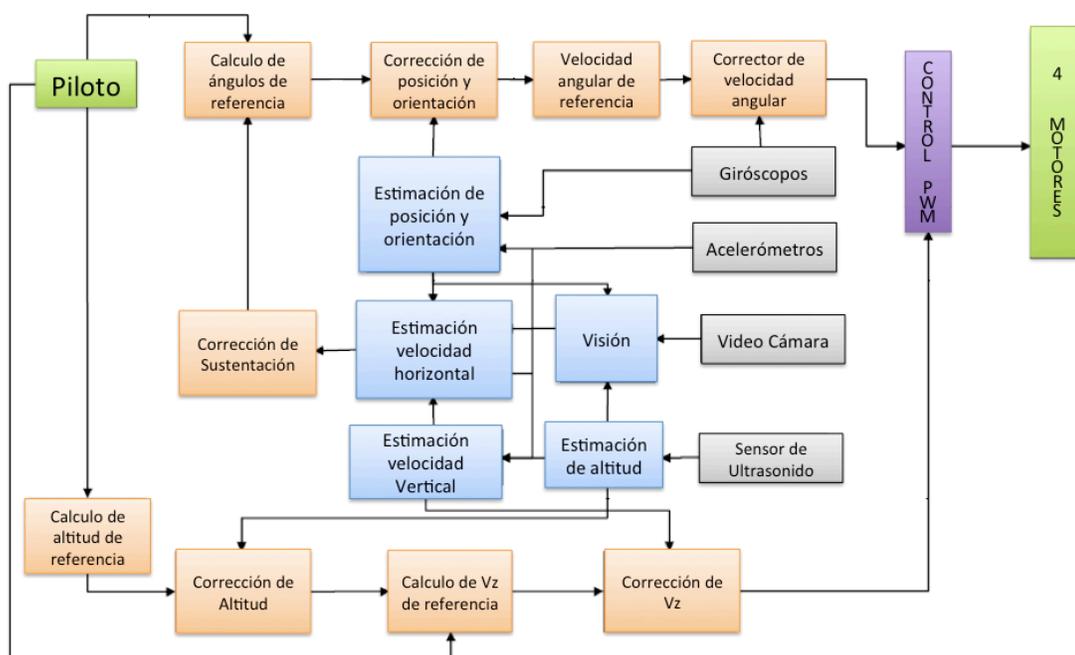


Figura 4.Arquitectura de Control AR Drone [96]

2.3.2 Procesamiento en vuelo: Raspberry Pi 2 modelo B

Raspberry Pi 2 es una tarjeta de computo reducida de bajo coste desarrollado para motivar las ciencias de la computación (figura 5a), es la segunda generación de las tarjetas Raspberry las cuales son productos registrados de uso libre. El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM Cortex A7 Quadcore a 900 MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM. Utiliza una tarjeta SD para el almacenamiento permanente

y ofrece cuatro puertos USB 2.0. La conectividad a red está dada para módulo WiFi de baja potencia y la fuente de alimentación es de 5 voltios vía Micro USB.

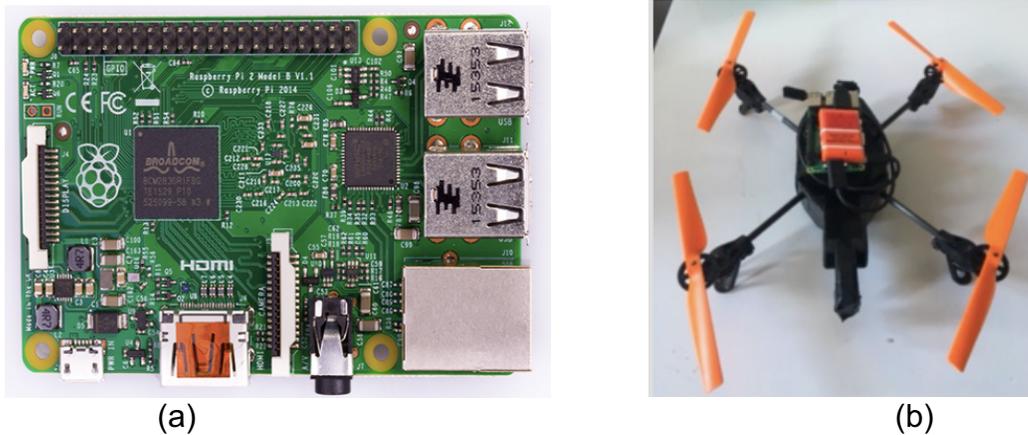


Figura 5. Módulo de procesamiento en vuelo a) Raspberry Pi 2 B y b) Montaje en la plataforma aérea

En el presente proyecto se utiliza la tarjeta Raspberry como recurso hardware intermedio entre el autopiloto del AR Drone y el software de procesamiento desarrollado para el *sensado remoto*. Este dispositivo se soporta en el sistema operativo Ubuntu 14.04 para correr el sistema operativo portable que ha sido seleccionado para el proyecto, denominado ROS por sus siglas en ingles Robot Operating System. Este módulo de procesamiento hace parte de la carga útil de la plataforma aérea para el *sensado remoto* (figura 5b) y su tarea es recibir la ruta calculada por el planificador en tierra, ejecutar el controlador del drone, ejecutar los algoritmos para posicionamiento y seguimiento de trayectoria en vuelo.

2.3.3 ROS

El sistema operativo para robots ROS el cual corresponde al marco de trabajo del proyecto propuesto, es un conjunto de herramientas, librerías y convenciones, que simplifican la tarea de controlar y simular un robot. El sistema está diseñado con una arquitectura basada en grafos, donde los nodos representan los procesos específicos de control de los robots, y la comunicación entre nodos se realiza mediante mensajes, los cuales se publican o subscriben a tópicos de comunicación, como se ve en la figura 5 [97]. El sistema operativo ofrece la posibilidad de crear nodos en lenguajes de programación como Python y C++, maneja un formato de mensajes particular para la geometría, la posición, navegación, mapeo, control y llamado de funcionalidades remotas.

ROS, tiene además herramientas para control y visualización de resultados como RVIZ, el cual visualiza muchos de los tipos de mensajes comunes proporcionados en ROS, como escaneos láser, nubes de puntos tridimensionales e imágenes de cámara. También muestra todos los datos de sensores en un marco de coordenadas

común de su elección, junto con una representación tridimensional de su robot. La visualización de todos sus datos en la misma aplicación permite identificar problemas tales como desalineamientos de sensor o inexactitudes de modelo de robot. Otra herramienta importante en ROS es Rqt, el cual es un marco de trabajo basado en Qt para el desarrollo de interfaces gráficas. En la práctica esta herramienta es utilizada para graficar el comportamiento de sensores y obtener el grafo que corre durante la ejecución de un proyecto como la figura 6.

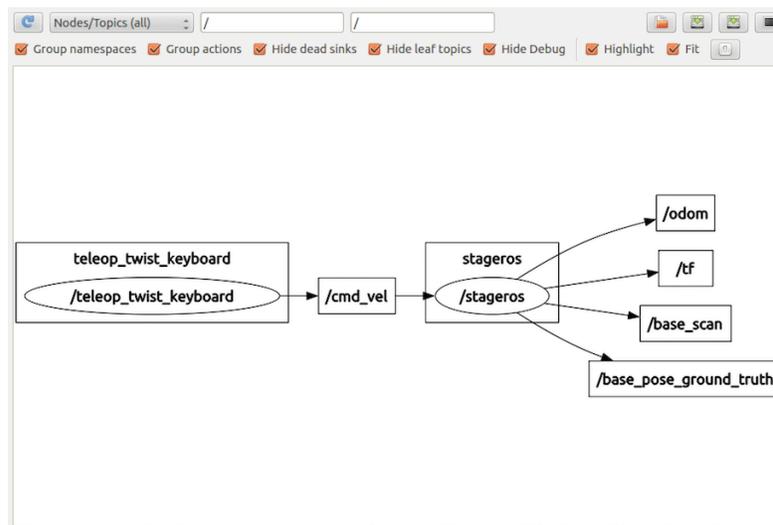


Figura 6. Ejemplo de implementación de nodos en ROS

Un grafo de nodos ROS se visualiza con los paquetes representados en rectángulos con su respectivo orden, nodos representados por elipses dentro de paquetes, tópicos representados por rectángulos. La relación entre tópicos y nodos se visualiza mediante flechas que se orientan indicando si un nodo se suscribe o publica en un tópico.

2.4 Resumen

Este capítulo presentó las bases de conocimiento para el desarrollo de la propuesta de investigación, los temas tratados incluyen: el concepto y tecnologías para agricultura de precisión; el alcance y componentes de los sistemas de *sensado remoto*; y la definición, clasificación y operación de los vehículos aéreos no tripulados. Todo lo anterior con el objeto de comprender el origen y la terminología del tema que fue declarado.

La comprensión de la temática da pie a la exploración de trabajos relacionados, de los cuales se resume que en la actualidad el principal reto en posicionamiento es para entornos abiertos utilizando fusión de datos que incluyen posicionamiento global o posicionamiento por visión sin embargo la última opción es cuestionada debido a que el procesamiento de los datos suele ser costoso y complejo. En cuanto a los algoritmos de planificación se tiene dos divisiones marcadas, una para ambientes

desconocidos los cuales consideran frecuentemente múltiples robots y no garantizan cobertura completa. La otra división abarca las estrategias de cobertura con conocimiento del escenario, que utilizan regularmente un solo robot. Una de sus principales limitaciones en ambientes reales es la falta de acciones reactivas en casos de perturbaciones externas debidas a cambios en el entorno. El capítulo termina describiendo los componentes hardware y software seleccionados para implementar el prototipo de *sensado remoto* para agricultura de precisión. Los componentes principales son: el AR Drone, quien porta el sensor de visión desde el cual se adquieren los datos, y una unidad de procesamiento en vuelo, la cual se encarga de comunicar el AR Drone con el software para optimización de posicionamiento y de ejecutar la ruta de cobertura global planificada.

Posicionamiento del Drone en Ambientes Abiertos

Para el proyecto es determinante contar con un sistema de posicionamiento preciso y robusto, que le permita superar las dinámicas del entorno propio de un cultivo agrícola. Por lo tanto, si no se cuenta con la precisión en el posicionamiento puede limitarse el seguimiento de una trayectoria específica; lo cual se percibe claramente en la fiabilidad de los datos recolectados durante la misión de *sensado remoto*. De acuerdo a esto, el problema de posicionamiento del cuadricóptero es prioridad en planificación un vuelo sobre cualquier área.

Para realizar el diseño de una estrategia de posicionamiento de un cuadricóptero es necesario comprender el fundamento físico-matemático relacionado con los movimientos traslacionales y rotacionales del robot en un sistema de navegación, ya sea local o global. De acuerdo al comportamiento esperado de un cuadricóptero, se propone explorar la precisión de los sensores para navegación aérea y establecer una estimación del posicionamiento con un algoritmo de fusión sensorial que se adapte a los requerimientos de un entorno como los cultivos agrícolas.

3.1 Navegación de un Robot Aéreo

El origen de la navegación o posicionamiento de un robot aéreo es el requerimiento de llegar a un destino, para lo cual se utilizan técnicas con las cuales se comandan maniobras para lograr dicho objetivo. Dos de las técnicas mayormente utilizadas es la navegación inercial y la navegación por satélites. El posicionamiento basado en satélites utiliza un receptor de señales denominado GPS, el cual es un sistema que proporciona continuamente a los usuarios información de posición en todo el mundo y en todo tiempo. El posicionamiento inercial está basado en la unidad de movimiento inercial IMU, la cual está montada en un objeto en movimiento y utiliza mediciones de sensores inerciales, es decir, acelerómetros y giroscopios y dinámicas de navegación para calcular la posición, la velocidad y la orientación de un objeto en movimiento. Una técnica alternativa para posicionamiento de un robot aéreo es usando triangulación y análisis de videos en tiempo real, la cual está basada regularmente en cámaras estereoscópicas.

Cualquiera de las técnicas utilizadas para navegación aérea está fundamentado en un sistema de referencia, con el cual se determina la posición de partida, la orientación y el desplazamiento, y un modelo físico-matemático mediante el cual se estima la posición y orientación de acuerdo a cualquiera de los sistemas de referencia.

3.1.1 Sistemas de Referencia para un Robot Aéreo

Los sistemas de referencia en la navegación de un robot aéreo, son sistemas de coordenadas utilizados para localizar el vehículo en un espacio, que puede ser local o global. La comprensión de los sistemas de referencia utilizados en la navegación de un *drone* es el primer paso para diseñar una estrategia de posicionamiento en ROS con cualquiera de las técnicas de navegación. Los sistemas de referencia que se nombran a continuación son modelados en herramientas de visualización y simulación como RVIZ y Gazebo respectivamente.

- **Sistema Inercial**

El sistema inercial se basa en la idea aproximada de un planeta como una esfera y mantiene sus coordenadas con el origen en el centro de masa de tal esfera maciza. El eje Z se alinea con el spin de la tierra, y los ejes X, Y se encuentran en el plano ecuatorial, de tal manera que un sistema coordenado es formado al lado derecho de la tierra. Este sistema no rota con la tierra y es denotado por el literal $\{i\}$ en la figura 7. El sistema $\{i\}$ es considerado como la base de los demás sistemas de referencia para declarar el posicionamiento de un vehículo aéreo.

- **Sistema ECEF (*earth-centered, earth-fixed*)**

Es similar al sistema inercial $\{i\}$, con la diferencia que este sistema rota con la tierra y por tanto las coordenadas de un punto fijo sobre la tierra no cambian, es denotado por el literal $\{e\}$ en la figura 7. Sus ejes están alineados con el polo internacional de referencia y el meridiano de referencia internacional que se fijan con respecto a la superficie de la tierra. De esta manera, el eje Z se extiende a través del norte verdadero, que no coincide con el eje de rotación instantáneo de la tierra. El eje X pasa a través de la intersección del meridiano principal (primer meridiano en Greenwich) y el ecuador, y el eje Y completa un sistema de coordenadas a la derecha de la tierra. Si el sistema de coordenadas utiliza dichos ejes se cataloga como un sistema rectangular ECEF, pero el marco ECEF también se puede describir en coordenadas geodésicas, las cuales son denominadas, longitud, latitud y altitud, y usadas para describir la localización de un punto relativo al sistema WGS-84, que es el modelo de coordenadas representado por un elipsoide.

En la aplicación implementada en ROS para optimizar posicionamiento, se considera que el receptor GPS utilizado se enmarca en el sistema WGS-84, y la conversión a coordenadas rectangulares ECEF es requerida para el cálculo de posición y velocidad de un vehículo aéreo. La conversión en ROS considera el sistema UTM (en inglés, Universal Transverse Mercator) donde las magnitudes se expresan en metros únicamente al nivel del mar, que es la base de la proyección del elipsoide de referencia.

- **Sistema Local**

Este sistema se etiqueta con el literal $\{n\}$ en la figura 7, y es el sistema local de navegación con los ejes X e Y tangenciales a la superficie de la tierra y el eje Z ortogonal a la superficie de la tierra, para completar un sistema de referencia a la derecha. Este sistema de referencia es utilizado para aplicaciones con robots aéreos navegando en espacios cerrados y delimitados, por lo que el origen del sistema se localiza en el punto inicial detectado en el arranque de una aeronave.

En la aplicación implementada en ROS para optimizar posicionamiento este sistema no rota y está alineado con el plano de la superficie donde se ubica el *drone*, desde el cual se estima la posición y velocidad local.

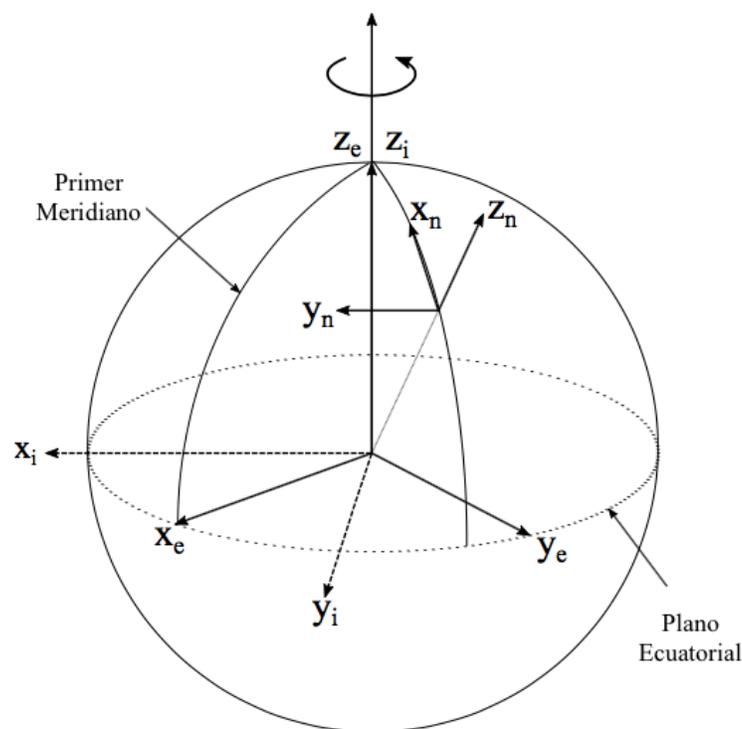


Figura 7. Sistemas de referencia para la navegación aérea

- **Sistema del Cuerpo**

Este sistema es utilizado en la navegación de un robot aéreo, ya que regularmente estas aeronaves portan sensores adheridos a la estructura. El centro de masa del vehículo es fijado como el origen del sistema, el eje Z es perpendicular al vehículo, el eje X es el frente y el eje Y es un eje lateral para completar un sistema a la derecha, como se puede apreciar en la figura 8 con el literal $\{b\}$. Para una implementación en ROS, pueden existir varios sistemas

de referencia relacionados con el centro de masa del drone y cada uno de los sensores adheridos a él.

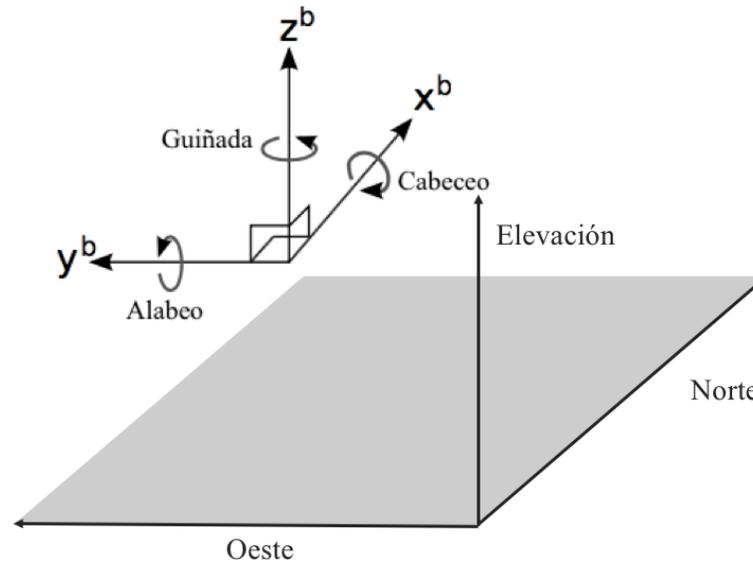


Figura 8. Sistemas de referencia para los sensores adheridos al cuerpo

3.1.2 Modelo Físico-Matemático de un Cuadricóptero

El diseño de cualquier estrategia de posicionamiento para un *drone*, está basada en la interpretación de la dinámica del vehículo mediante un precedente de ecuaciones que describen los fundamentos para modelar el movimiento en cualquier sistema de coordenadas mencionado anteriormente. De esta manera los movimientos relacionados al sistema $\{e\}$, están fundamentados en los movimientos del sistema local de navegación y estos a su vez se derivan de los movimientos del sistema del cuerpo $\{b\}$. Estas conversiones y transiciones en el posicionamiento de un cuadricóptero son modeladas matemáticamente considerando los ejes X, Y, Z (vector ξ) y ángulos de Euler (vector η) del sistema $\{n\}$ [98].

$$\xi = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \eta = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \quad (3.1.1)$$

La conversión completa entre el sistema $\{n\}$ y el sistema $\{b\}$, se obtiene considerando las rotaciones a través del ángulo de Euler φ (cabeceo) alrededor del eje X, del ángulo de Euler θ (alabeo) alrededor del eje Y y del ángulo de Euler ψ (guiñada) alrededor del eje Z. Las rotaciones entre los dos sistemas se interpretan como las matrices de rotación para cada ángulo de Euler, es decir que a cada ecuación de movimiento le corresponde una matriz de rotación como se describe en las siguientes ecuaciones:

Para cabeceo,

$$\begin{bmatrix} Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} Y_n \\ Z_n \end{bmatrix}$$

la matriz de rotación es

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \quad (3.1.2)$$

Para alabeo,

$$\begin{bmatrix} X_b \\ Z_b \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_n \\ Z_n \end{bmatrix}$$

la matriz de rotación es

$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (3.1.3)$$

Para guiñada,

$$\begin{bmatrix} X_b \\ Y_b \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$$

la matriz de rotación es

$$R_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1.4)$$

De acuerdo a las matrices de rotación se pueden calcular los cambios de posición y orientación del cuadricóptero en cada momento, utilizando el producto de las matrices de rotación $R = R_x \cdot R_y \cdot R_z$ (ecuación 3.1.5). La matriz de rotación R es ortogonal por lo tanto $R^{-1} = R^T$.

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \quad (3.1.5)$$

Para lograr la rotación entre los sistemas de navegación, se requieren los ángulos de movimiento cabeceo, alabeo y guiñada obtenidos desde la velocidad de rotación de los cuatro rotores, peso y gravedad. La velocidad de rotación, es producida por cada rotor a través del torque aplicado por motores de corriente continua sin escobillas que son accionados por controladores electrónicos de velocidad. La velocidad de cada rotor es w_i (para i rotores), la cual crea fuerza en la dirección del eje del rotor f_i como la ecuación 3.1.6. La velocidad angular y la aceleración del rotor también crean un torque τ_{M_i} alrededor del eje del rotor como la ecuación 3.1.7 (figura 9). En las que la constante de elevación es k , la constante de arrastre es b y el momento de inercia del rotor es I_M .

$$f_i = kw_i^2 \quad (3.1.6)$$

$$\tau_{M_i} = bw_i^2 + I_M\dot{w}_i \quad (3.1.7)$$

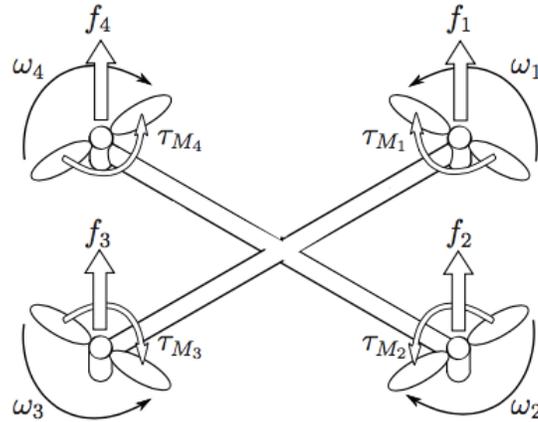


Figura 9. Dinámica de un Cuadricóptero

I_M es una matriz diagonal que corresponde a los momentos de inercia del cuerpo en los ejes X, Y, y Z, los cuales coinciden con una estructura simétrica, por lo tanto I_M se define como la ecuación 3.1.8.

$$I_M = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.1.8)$$

Las fuerzas combinadas de los rotores crean el empuje T (ecuación 3.1.9) en la dirección del eje z del cuerpo y el τ_b consiste en los torques dados por cada angulo de Euler τ_ϕ , τ_θ y τ_ψ , en la dirección de los correspondientes ángulos del sistema $\{b\}$ (ecuación 3.1.10). En las que l es la distancia entre el rotor y el centro de masa del cuadricóptero.

$$T = \sum_{i=1}^4 f_i = k \sum_{i=1}^4 \omega_i^2, \quad T_b = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (3.1.9)$$

$$\tau_b = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lk(-\omega_2^2 + \omega_4^2) \\ k(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^4 \tau_{M_i} \end{bmatrix} \quad (3.1.10)$$

Siguiendo el modelo del cuadricóptero descrito anteriormente, se asume que el cuadricóptero es un cuerpo rígido, por lo cual su dinámica se puede describir mediante las ecuaciones Newton-Euler. De esta manera en el sistema del cuerpo, la fuerza requerida para la aceleración del cuadricóptero $m\dot{V}_b$ y la fuerza centrífuga $W_b \times (m\dot{V}_b)$ son iguales a la gravedad $R^T G$ y al empuje total de los rotores T_b como se

describe en la ecuación 3.1.11, donde V_b y W_b corresponden a las velocidades lineales y angulares.

$$m\dot{V}_b + W_b \times (m\dot{V}_b) = R^T G + T_b \quad (3.1.11)$$

$$V_b = \begin{bmatrix} v_{x,b} \\ v_{y,b} \\ v_{z,b} \end{bmatrix}, \quad W_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

En el sistema local, la fuerza centrífuga es nula. Por lo tanto, solo la fuerza gravitacional y la magnitud y dirección del empuje contribuyen a la aceleración del cuadricóptero ξ (ecuación 3.1.12).

$$m\ddot{\xi} = G + RT_b$$

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} \cos\psi \sin\theta \cos\varphi + \sin\psi \sin\varphi \\ \sin\psi \sin\theta \cos\varphi - \cos\psi \sin\varphi \\ \cos\theta \cos\varphi \end{bmatrix} \quad (3.1.12)$$

En el sistema $\{b\}$, la aceleración angular $I_M \dot{W}_b$, las fuerzas centrípetas $W_b \times (I_M W_b)$, y las fuerzas giroscópicas Γ son iguales al torque externo τ como la ecuación 3.1.13. En la que $w_\Gamma = w_1 - w_2 + w_3 - w_4$.

$$I_M \dot{W}_b + W_b \times (I_M W_b) + \Gamma = \tau$$

$$\dot{W}_b = I_M^{-1} \left(- \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_\Gamma + \tau \right),$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})qr/I_{xx} \\ (I_{zz} - I_{xx})pr/I_{yy} \\ (I_{xx} - I_{yy})pq/I_{zz} \end{bmatrix} - I_r \begin{bmatrix} q/I_{xx} \\ -p/I_{yy} \\ 0 \end{bmatrix} w_\Gamma + \begin{bmatrix} \tau_\varphi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix} \quad (3.1.13)$$

Las aceleraciones angulares $\ddot{\eta}$ en el sistema $\{n\}$ son obtenidas desde las aceleraciones en el sistema $\{b\}$, con la matriz de transformación E_η^{-1} (rotación de las velocidades angulares del cuerpo y para calcular las velocidades angulares del sistema local) y su derivada en el tiempo como se describe en la ecuación 3.1.14.

$$E_\eta^{-1} = \begin{bmatrix} 1 & \sin\varphi \tan\theta & \cos\varphi \tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix}$$

$$\ddot{\eta} = \frac{d}{dt}(E_{\eta}^{-1}W_b) = \frac{d}{dt}(E_{\eta}^{-1})W_b + E_{\eta}^{-1}\dot{W}_b$$

$$\ddot{\eta} = \begin{bmatrix} 0 & \dot{\varphi}\cos\varphi\tan\theta + \frac{\dot{\theta}\sin\varphi}{\cos^2\theta} & -\dot{\varphi}\sin\varphi\cos\theta + \frac{\dot{\theta}\cos\varphi}{\cos^2\theta} \\ 0 & -\dot{\varphi}\sin\varphi & -\dot{\varphi}\cos\varphi \\ 0 & \frac{\dot{\varphi}\cos\varphi}{\cos\theta} + \frac{\dot{\varphi}\sin\varphi\tan\theta}{\cos\theta} & -\frac{\dot{\varphi}\sin\varphi}{\cos\theta} + \frac{\dot{\theta}\cos\varphi\tan\theta}{\cos\theta} \end{bmatrix} W_b + E_{\eta}^{-1}\dot{W}_b \quad (3.1.14)$$

3.1.3 Odometría en ROS

La estimación de la posición de un robot ya sea terrestre, submarino o aéreo parte de un sistema de navegación conformado por una unidad de movimiento inercial IMU, con la cual se calcula la odometría del robot. La odometría es una técnica para estimar (y no determinar) la posición con respecto a su localización inicial (es decir con respecto a un sistema de referencia local), esta es ampliamente utilizada por su bajo consumo de recursos computacionales, elevada frecuencia de cálculo y buena precisión a distancias cortas; sin embargo, existe una acumulación inevitable de errores, a medida que la distancia va aumentando. Esta dificultad se soporta con otros métodos para calcular la posición absoluta del robot, como por ejemplo el uso de sistemas de referencia global con o sin etiquetas visibles al robot.

La odometría para un cuadricóptero en ROS está basada en la predicción de una nueva posición con base en el modelo físico-matemático de un cuadricóptero, en conjunto con la velocidad y dirección de los motores que ejercen el empuje en el sistema local de navegación y los sensores adheridos al cuerpo. En la figura 10 se indican cada una de las variables que considera la odometría básica ($x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \theta, \psi$), donde x, y, z representan la posición en tres dimensiones, φ, θ, ψ representan los ángulos de viraje del cuadricóptero, $\dot{x}, \dot{y}, \dot{z}$ representan las velocidades lineales en tres dimensiones, $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ las velocidades angulares respectivas.

Odometria $\{x, y, z, \varphi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\varphi}, \dot{\theta}, \dot{\psi}\}$

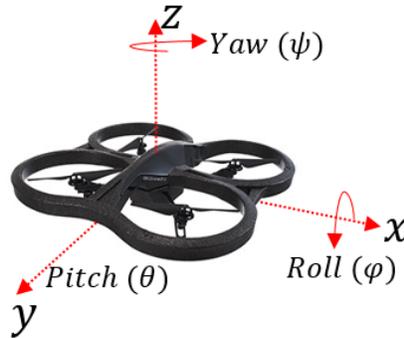


Figura 10. Odometría básica de AR Drone

El t3pico de odometr3a es el resultado de aplicar cualquier estrategia de posicionamiento en el marco de ROS. El tipo de mensaje del t3pico es de la librer3a de navegaci3n, llamado *nav_msgs/odometry*, que para el caso del AR Drone es conocido como */ardrone/odometry*.

3.2 Selecci3n de la Estrategia de Posicionamiento.

De acuerdo a lo anterior, se realiza una exploraci3n de los sistemas de navegaci3n por los cuales se puede obtener odometr3a del AR Drone. El primero est3 basado en el sistema inercial y la visi3n vertical propia del *drone*, el segundo est3 basado en la navegaci3n apoyada en la visi3n frontal del AR Drone y el 3ltimo est3 basado en un referenciaci3n global mediante el receptor GPS Flight Recorder.

3.2.1 Navegaci3n del Ar Drone

Los principios de la navegaci3n para el cuadric3ptero AR Drone son [96]:

- Los aceler3metros y gir3scopos est3n integrados para estimar los 3ngulos de orientaci3n y velocidades.
- El segundo principio relaciona el vector de velocidad del centro de gravedad $V = [u \ v \ w]^T$, los 3ngulos de Euler roll, pitch, yaw del cuerpo $Q = [\phi \ \theta \ \psi]^T$ y la velocidad angular del cuerpo $\Omega = [p \ q \ r]^T$ como se describe en las siguientes ecuaciones:

$$\dot{V} = -\Omega x V + F \quad (3.2.1)$$

$$\dot{Q} = G(\Omega, Q) \quad (3.2.2)$$

$$G(\Omega, Q) = \begin{bmatrix} p + (q \sin \phi + r \cos \phi) \tan \theta \\ q \cos \phi - r \sin \phi \\ (q \sin \phi + r \cos \phi) \sec \theta \end{bmatrix} \quad (3.2.3)$$

- Los aceler3metros no miden la aceleraci3n del cuerpo sino la aceleraci3n menos la gravedad.
- Los gir3scopos son fragmentados y ruidosos.

Para la regulaci3n y correcci3n de los principios de navegaci3n, los sensores inerciales compensan las microrrotaciones en las im3genes de la c3mara. Esto ayuda a calcular el flujo 3ptico del algoritmo de visi3n, considerando dos marcos de imagen sucesivos a 60Hz desde lo cual se deduce una velocidad lineal en 3D, utilizando el desplazamiento de p3xeles. El desplazamiento de los objetos se relaciona con la velocidad lineal en el plano horizontal, a su vez la velocidad vertical lineal y angular se compensan con la altitud que surge de un algoritmo de fusi3n entre el son3metro y aceler3metros.

El resultado de los algoritmos, le da al AR Drone una mejor precisión en la estimación de la velocidad haciéndolo robusto en todas las condiciones.

3.2.2 Navegación por Visión

Este método se enfoca en la implementación de navegación utilizando la cámara monocular frontal con el propósito de realizar vuelos autónomos sin un conocimiento previo del ambiente. De manera general se utilizan técnicas de fusión con el sistema inercial y la cámara frontal. Las estrategias para posicionamiento por visión en orden son [99]:

- Mapeo y localización simultanea monocular: está basado en una técnica de rastreo y mapeo paralelo conocido por sus siglas en inglés como PTAM, realacionando el mapa visual con el plano x-y que corresponde al plano horizontal determinado por los acelerómetros y por la escala desde un punto de referencia.
- La estimación de posición se realiza mediante un Filtro de Kalman Extendido (FKE): Desde este método se calibra el movimiento completo del AR Drone y la reacción a los comandos de control. Un algoritmo FKE también se utiliza para compensar los retardos en el sistema debidos a la comunicación inalámbrica y al consto computacional del rastreo visual.
- Control proporcional-integral-derivativo: Este componente se basa en la posición y velocidad estimada del FKE, para aplicar un control proporcional integral derivativo, PID, que lleve al cuadricóptero a una localización deseada en un sistema de coordenadas global. Este control se aplica para los cuatro grados de libertad involucrados, en orden x, y, z y orientación.

El sistema planteado procesa los algoritmos en una plataforma diferente a la de AR Drone y es capaz de navegar en ambientes no conocidos, sin necesidad de GPS, utilizando la cámara tipo HD del cuadricóptero y un estimador para la escala del mapa visual por encima del 1,7% del valor verdadero, que lleva un promedio de error en posicionamiento de 4,2 cm en interiores y 18 cm en exteriores.

3.2.3 Navegación por GPS

Un conjunto de satélites en órbita alrededor de la tierra asegura que se pueda localizar un receptor GPS en cualquier parte del globo, siempre que el receptor esté dentro de la línea de visión de al menos siete de los satélites. Los satélites difunden mensajes que contienen el tiempo preciso para el envío de un mensaje, la identidad del satélite y su posición actual. Un receptor GPS determina la distancia a un único satélite comparando el tiempo de llegada y envío de un mensaje y sabiendo que las señales viajan con la velocidad de la luz. Conociendo la distancia a por lo menos siete satélites y sus posiciones relativas de tierra, el receptor determina su posición en el sistema de coordenadas ECEF usando triangulación.

Este tipo de navegación conocida como navegación por puntos de paso o *waypoints* en inglés, es quizá la más útil en las aplicaciones de los *drones* comerciales. Esto es posible realizarlo partiendo de que dispositivos de navegación satelital pueden marcar una ubicación en cualquier lugar de la tierra, es decir un punto de referencia en el espacio físico. Dichos puntos remotos o cercanos se almacenan electrónicamente en el *drone* y se pueden transferir a un software de cartografía digital. Para el caso del AR Drone, QgroundControl [100] y Paparazzi [101] son software de escritorio que permiten realizar el seguimiento del *drone* durante su navegación por GPS. Este método de navegación trata sobre las transformaciones que se realizan entre el sistema de coordenadas local y coordenadas geográficas. Dicha transformación se realiza después que se ha predefinido los puntos por los cuales el cuadricóptero debe pasar. Regularmente se puede planificar la ruta con un punto de partida y un punto de llegada, la altura de vuelo, la velocidad y el tiempo de sustentación en cada punto. La navegación usando GPS se considera un sistema fiable para la localización al aire libre. Sin embargo, los datos GPS son generalmente demasiado ruidosos y están provistos de una tasa de muestreo demasiado baja para proporcionar información sobre las derivadas de posición.

3.2.4 Comparación de sistemas de navegación

De acuerdo a los métodos de navegación descritos anteriormente se realiza una caracterización de cada estratégica según requerimientos puntuales en la estimación de la posición y orientación, rango de acción, y complejidad computacional. A estas características se les asigna un peso cualitativo (bajo, medio y alto) de acuerdo al alcance registrado por la literatura estudiada (tabla 3).

Características	Precisión en odometría	Rango de acción	Recursos computacionales
Navegación del Ar Drone	Media	Bajo	Media
Navegación por Visión	Alta	Bajo	Alta
Navegación por GPS	Media	Alto	Baja

Tabla 3. Comparación de sistemas de navegación

Con base en la tabla 3 es posible deducir las falencias en el posicionamiento propio del AR Drone en espacios abiertos. Si bien es posible navegar con el diseño y la implementación básica del Ar Drone de una forma segura, el posicionamiento presenta un rango de acción limitado ya que depende de la comunicación continua en tierra, además de que el diseño de fábrica no califica para ambientes agrestes como el que se presenta en los cultivos colombianos. Por el lado de las técnicas de visión se rescata la capacidad de adaptación del *drone* al entorno y la precisión en posición y orientación, sin embargo, la insuficiencia no solo está en la dependencia

con estación en tierra para navegar, sino en los recursos computacionales necesario para el procesamiento de video para lograr la correlación del posicionamiento entre la IMU y la traslación de la cámara. Finalmente, la navegación por GPS no evidencia precisión en posicionamiento, sin embargo, es independiente de la estación en tierra, siempre y cuando se tenga acceso a las señales satelitales.

Considerando que los requerimientos para precisar posicionamiento en ambientes abiertos corresponden a tener una odometría optima, un amplio rango de acción para navegar grandes áreas y uso de pocos recursos computacionales para no restringir el tiempo de vuelo. La anterior cualificación de los métodos de posicionamiento y orientación, da pie para proponer la implementación de un filtro de Kalman, mediante el cual se integren las ventajas que tiene el posicionamiento inercial y las del posicionamiento georreferenciado con GPS. De esta manera es posible construir una solución que permita navegar en un sistema de coordenadas geográfico de manera precisa; con el objetivo de llegar a lugares más distantes, sin comprometer el tiempo de vuelo del AR Drone.

3.2.5 Fusión Sensorial: Filtro de Kalman (FK)

El filtro de Kalman es uno de los algoritmos de fusión de datos más relevantes de la actualidad a pesar que en la práctica lleva más de 50 años. Este algoritmo fue nombrado por Rudolf E. Kalman y su éxito se debe a que tiene requerimientos computacionales pequeños, propiedades recursivas y la estimación del estado asume errores estadísticos gaussianos. El filtro se utiliza desde los receptores de posicionamiento global, en estaciones de radio o en la salida de los paneles táctiles de los dispositivos móviles.

Desde un punto de vista matemático el FK es un modelo bayesiano similar al modelo de Markov, pero donde el espacio de estados de las variables es continuo y las variables y estimaciones tienen una distribución gaussiana. Regularmente se usa para sistemas dinámicos con información incierta, a partir de la cual se puede predecir lo que este hará en instante siguiente, aunque sin dejar de precisar lo que sucede en el instante evaluado. De aquí que el FK se divide en dos etapas una de *Predicción* (o actualización de tiempo) y otra de *Corrección* (o actualización de medida). De acuerdo esto, el FK está diseñado para sistemas dinámicos lineales, pero en la práctica estos sistemas no son comunes, por lo que el filtro de Kalman se extiende para sistemas no lineales. Las extensiones son conocidas como Filtro de Kalman Extendido (FKE) y Filtro de Kalman "Unscented" (FKU).

El **Filtro de Kalman Extendido FKE** [102] aborda el problema como un modelo con representación de variables de estado en tiempo discreto (ecuaciones 3.31 y 3.3.2):

$$x_k = f_{k-1}(x_{k-1} + u_{k-1}) + w_{k-1}, \quad k \geq 1 \quad (3.3.1)$$

$$y_k = h_k(x_k) + v_k, \quad k \geq 1 \quad (3.3.2)$$

Donde $\{x_k; k \geq 0\}$ es el espacio que integra n estados de un sistema dinámico no lineal y sea $\{y_k; k \geq 0\}$ el espacio de medidas desde los sensores para la estimación, u_k es la entrada conocida del sistema. $f_k: \mathbb{R}^{n+p} \rightarrow \mathbb{R}^n$ y $h_k: \mathbb{R}^n \rightarrow \mathbb{R}^m$ son funciones no lineales, continuas y diferenciables. Para este filtro se siguen las siguientes premisas:

- El estado inicial x_0 , es un espacio aleatorio n -dimensional, con media cero y matriz de covarianzas conocida P_0 .
- $\{u_k; k \geq 0\}$ es un proceso de control p -dimensional, no aleatorio.
- $\{w_k; k \geq 0\}$ es un ruido del proceso con covarianzas conocidas Q_k
- $\{v_k; k \geq 1\}$ es un ruido de la medida con covarianzas conocidas R_k .
- El estado inicial x_0 , y los ruidos $\{w_k; k \geq 1\}$ y $\{v_k; k \geq 1\}$ son mutuamente independientes.

El algoritmo viene dado por las siguientes expresiones donde las funciones son interpretadas en la práctica mediante matrices [103]:

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k [y_k - H_k \hat{x}_{k/k-1}], \quad k \geq 1 \quad (3.3.3)$$

$$\hat{x}_{k/k-1} = F_k \hat{x}_{k-1/k-1} + u_{k-1}, \quad k \geq 1 \quad (3.3.4)$$

$$K_k = P_{k/k-1} H_k^T [H_k P_{k/k-1} H_k^T + R_k]^{-1}, \quad k \geq 1 \quad (3.3.5)$$

$$P_{k/k} = [I_n - K_k H_k] P_{k/k-1}, \quad k \geq 1 \quad (3.3.6)$$

$$P_{k/k-1} = F_{k-1} P_{k-1/k-1} F_{k-1}^T + Q_{k-1}, \quad k \geq 1 \quad (3.3.7)$$

El objetivo del algoritmo es estimar en cada instante $k \geq 1$, el estado x_k a partir de las medidas $y_{1:k} = [y_1, \dots, y_k]$. El estimador $\hat{x}_{k/k}$ de x_k es óptimo, si el error cuadrático medio es mínimo. Conociendo el error de covarianza $P(x_k | y_{1:k})$ es posible determinar el error cuadrático medio mínimo y en conjunto con el error de covarianza inicial $P(x_0 | y_{1:0}) \triangleq P(x_0)$ se determina el error de covarianzas en cualquier instante de tiempo. La etapa de *predicción* del FKE la componen las expresiones 3.3.4 y 3.3.7, con ellas se calcula las estimaciones un instante antes $\hat{x}_{k/k-1}$ y su respectivo error de covarianza $P_{k/k-1}$, en la etapa de predicción la matriz F_k , es una matriz resultante de la linealización del sistema mediante el cálculo del Jacobiano. La etapa de *corrección* comprende las expresiones 3.3.3 y 3.3.6, con las cuales se estima el estado un instante después $\hat{x}_{k/k}$ y $P_{k/k}$. En la etapa de corrección se calcula primero la ganancia de Kalman K_k con la expresión 3.3.5 para luego actualizar la medida.

Aunque el FKE es el algoritmo de fusión sensorial más, este filtro presenta fallas si los sistemas no lineales son rígidos, en este caso la alternativa es el **Filtro de Kalman "Unscented"**. El problema con los sistemas no lineales es que la función de densidad

de probabilidad o error de covarianza, es una función no lineal general. El FKE establece que una transformación lineal de medias y covarianzas, es aproximadamente igual a la verdadera transformación no lineal. La transformación “*Unscented*” calcula los primeros momentos de la distribución de densidad de probabilidad de una variable aleatoria, que resulta de aplicar una transformación no lineal a una variable aleatoria de media y varianza conocidas [104].

Una transformación “*Unscented*” se basa en fundamentos intuitivos: Primero, es fácil realizar una transformación no lineal en un punto singular, y segundo, no es muy difícil encontrar un conjunto de puntos individuales en un espacio de estados cuya función de densidad de probabilidad se aproxime a los valores de probabilidad verdaderos. Considerando estos dos principios, se busca un conjunto de puntos determinísticos llamados puntos sigma. Después se aplica una función no lineal conocida a cada punto para obtener los puntos transformados. La media y covarianza de los puntos transformados proporcionarán un buen estimado de la verdadera media y covarianza.

Lo anterior puede generalizarse como sigue a continuación:

- Se considera un sistema no lineal en tiempo discreto n-dimensional dado por:

$$x_{k+1} = f(x_k, u_k, t_k) + w_k \quad (3.3.8)$$

$$y_k = h_k(x_k, t_k) + v_k \quad (3.3.9)$$

Donde w_k es el ruido de proceso con covarianza Q_k y v_k es el ruido de la medida con covarianza R_k

- Las siguientes expresiones describen la etapa de *predicción* mediante la propagación del estado estimado y la covarianza desde una medida hasta la próxima. Para propagar desde el tiempo k hasta un instante $k+1$, primero se eligen los puntos sigma $\hat{x}_k^{(i)}$:

- a) Se usa la ecuación no lineal del sistema f (regularmente es el jacobiano) para transformar los puntos sigma en los vectores $\hat{x}_k^{(i)}$:

$$\hat{x}_k^{(i)} = f\left(\hat{x}_{k-1}^{(i)}, u_k, t_k\right) \quad (3.3.10)$$

- b) Se suman los puntos sigma $\hat{x}_k^{(i)}$ para obtener la estimación en un instante del anterior:

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \quad (3.3.11)$$

- c) Se estima el error de covarianza en el instante anterior, añadiendo Q_{k-1} al final de la ecuación para tener en cuenta el ruido del proceso:

$$P_k^- = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^-) (\hat{x}_k^{(i)} - \hat{x}_k^-)^T + Q_{k-1} \quad (3.3.12)$$

- La etapa de *corrección* del filtro FKU se realiza con las siguientes expresiones:

- a) Se eligen puntos sigma $\hat{x}_k^{(i)}$ asociados a \hat{x}_k^- y a P_k^- :

$$\hat{x}_k^{(i)} = \hat{x}_k^- + \bar{x}^{(i)}, i = 1, \dots, 2n$$

$$\hat{x}_k^{(i)} = (\sqrt{nP_k^-})_i^T, i = 1, \dots, 2n$$

$$\hat{x}_k^{(n+i)} = -(\sqrt{nP_k^-})_i^T, i = 1, \dots, 2n \quad (3.3.13)$$

- b) Se obtiene el espacio de medidas utilizando la función no lineal h (regularmente es la matriz identidad) para transformar puntos sigma de las medidas $\hat{y}_k^{(i)}$:

$$\hat{y}_k^{(i)} = h(\hat{x}_k^{(i)}, t_k) \quad (3.3.14)$$

- c) Se suman los puntos sigma para obtener la medida en el próximo instante.

$$\hat{y}_k = \frac{1}{2n} \sum_{i=1}^{2n} \hat{y}_k^{(i)} \quad (3.3.15)$$

- d) Se estima la covarianza del error de la medida, añadiendo R_{k-1} para tener en cuenta el ruido de la medida:

$$P_k^y = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{y}_k^{(i)} - \hat{y}_k^-) (\hat{y}_k^{(i)} - \hat{y}_k^-)^T + R_{k-1} \quad (3.3.16)$$

- e) Se estima la covarianza cruzada entre \hat{x}_k^- y \hat{y}_k^- :

$$P_k^{xy} = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^-) (\hat{y}_k^{(i)} - \hat{y}_k^-)^T \quad (3.3.17)$$

- f) Finalmente se calcula la ganancia de Kalman K_k para actualizar los estimadores del estado y error de covarianza correspondiente \hat{x}_k^+ y P_k^+

$$K_k = P_k^{xy} (P_k^y)^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - \hat{y}_k)$$

$$P_k^+ = P_k^- - K_k P_k^y K_k^T \quad (3.3.18)$$

Los algoritmos de extensión del FK descritos anteriormente fueron seleccionados ya que consolidan las condiciones para ambientes no lineales como los cultivos agrícolas. La comparación de la veracidad y precisión de los movimientos de traslación y rotación del cuadricóptero, son los argumentos para establecer cuál de los dos filtros es el método que optimiza el posicionamiento y orientación del *drone* para el sistema de *sensado remoto* propuesto.

3.3 Diseño de la Estrategia de Posicionamiento

El método de posicionamiento que propone el proyecto está planteado como se aprecia en la figura 11. El primer paso para aplicar el filtro de Kalman es tomar el modelo del sistema (modelo de Euler en la sección 3.1.2) x_k y las mediciones desde la odometría, IMU y sensor de altura, para formar espacio y_k . El siguiente paso es iniciar la etapa de predicción para calcular la estimación del estado anterior $\hat{x}_{k/k-1}$ con su respectivo error de covarianza $P_{k/k-1}$. Cuando se consigue dicha estimación se pase a la etapa de corrección, la cual fusiona la estimación de estado final \hat{x}_k con el posicionamiento obtenido desde el receptor GPS, para soportar una navegación en exteriores. Esta fusión requiere un procedimiento de transformación de sistemas de referencia para poder ser integrados en el algoritmo. La transformación de coordenadas geográficas a coordenadas rectangulares UTM se realiza mediante las fórmulas de Cotichia-Surace publicadas en el “Bolletino di Geodesia e Science Affini”, Num. 1, con las cuales se puede lograr una precisión de un centímetro con la suficiente cantidad de decimales. Las formulas utilizan la geometría del elipsoide de Hayford, las coordenadas geodésicas en radianes y el huso horario de la localización del lugar.

Esta transformación se encuentra en el sistema de referencia ECEF, por lo cual se debe interpretar nuevamente mediante rotación, para ser transformado en sistema local, en el se calculó el espacio de estados corregido, para cualquiera de las extensiones del filtro de Kalman, FKE o FKU. La implementación de los filtros asume como entrada un espacio de estados con seis grados de libertad (x, y, z, pitch (φ), roll (θ), yaw (ψ)), velocidades y aceleraciones respectivas, obtenidas desde la odometría del AR Drone y de los datos directamente capturados por los sensores.

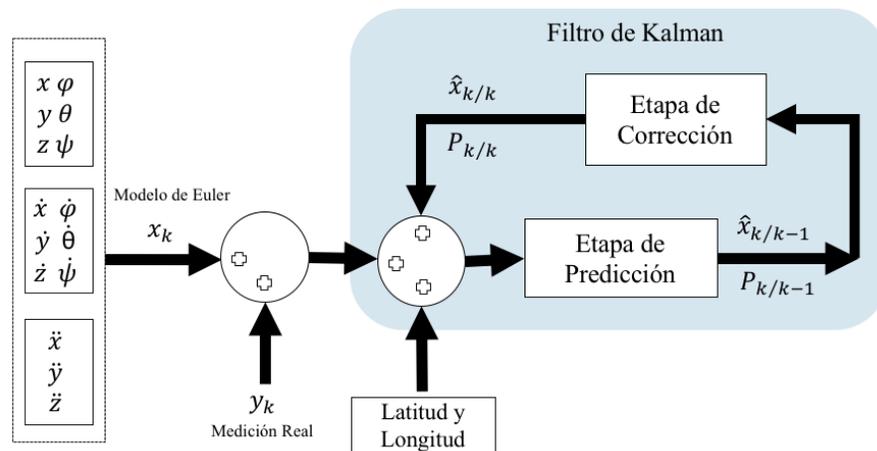


Figura 11. Estrategia de posicionamiento para AR Drone

3.4 Resumen

El capítulo 3 inicia con definición de la navegación de un robot aéreo tipo cuadricóptero, con lo cual fue posible construir un modelo físico-matemático, que después es utilizado en la estrategia para optimización de posicionamiento. El siguiente es la exploración de los sistemas de navegación que la plataforma Ar Drone soporta basado en sus sensores. De los métodos de navegación se destaca la unidad básica que viene de fábrica con el *drone*, la cual refleja los esfuerzos por dotar a la plataforma de robustez y precisión en los movimientos, utilizando una cámara vertical para capturar el flujo óptico y complementar la información de velocidad relativa que notifica la IMU. La navegación basada en visión por su parte, da un paso más a la fusión de sensores integrando la cámara frontal a los procedimientos de posicionamiento del *drone* desde una estación en tierra. Y sin quitarle importancia se describe la navegación utilizando un posicionamiento global que permite dirigir la navegación hacia ambientes como los cultivos.

Los métodos de posición y orientación son valorados cualitativamente de acuerdo a las características predominantes del posicionamiento para ambientes abiertos como los cultivos agrícolas, sin embargo, la literatura arroja que ninguno de los métodos por si solo es lo suficiente óptimo para este tipo entornos, por lo cual el proyecto selecciona una estrategia en la que utilizando filtros de Kalman, se logre una fusión sensorial que integre la navegación inercial del Ar Drone y la navegación por GPS disponible con FlightRecorder.

Planificación de Vuelo Para Sensado Remoto de Cultivos

Este capítulo aborda el principal objetivo del trabajo de maestría que es implementar un sistema de *sensado remoto* basado en una plataforma aérea de bajo costo, lo cual implica que no está diseñada para aplicaciones de alto impacto. Por lo tanto, es necesario optimizar su navegación en ambientes abiertos como se describió en el capítulo tres, y diseñar un plan de vuelo que le permita no solo gestionar eficientemente la batería disponible y sino también adquirir información completa del área de cobertura.

El plan de vuelo del sistema de *sensado remoto* propuesto se basa en la implementación de algoritmos heurísticos que calculen la mejor ruta de cobertura y en la determinación de los parámetros de vuelo. Los algoritmos seleccionados consideran un conocimiento previo de los obstáculos o zonas de poco interés, la georreferenciación del área y la dirección del viento. Los algoritmos implementados tienen origen en estrategias de cobertura diseñadas para robots móviles terrestres, además que todos ellos se apoyan en una técnica de retroceso para visitar toda el área. Finalmente, los parámetros de vuelo, están relacionados con la captura automática de imágenes, lo cual implica calcular la altura óptima de vuelo, el tiempo obturación de la cámara y el punto de partida de la ruta para adquisición de información de un cultivo.

4.1 Algoritmos de Cobertura Global para Robots Móviles

En concordancia con el capítulo anterior y el prototipo propuesto, la adquisición de datos confiables para agricultura de precisión mediante *sensado remoto*, implica que no solo se tenga un posicionamiento preciso en ambientes cambiantes como los cultivos, sino que también se realice monitorización completa de los cultivos. Dicha monitorización se realiza regularmente con una trayectoria en Zig-Zag que cubra el área completa, sin detenerse en la gestión eficiente de los recursos del *drone*, ni en las condiciones del área de cobertura, como por ejemplo zonas de poco interés. Por tanto, la concepción de estrategias que superen la cobertura tradicional, es una apuesta del presente trabajo.

Desde el punto de vista de la Robótica, el problema de cobertura completa es investigado desde hace varios años. La solución al problema de cobertura, utiliza robots móviles para realizar tareas como limpiar áreas, mapear calles, distribución de periódicos y monitoreo continuo de grandes zonas. Para el presente trabajo de

investigación el reto principal es construir una ruta de vuelo que permita crear una copia digital del espacio mediante imágenes. Los dos métodos para reconstrucción digital y/o vigilancia del espacio son denominados cobertura local y cobertura global. La cobertura local se basa en sensores como cámaras, láseres y radares portados en los robots. Con estos sensores el robot es capaz de localizarse y cubrir un área desconocida, percibiendo obstáculos y límites del área, para evitar colisiones. En cuanto a la cobertura global, los robots están limitados a tener un conocimiento previo del área de cobertura con sus respectivos límites y obstáculos.

De los dos métodos de cobertura nombrados anteriormente, este proyecto se enfoca en una cobertura global ya que la localización y mapeo simultaneo llevan implícito un coste computacional elevado lo cual requeriría un consumo de energía mayor, disminuyendo así el tiempo de vuelo. La cobertura global entonces se fundamenta en determinar la ruta de cobertura mediante búsqueda de caminos en grafos. La idea de modelar el mundo de un robot como un grafo conexo fue inspirada en el problema conocido por los investigadores como “El problema del agente viajero” o TSP por sus siglas en ingles [105]. El problema asume las ciudades (celdas libres) como nodos de un grafo conexo no dirigido y las distancias entre ellas como el peso de las aristas entre nodos. La mejor ruta, es la más corta que visite todos los nodos una sola vez. A continuación, se describe en detalle los algoritmos de cobertura de área seleccionados y sus consideraciones en implementación.

4.1.1 Modelo del Mundo para Algoritmos de Cobertura Global

Como se mencionó anteriormente el modelo del mundo es tratado como un grafo conexo. La teoría de grafos es parte de las soluciones matemáticas más utilizadas en aplicaciones de los últimos años como por ejemplo en el modelado de estructuras moleculares, en el diseño de redes de telecomunicaciones, y para el caso particular para planificación de rutas de un robot. Un grafo es una estructura de gráfica que representa un conjunto de vértices o nodos conectados por arcos o aristas.

Matemáticamente un grafo es denotado como G , el cual tiene vértices V y aristas E . Donde V es un conjunto finito de vértices de G y E es un conjunto de pares no ordenados de vértices de G , denotados como (a, b) , con $a \neq b$.

A continuación, algunas definiciones:

- La notación de un grafo es $G = (V, E)$
- El grado de un vértice es el número de vecinos que tiene en el grafo $gr(v)$
- Una ruta cíclica en $G = (V, E)$ es un camino con un sucesión finita de vértices $\{x_0, x_1, x_2, x_3 \dots, x_n\}$, donde $x_0 = x_n$ y cuyos vértices son todos distintos.
- Un sub-grafo es un subconjunto de vértices y aristas que pertenecen al grafo $h = (V_s, E_s) \in G$

Los grafos utilizados para implementar algoritmos de cobertura global, están clasificados como grafos conexos. Un grafo $G = (V, E)$ es conexo si entre cualquier par de vértices $a, b \in V$, existe un camino de vértices que los conecte. Un grafo conexo representa el mapa de celdas de un área de cobertura como se observa en

la figura 12, donde los puntos remarcados corresponden a los puntos centrales de celdas y las líneas punteadas a las aristas. Este grafo es catalogado como grafo tipo rejilla, ya que la distribución gráfica de los nodos es una red cuadrada uniforme. Esta representación puede dividirse en sub-grafos con grado 4 hasta formar el área completa.

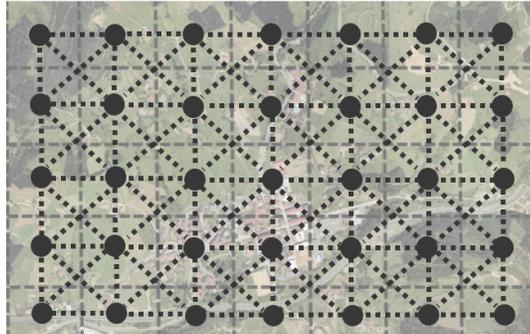


Figura 12. Representación del área de cobertura como un grafo tipo rejilla

En computación los grafos pueden representarse con estructuras como listas y matrices, o la combinación de ambas. Para la implementación de los algoritmos evaluados en este trabajo, se utiliza la representación por lista de adyacencia, ya que el uso de matrices implica un mayor uso de memoria. Esta lista de adyacencia atribuye a cada vértice una lista de vértices los cuales son adyacentes a él. Los nodos adyacentes son las celdas vecinas, que pueden tomar la configuración de Von Neumann (4 nodos vecinos) y Moore (8 nodos vecinos) como se exhibe en la figura 13.

	$C_{f-1,c}$	
$C_{f,c-1}$	$C_{f,c}$	$C_{f,c+1}$
	$C_{f+1,c}$	

(a)

$C_{f-1,c-1}$	$C_{f-1,c}$	$C_{f-1,c+1}$
$C_{f,c-1}$	$C_{f,c}$	$C_{f,c+1}$
$C_{f+1,c-1}$	$C_{f+1,c}$	$C_{f+1,c+1}$

(b)

Figura 13. Configuración de búsqueda de vecinos a) Von Neumann y b) Moore

La solución al problema de cobertura utilizando grafos está catalogada en computación como un algoritmo de complejidad NP. Un algoritmo de este tipo implica problemas de búsqueda y optimización para encontrar una mejor solución que las conocidas. Regularmente este tipo de problemas implican altos recursos computacionales, sin embargo, para la propuesta se ha delimitado dicha complejidad teniendo en cuenta las características del cuadricóptero, con la descomposición del área máximo en 100 celdas. Con esta cantidad límite de celdas y el modelo digital del

mundo se logra encontrar una ruta como menor costo computacional para cobertura de áreas.

La implementación para encontrar la ruta de cobertura óptima sobre el modelo del mundo está basada en rutas cíclicas que visiten todas las aristas del sub-grafo que contenga los nodos libres (figura 14). Los algoritmos que calculan la ruta de cobertura tienen como entrada una imagen digital del modelo del mundo que se va a mapear. El siguiente paso es determinar el umbral para etiquetar los pixeles que son límites y obstáculos; para lo cual se utiliza el mapa de grises de las bandas roja, verde y azul de acuerdo a la ecuación 4.1.1. Cuando el modelo está pre-procesado, inicia la descomposición del área y seguidamente se etiqueta cada celda como obstáculo, límite o libre, dependiendo del valor promedio de los pixeles que tiene el mapa. El paso final es construir un grafo cobertor con las celdas libres, obstáculos y límites, para que sea la entrada de los algoritmos de cobertura y con ellos obtener una lista de puntos específicos sobre el modelo digital, los cuales serán luego transformados en coordenadas reales por las cuales el robot debe pasar para cubrir el área completa.

$$Umbral = 0.8 * Banda Verde + 0.15 * Banda Roja + 0.5 * Banda Azul \quad (4.1.1)$$

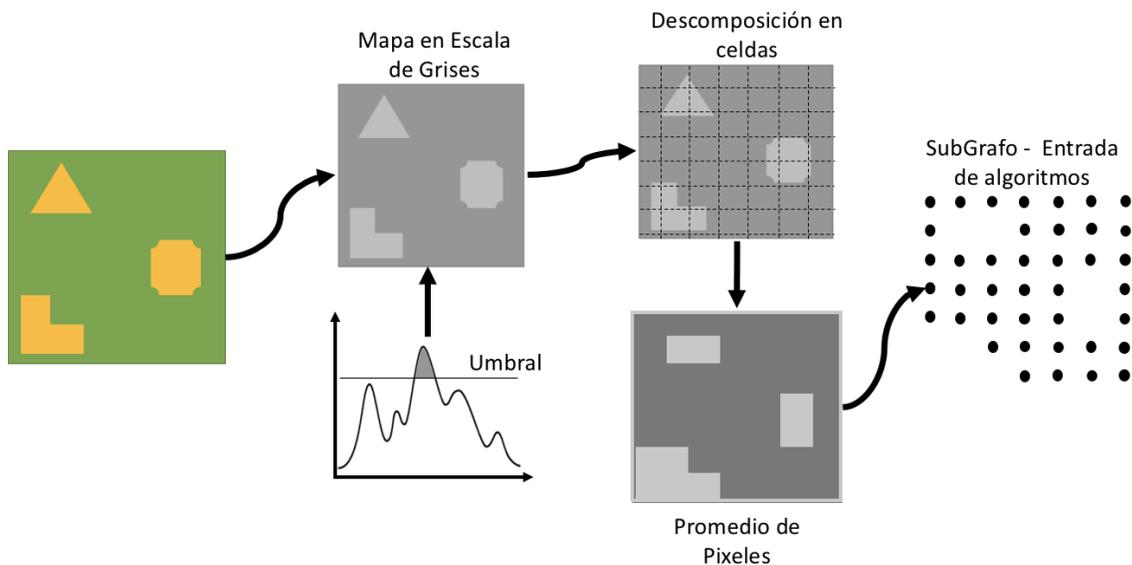


Figura 14. Procesamiento del modelo del mundo

4.2 Algoritmo De Cobertura basado en Dijkstra

El algoritmo planteado por Dijkstra en 1959 es un método para encontrar el camino más corto entre dos puntos diferentes de un mapa representado por un grafo. El grafo puede tener n nodos, los cuales están conectados por pares, y los pares de nodos se relacionan con pesos dados en las aristas. Dijkstra limita el problema para que una sola ruta exista entre dos puntos. Para resolver el problema matemático, el algoritmo

plantea dos retos: el primero, construir un árbol de longitud mínima (una ruta que conecte cada par de nodos) entre los nodos, y el segundo, es encontrar una ruta de mínima longitud entre dos nodos P y Q [106].

En computación la solución a los dos retos se interpreta realizando una búsqueda a profundidad, en inglés BFS de los nodos del mapa y luego adicionar nodos adyacentes a la ruta con el menor peso. El algoritmo marca todos los vértices como no visitados. La búsqueda inicia con un nodo inicial que se debe indicar previamente, seguidamente se encuentra el nodo adyacente más cercano al punto de origen, este nodo se toma como un punto intermedio y se realiza nuevamente la búsqueda del nodo más cercano al origen con las distancias actualizadas y se repite el proceso hasta llegar al nodo destino. El pseudocódigo 1 describe el algoritmo de Dijkstra con la premisa de que no puede tomar distancias negativas.

Pseudocódigo 1: Distancia mínima entre dos puntos

Dijkstra (Grafo G, Origen s, Meta g):

```

actual ← s
visitado[G] ← falso
Ruta ← s
Inicio ciclo
  vecino ← extraer_minimo (G, actual)
  Si vecino = g
    visitado[g] ← true
    salir del ciclo
  Sino
    actual ← vecino
    Ruta ← vecino
    visitado[vecino] ← true
Fin ciclo
Retornar (Ruta)

```

Para el algoritmo de cobertura que calcula la ruta de vuelo del cuadricóptero descrito en esta investigación, se plantea una alternativa desde los retos descritos por Dijkstra, adicionando una nueva condición a la ruta de mínima longitud para que visite todos los nodos. De esta manera el algoritmo de cobertura no solo expande el árbol cobertor mínimo que conecta los pares de nodos, sino que también lista todos los nodos no visitados. Para lograr tal objetivo, la implementación se apoya en encontrar los vecinos de la configuración Moore menos distantes del origen, en la inclusión de distancias negativas y en una técnica de Retroceso, para buscar los nodos no visitados. El algoritmo de cobertura se presenta a continuación en el pseudocódigo 2.

Pseudocódigo 2: Ruta de cobertura basada en Dijkstra

Cobertura_Dijkstra (Grafo G, Origen s):

```

actual ← s
visitado[G] ← falso
Ruta ← s
Inicio ciclo
  vecino ← extraer_minimo (G, actual)
  Si vecino = ∞
    vecino ← Retroceso(Ruta, vecino)
  Si vecino = s
    salir del ciclo
  Sino
    Si vecino = s
      salir del ciclo
  actual ← vecino
  Ruta ← vecino
  visitado[vecino] ← true
Fin ciclo
Retornar (Ruta)

```

El algoritmo de cobertura descrito anteriormente cumple con dos condiciones particulares para los propósitos de planificación de vuelo con robots aéreos:

1. El nodo inicial y el de destino son los mismos, por tanto, el nodo inicial no es etiquetado como visitado.
2. Los nodos adyacentes a los nodos intermedios están determinados por la configuración de Moore.

La figura 8 visualiza el proceso para encontrar la ruta óptima de cobertura con el algoritmo basado en Dijkstra. De acuerdo a la configuración de Moore los nodos vecinos que se encuentran a la derecha (W), izquierda (L), arriba (U) y abajo (D) tienen una distancia constante al nodo evaluado de 1 y por tanto los vecinos en la esquina superior izquierda (SL), esquina superior derecha (SW), esquina inferior derecha (IW) y esquina inferior izquierda (IL), tiene una distancia de $\sqrt{2}$ (figura 15(a)). La selección del siguiente paso en la ruta toma inicialmente los nodos más cercanos y en el caso que tenga más de una opción, siempre escoge el vecino cercano en el sentido contrario de las manecillas del reloj, iniciando desde el nodo W (figura 15(b)). El algoritmo continúa su búsqueda de los vecinos libres, no visitados y menos distantes, en el caso que no tenga vecinos libres con la menor distancia, selecciona los vecinos de las esquinas en el sentido contrario de las manecillas del reloj (figura 15(c) y 15(d)). Si en la búsqueda de los nodos de la ruta, no encuentra vecinos libres como se observa en la figura 15(e), se inicia el mecanismo de Retroceso hasta que encuentre un nodo con vecinos no visitados (figura 15(f)). El proceso de búsqueda continua y si es necesario retoma cuantas veces sea necesario el mecanismo de Retroceso hasta encontrar el nodo de partida (figuras 15(g) - 15(i)).

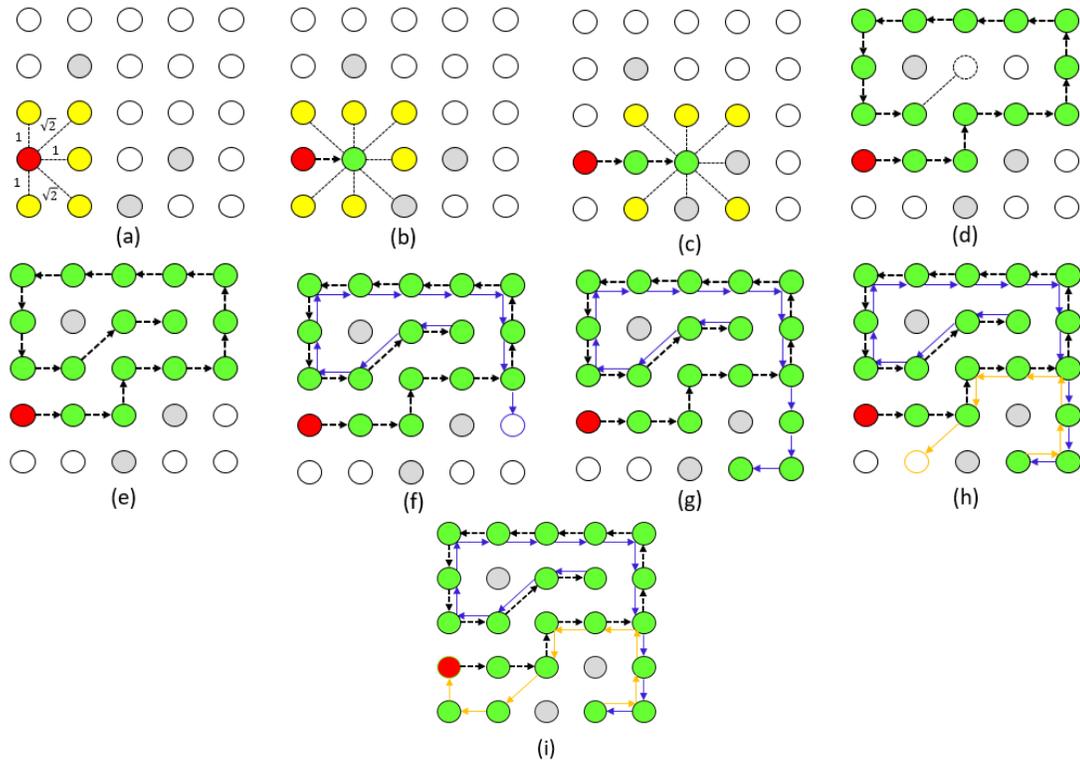


Figura 15. Ruta de cobertura basada en Dijkstra

4.3 Algoritmo De Cobertura con Frente de Onda

Al igual que el algoritmo de Dijkstra, el algoritmo basado en transformada de distancia tiene su primer planteamiento para encontrar la ruta más corta entre dos nodos diferentes pertenecientes a un grafo conexo. La característica de este algoritmo es que la transformada de distancia se comporta como un frente de onda que se propaga en los nodos libres del grafo que representa el mapa, partiendo desde la meta, por lo que algoritmo es conocido como Propagación de Frente de Onda. Para encontrar la ruta de longitud mínima sobre el mapa, el algoritmo inicia en el origen buscando los nodos vecinos con el valor de la menor transformada, de manera consecutiva hasta llegar a la meta [107].

En computación el algoritmo realiza una búsqueda a profundidad, para conectar los nodos del mapa que representan el entorno del robot de manera digital, para luego etiquetar cada nodo con un valor de transformada. Para expandir el frente de onda, el algoritmo inicia una búsqueda con los vecinos de la configuración de Moore, asignando un valor de 0 al nodo destino (figura 16(a)). Los nodos adyacentes al nodo destino tendrán el valor de la transformada del nodo origen incrementado en uno "1" (figura 16(b)). El nodo adyacente, será un nodo intermedio para buscar los vecinos a los cuales se les asigna un valor de transformada incrementado en 1 con respecto al valor del nodo intermedio, y de esta manera se repite el proceso hasta completar

todos los nodos del mapa (figuras 16(c) - 16(i)). Para encontrar la ruta de longitud mínima, el algoritmo toma el origen y busca los nodos con el valor más pequeño hasta encontrar el cero correspondiente a la meta.

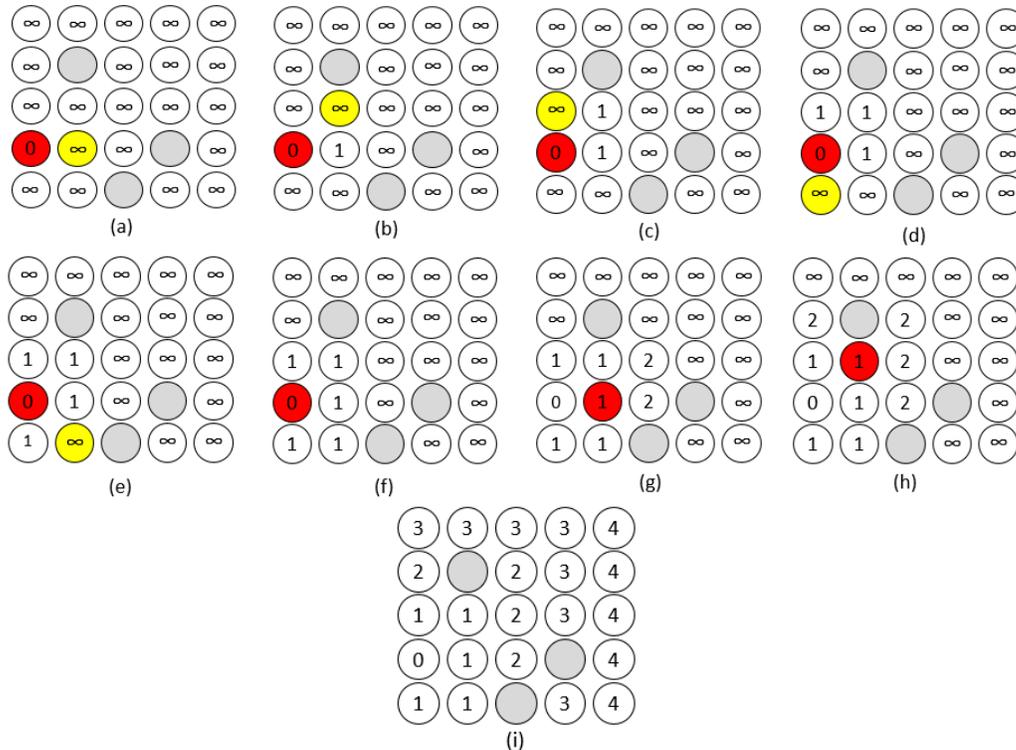


Figura 16. Propagación de frente de onda en un grafo

Para lograr establecer la ruta óptima de cobertura, se realizan algunas modificaciones al algoritmo inicial como, no listar los nodos vecinos con menor valor de transformada (VT) sino con el mayor valor, y en el caso que dos nodos adyacentes tengan el mismo valor de transformada, el algoritmo selecciona el vecino más cercano al origen, en el sentido contrario de las manecillas del reloj. El pseudocódigo 3 describe el algoritmo de cobertura basado en la propagación del frente de onda.

Pseudocódigo 3: Ruta de cobertura basada en Propagación de Frente de Onda

Cobertura_FrenteOnda (Grafo G, Origen s):

```

actual ← s
visitado[G] ← falso
Ruta ← s
Etiqueta(G) ← propagación_onda(G)
Inicio ciclo
  vecino ← maximoVT (Etiqueta, actual)
  Si vecino = ∞
    vecino ← Retroceso(Ruta, vecino)

```

```

    Si vecino = s
      salir del ciclo
    Sino
      Si vecino = s
        salir del ciclo
      actual ← vecino
      Ruta ← vecino
      visitado[vecino] ← true
    Fin ciclo
  Retornar (Ruta)
  
```

El algoritmo de cobertura descrito anteriormente cumple con dos condiciones particulares para los propósitos de planificación de vuelo con robots aéreos:

1. El nodo inicial y el de destino son los mismos, por tanto, el nodo inicial no es etiquetado como visitado.
2. Si el último nodo intermedio adicionado a la ruta no tiene como nodo adyacente el nodo destino, entonces se inicia un Retroceso en la lista de nodos de la ruta de cobertura para buscar nodos no visitados.

La figura 17 visualiza el proceso gráfico para encontrar la ruta óptima de cobertura con el algoritmo basado en propagación de frente de onda.

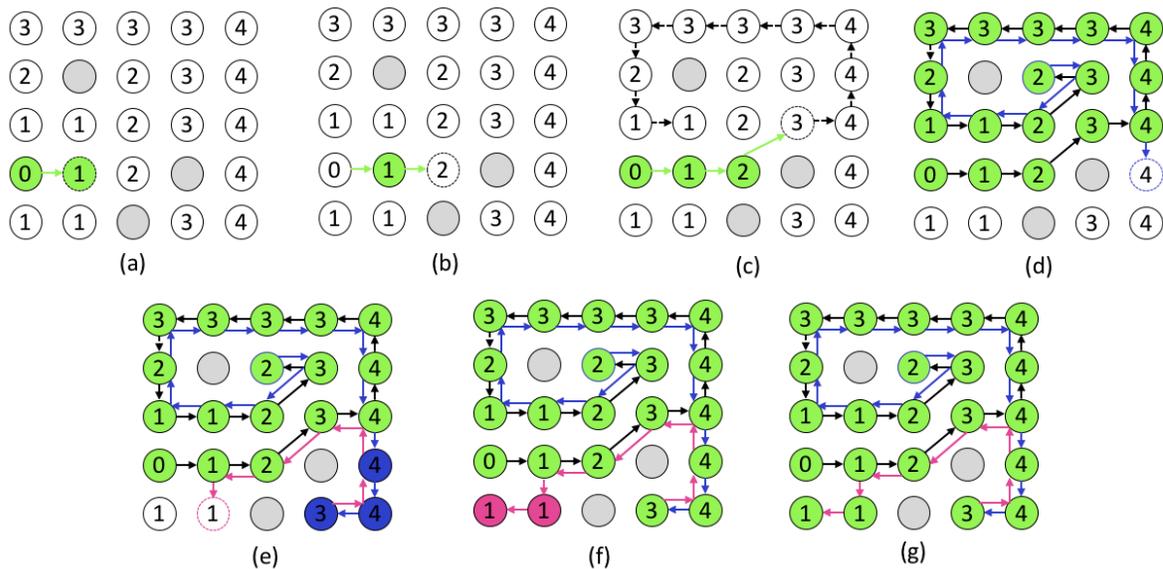


Figura 17. Ruta de cobertura basada en Transformada de Distancia

Después de tener el mapa de nodos del entorno etiquetados como en la figura 16, el paso siguiente es seleccionar los nodos que harán parte de la ruta de cobertura. El algoritmo planteado toma la configuración de Moore para encontrar el próximo vecino, iniciando en el nodo W y continuando la búsqueda en el sentido contrario de las manecillas del reloj hasta encontrar el de mayor valor de transformada (figuras 17(a),

17(b)). La posición de cada nodo se va almacenando en una lista de puntos de paso hasta que encuentre ya no encuentre vecinos libres (figura 17(c)), cuando esto sucede se inicia el proceso de Retroceso en la búsqueda de vecinos libres cuantas veces sea necesario hasta encontrar el nodo de partida (figuras 17(d)-17(g)).

4.4 Algoritmo De Cobertura en Espirales

Este algoritmo es una alternativa para evitar el número de giros en los algoritmos de cobertura anteriormente descritos. Las trayectorias en espiral facilitan de una manera indirecta la localización del robot, ya que en su primer ciclo siguen los obstáculos o límites del área, y son más robustas en comparación con las trayectorias en zig-zag, resultado de los algoritmos anteriores.

La implementación del algoritmo en computación está basada en un pequeño conjunto de reglas, las cuales permiten una cobertura apropiada con una demanda mínima de recursos de memoria. Y aunque inicialmente el algoritmo es planteado para cobertura on-line, este proyecto realiza la extensión en las reglas, para que el algoritmo tome como entrada el mapa del ambiente en el que se va a localizar. El planteamiento de la solución al problema de cobertura completa estudia dos cuestiones: la primera es la manera de generar trayectorias para cubrir regiones simples y la segunda es la búsqueda de una relación entre las regiones simples y la superficie total que se quiere cubrir. El desarrollo de la primera cuestión considera cubrir áreas simples mediante rutas en espiral la cuales se conforman de anillos concéntricos, que finalizan cuando el robot está rodeado de celdas ya visitadas. Las condiciones para este tipo de cobertura implican que el robot toma regiones cuadradas y que el cubrimiento de una región simple inicia identificando sus obstáculos. Para la cuestión de enlazar las rutas en espiral de cada región simple, el algoritmo usa la técnica de Retroceso para buscar las próximas regiones simples a conectar [59].

La implementación de este algoritmo para el proyecto de investigación requirió algunas modificaciones en el planteamiento descrito inicialmente, ya que la identificación de los obstáculos está prevista en el mapa. En efecto, las rutas en espiral consideran los obstáculos laterales (izquierda o derecha), los cuales pueden ser reales o virtuales, como es el caso de los límites y las celdas visitadas. Además, la identificación de los obstáculos depende de la orientación del robot, la cual, en el caso particular, siempre se establece paralela al obstáculo. El siguiente paso es utilizar Retroceso de la ruta en espiral para encontrar regiones vecinas en la cuales se despliegue una nueva espiral. De acuerdo a lo anterior, el algoritmo realiza una ruta en espiral sobre una región simple, considerando la orientación del robot paralela a los obstáculos, desde lo cual se estudian los casos de la figura 18. Los movimientos del robot siguen una configuración de Von Neumann.

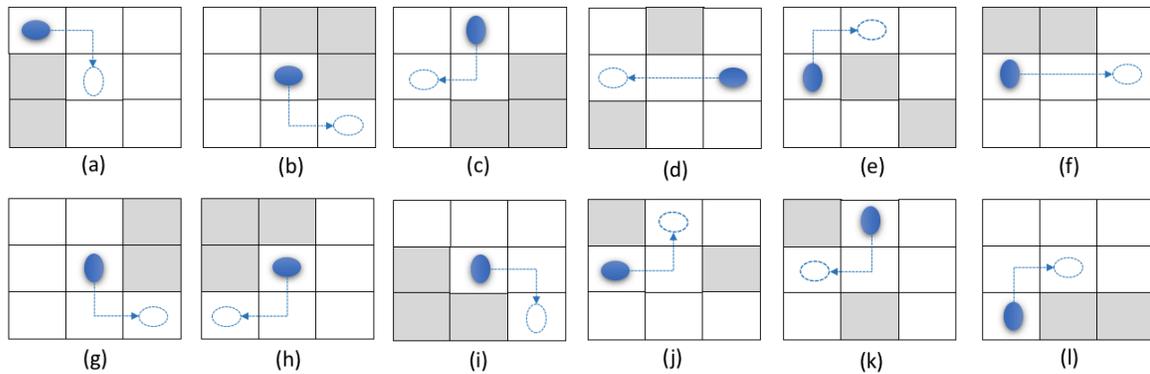


Figura 18. Casos de estudio para construir espirales en una región simple

De acuerdo a la figura 18, si el robot encuentra obstáculos laterales a su orientación, avanza hacia el frente y busca vecinos libres laterales (figura 18(a) y figura 18(l)). Si el robot encuentra un obstáculo en el frente busca vecinos laterales libres, en el caso de tener más de uno, selecciona el vecino en el sentido contrario de las manecillas del reloj (figura 18(b), figura 18(h), figura 18(i), figura 18(f)). Si el robot no tiene obstáculos siempre avanza (figura 18(c) y figura 18(d)) y finalmente si el robot solo tiene un obstáculo lateral, avanza y busca nodos libres laterales (figura 18(g), figura 18(j) y figura 18(k).) El pseudocódigo 4 describe el algoritmo de cobertura basado en espirales y los casos de la figura 18.

Pseudocódigo 4: Ruta de cobertura basada en Espirales

Cobertura_Espiral (Grafo G, Origen s):

```

actual ← s
visitado[G] ← falso
Ruta ← s
Inicio ciclo
  vecino ← busqueda_frontal (G, actual)
  Si vecino = ∞
    vecino ← busqueda_laterales (G, vecino)
    Si vecino = ∞
      vecino ← Retroceso (Ruta, vecino)
      Si vecino = ∞
        salir del ciclo
    actual ← vecino
    Ruta ← vecino
    visitado[vecino] ← true
  Fin ciclo
  Retornar (Ruta)

```

La figura 19 visualiza el proceso gráfico para encontrar la ruta óptima de cobertura con el algoritmo basado en espirales, el cual considera las reglas particulares de la figura 19, en las que la referencia de los movimientos siempre está relacionada con el frente del robot y los obstáculos cercanos. En la figura 19(a) el algoritmo toma el nodo como punto de paso de la ruta, ya que el obstáculo más cercano es el límite de área y la búsqueda siempre se realiza en el sentido contrario de las manecillas del reloj. El algoritmo continúa tomando los nodos cercanos al límite de área hasta encontrar una zona restringida u obstáculo (figuras 19(b) - 19(d)), con el cual el sentido del robot debe cambiar para rebordear el obstáculo. De acuerdo a las reglas, la ruta toma los vecinos que formen una espiral completa en la región simple hasta no encontrar vecinos libres (figuras 19(e) y 19(f)). En tal caso, se inicia el proceso de Retroceso buscando el próximo vecino libre cercano para iniciar una nueva espiral de región simple (figura 19(g)), este proceso se realiza las veces que sea necesario hasta que se visiten todos los nodos libres (figuras 19(h) – 19(j)).

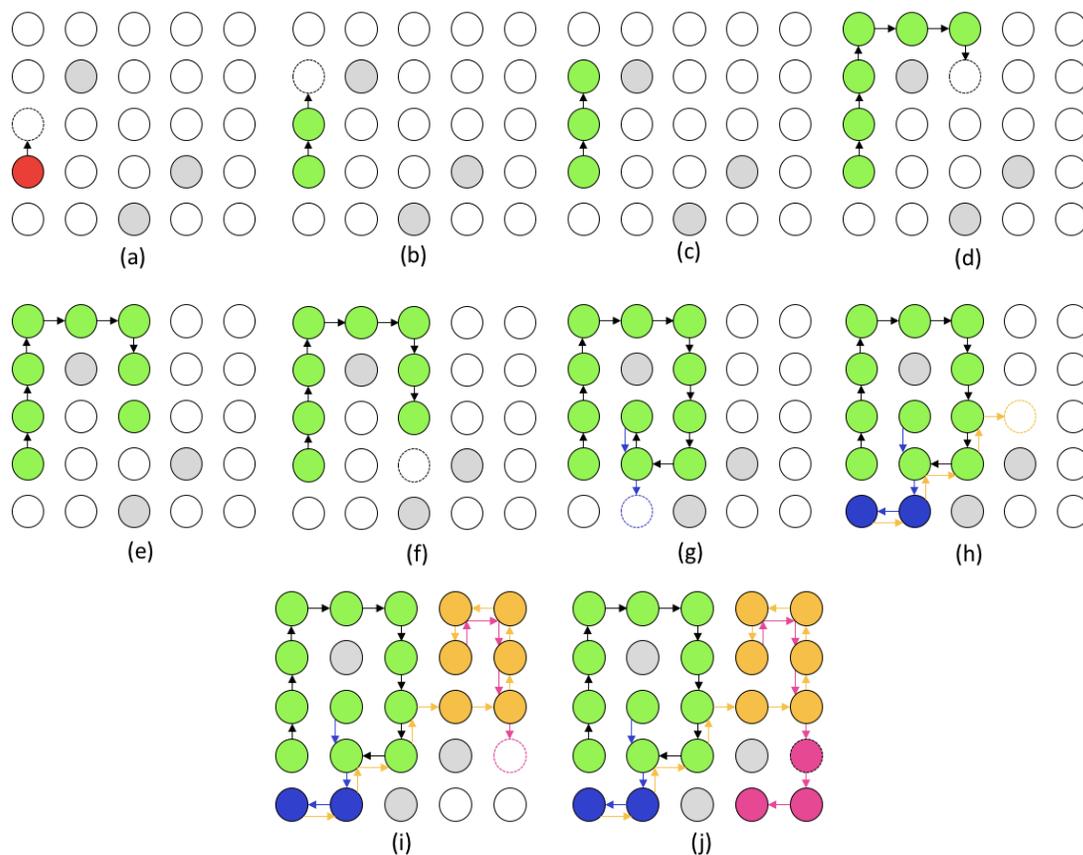


Figura 19. Ruta de cobertura basada en Espirales

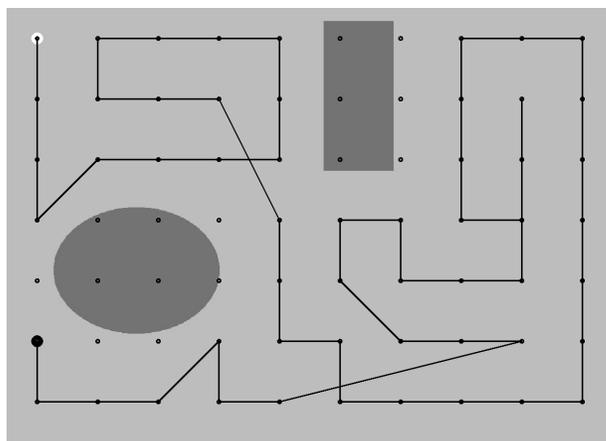


Figura 22. Ruta de cobertura basada en Espirales

Algoritmo	Tiempo (ms)	Giros	Cobertura de nodos libres	Revisitas
Cobertura basada en Dijkstra	16,012	22	55 de 55	9
Cobertura con Frente de Onda	16,351	25	55 de 55	29
Cobertura en Espirales	15,717	29	55 de 55	28

Tabla 4. Comparación de características computacionales de los algoritmos de ruta de cobertura

En el ejemplo anterior la descomposición del área resulta en 70 celdas con y 15 celdas como obstáculo o zonas de poco interés. Las figuras 20, 21 y 22 representan las rutas que los algoritmos basados en Dijkstra, propagación de frente de onda y en espirales calculan de acuerdo a un área de prueba simulada. El propósito de la simulación es realizar a nivel computacional, una validación de los algoritmos, la cual se describe en la tabla 4. De acuerdo a los resultados de la planificación se tienen las siguientes inferencias:

- Los algoritmos realizan cobertura completa de todos los nodos libres, siempre que exista un mecanismo de Retroceso.
- La ruta calculada con el algoritmo de cobertura basado en Dijkstra tiene menos giros y menor número de revisitas.

- El tiempo que la maquina toma para calcular las rutas de cobertura es similar en cada algoritmo, sin embargo, el tiempo que toma calcular las rutas de cobertura en espirales, es menor al de los otros algoritmos.
- El algoritmo de cobertura basado Frente de onda es propenso a atravesar obstáculos en el proceso de Retroceso.

Las inferencias anteriores hacen parte del argumento para seleccionar un algoritmo óptimo en la tarea de *sensado remoto*. Sin embargo, esta comparación no puede ser definitiva ya que el comportamiento del robot no solo depende de la computación de los algoritmos sino de las pruebas en ambientes reales y las diferentes áreas de cobertura.

4.6 Plan de Vuelo para Sensado Remoto

El desarrollo de esta última fase del proyecto se enfoca en llevar a cabo la adquisición de datos de un cultivo agrícola, utilizando como sensor remoto una cámara digital y como plataforma aérea el cuadricóptero AR Drone. En este procedimiento se realiza la captura de fotografías aéreas de acuerdo a las características del sensor, el cuadricóptero, y a la aeronavegabilidad de los cultivos agrícolas.

La técnica de *sensado remoto* en cultivos agrícolas propuesta para el proyecto expone una estrategia para adquisición de imágenes que garantice la obtención de información completa gestionando un tiempo de captura óptimo.

4.6.1 Proceso de captura automática de imágenes

Esta estrategia es desarrollada utilizando los servicios de ROS (paquete *image_view*) para controlar la cámara frontal de AR Drone HD, mediante la cual se observan el área de cobertura. Los parámetros de calibración de la cámara se describen a continuación:

- Ancho de la imagen (píxeles): 640
- Altura de la imagen (píxeles): 360
- Matriz de la Cámara:

569.883158064802	0	331.403348466206
0	568.007065238522	135.879365106014
0	0	1

- Coeficientes de Distorsión:

[-0.526629354780687, 0.274357114262035, 0.0211426202132638, -0.0063942451330052, 0]

- Matriz de rectificación:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Matriz de proyección:

$$\begin{bmatrix} 463.275726318359 & 0 & 328.456687172518 & 0 \\ 0 & 535.977355957031 & 134.693732992726 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

El proceso para captura de información está basado en los principios básicos de sensado remoto [108] y se conforma de las siguientes dos fases:

Fase 1: Cálculo de la altura de vuelo

El primer paso para realizar la adquisición de información mediante imágenes es considerar la altura de vuelo, la cual se calcula conociendo parámetros de la cámara del AR Drone, como la distancia focal f_c (18 mm), longitud de la imagen en el objetivo de la cámara l_i (38 mm) y su equivalente real l_r , de acuerdo a la relación de la ecuación 4.6.1 (figura 24).

$$\frac{f_c}{l_i} = \frac{H}{l_r} \quad (4.6.1)$$

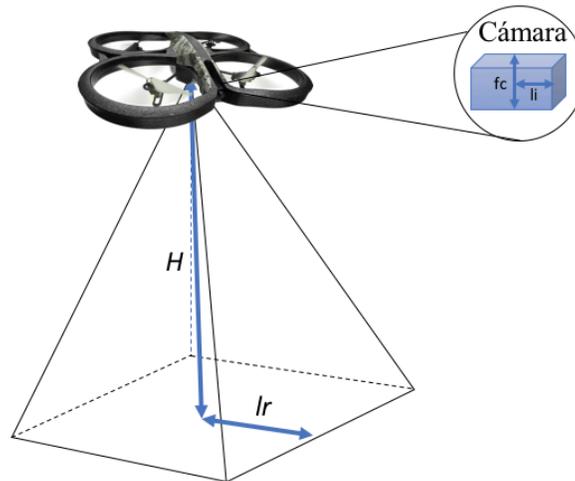


Figura 23. Altura de Vuelo

La longitud de la fotografía real l_r se determina desde el módulo de planificación de ruta de cobertura (ecuación 4.6.2), la cual es el resultado del número de columnas y

filas ($n \times m$) en las que fue descompuesta el área de cobertura y las dimensiones en metros del terreno ($W \times L$).

$$l_r = \sqrt{\left(\frac{W}{m}\right)^2 + \left(\frac{L}{n}\right)^2} \quad (4.6.2)$$

Fase 2: Cálculo del intervalo de tiempo entre cada fotografía

El siguiente paso es calcular el intervalo de tiempo t_c entre cada fotografía capturada, esto se obtiene considerando la distancia entre dos puntos de paso D consecutivas en el frente (figura 24) y la velocidad de la aeronave v_{drone} , que para el caso del AR Drone es 1.2 m/s (ecuación 4.6.3). De acuerdo a lo anterior y a las ecuaciones de fotogrametría propuestas en [], el tiempo de captura de fotografías está dado por:

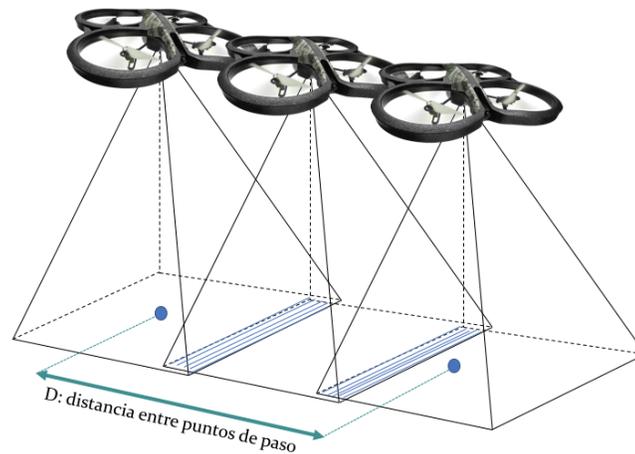


Figura 24. Distancia entre puntos de paso

$$t_c = \frac{D}{v_{drone}} \quad (4.6.3)$$

donde D esta definida como la ecuación 4.6.4, siendo l la dimensión de la fotografía y p el porcentaje de solapamiento frontal de las fotografías el cual es fijado en 63%.

$$D = l * \left(1 - \frac{p}{100}\right) \quad (4.6.4)$$

Para encontrar l se referencia la ecuación 4.6.5, donde l_i es la dimensión de la imagen en el sensor y f es la distancia focal, ambos definidos en los parámetros de la cámara del AR Drone.

$$D = \frac{l_i * H}{f} * \left(1 - \frac{p}{100}\right) \quad (4.6.5)$$

Finalmente, el tiempo de captura se obtiene reemplazando todos los valores conocidos y queda definido como la ecuación 4.6.6.

$$t_c = \frac{l_i * H}{f * v_{drone}} * \left(1 - \frac{p}{100}\right) \quad (4.6.6)$$

4.6.2 Punto de Partida de Ruta de Cobertura

El particular relieve y las condiciones atmosféricas propias de un país como Colombia, impiden realizar tareas de *sensado remoto* similares a los de otros países lejanos del trópico. Ya que, no se tiene certeza de las condiciones externas del entorno donde se planea previamente la misión con el *drone*. Condiciones ambientales como el viento, tienen un comportamiento variable localmente (vientos locales) que puede ser ignorado en la precisión del plan de vuelo.

Esta variable atmosférica afecta el vuelo de los robots aéreos multirrotor, ya que su capacidad de suspensión y avance depende en gran manera de la intensidad y dirección del viento. El efecto de los vientos sobre el robot repercute en un consumo mayor de batería, y aún más, en la deriva o pérdida inesperada del *drone*. Estas observaciones llevan a plantear un punto de partida del vuelo que implique la gestión eficiente de la batería para la misión de *sensado remoto* de cultivos. Para establecer la restricción, se tiene como premisa desde la literatura que un vuelo seguro se debe realizar con una velocidad del viento de a 2.5 m/s (5 nudos en aeronavegación) o preferiblemente en calma [109]. Si la intensidad del viento no supera el límite, entonces el vehículo aéreo debe ser maniobrado para que la orientación de la línea de vuelo sea congruente con la dirección del viento. Para considerar la dirección del viento se realizan pruebas (ver capítulo de Experimentación y Evaluación) de consumo energético experimentales con las cuales se determina que un robot aéreo multirrotor como el AR Drone tiene un gasto menor de batería si la mayoría de su vuelo se realiza atravesando la dirección del viento. Esta condición y la configuración de los algoritmos (rutas buscan los vecinos cercanos siguiendo el sentido contrario de las manecillas del reloj) dan pie para fijar la posición inicial de la misión de la siguiente manera.

- Viento desde el norte: posición de partida en la esquina inferior izquierda del mapa
- Viento desde el sur: posición de partida en la esquina superior derecha del mapa
- Viento desde el este: posición de partida en la esquina inferior derecha del mapa
- Viento desde el oeste: posición de partida en la esquina superior izquierda del mapa

4.7 Resumen

En este capítulo se describe de manera detallada los algoritmos para el cálculo de ruta de cobertura con sus respectivas limitaciones. Los algoritmos tienen como principio la representación del mundo que van a cubrir en un grafo conexo, el cual no excede en más de 100 nodos. Los tres algoritmos de cobertura en mención basado en Dijkstra, propagación de frente de onda y rutas en espiral, están implementados con una configuración de celdas vecinas de acuerdo a Von Neumann o Moore. Con base en ello, los algoritmos son comparados bajo métricas como el tiempo que toma calcular la ruta, número de giros, porcentaje de éxito de cobertura, celdas re visitadas y número de choques. El análisis de la comparación es clave para determinar las ventajas y desventajas de cada algoritmo, sin embargo, solo hace parte de la evaluación del componente de planificación de vuelo diseñado para el sistema de *sensado remoto* que se propone en el proyecto. Finalmente, se describen los parámetros considerados para planear una misión de *sensado remoto* con AR Drone, los cuales son la altura de vuelo, el intervalo de tiempo entre cada captura fotográfica y el punto de partida de las rutas de cobertura a desplegar.

Experimentación y Evaluación

En este capítulo se describe el prototipo implementado para concluir con la propuesta de investigación. El prototipo está basado en el diseño del proceso de ejecución de la misión de *sensado remoto* y en una arquitectura de control híbrida diseñada para integrar el AR Drone con los componentes software desarrollados en ROS. Dichos componentes están orientados a la optimización de la posición mediante fusión sensorial y a la eficiencia del *sensado remoto* mediante rutas de cobertura heurísticas y consideraciones del entorno. El capítulo también describe el plan de pruebas para validar el prototipo implementado y sus respectivos resultados de los cuales se destaca la precisión del filtro FKE para posicionamiento del AR Drone, y los resultados de los algoritmos de cobertura en conjunto con los respectivos mosaicos de fotografías adquiridos en cada ruta.

5.1 Prototipo

El prototipo implementado para el proyecto nombrado en este documento está conformado por un conjunto de componentes hardware y software. Los componentes principales son: el robot aéreo multirroto, que para el caso es el cuadricóptero AR Drone versión 2.0, sistema de procesamiento en vuelo, RaspberryPi 2 modelo B, que se encarga de ejecutar la misión y optimizar el posicionamiento; y una estación en tierra, con la cual se planifica las rutas de cobertura. De acuerdo a los componentes, el despliegue del *sistema de sensado remoto* en un área para agricultura de precisión sigue el procedimiento establecido en el diagrama de la figura 25.

El diagrama indica que el primer paso es desplazar la plataforma aérea integrada y la estación en tierra a la zona de monitoreo elegida. El siguiente paso es georreferenciar el lugar con herramientas como QGIS en la estación en tierra. Con el mapa digital georreferenciado, se introducen parámetros del entorno para contextualizar el software de planificación. Al ejecutar el software se obtiene la mejor ruta de cobertura comparando los tres algoritmos heurísticos planteados en el capítulo 4. La ruta seleccionada por el software es exportada hacia el sistema de procesamiento en vuelo, el cual espera la orden para iniciar la misión y ejecutar el algoritmo de Kalman que optimiza el posicionamiento del *drone*. Finalmente, al terminar la misión, se cargan las fotografías capturadas y se realiza el procesamiento adecuado para obtener el mosaico final.

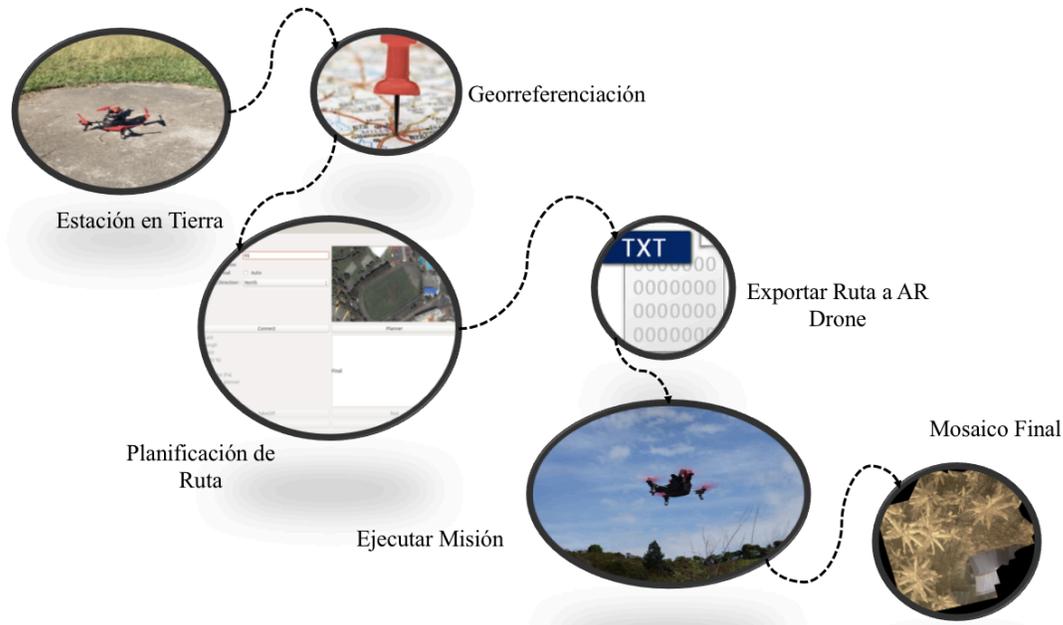


Figura 25. Procedimiento para ejecución de sensado remoto

5.1.1 Arquitectura de Control para AR Drone

De acuerdo al procedimiento diseñado para la misión de la figura 25, se pueden destacar los dos componentes software principales, los cuales están soportados en el sistema operativo ROS versión Indigo. El software en tierra implementado incluye un planificador de cobertura basado en algoritmos heurísticos, y el software en la plataforma aérea optimiza el posicionamiento y ejecuta las misiones del cuadricóptero. Estos componentes hacen parte de la arquitectura de control diseñada para el prototipo como se observa en la figura 26. La arquitectura es híbrida, es decir, se combinan dos paradigmas de control denominados reactivo y deliberativo. El paradigma reactivo conserva la lógica de acción y reacción, lo que implica una conexión directa entre los sensores y actuadores del robot, con la idea de no comprometer procesamientos complejos para percibir el mundo. El paradigma deliberativo realiza una serie de pasos lógicos y complejos para ejecutar las acciones apropiadas, lo cual implica tener una previa interpretación del entorno.

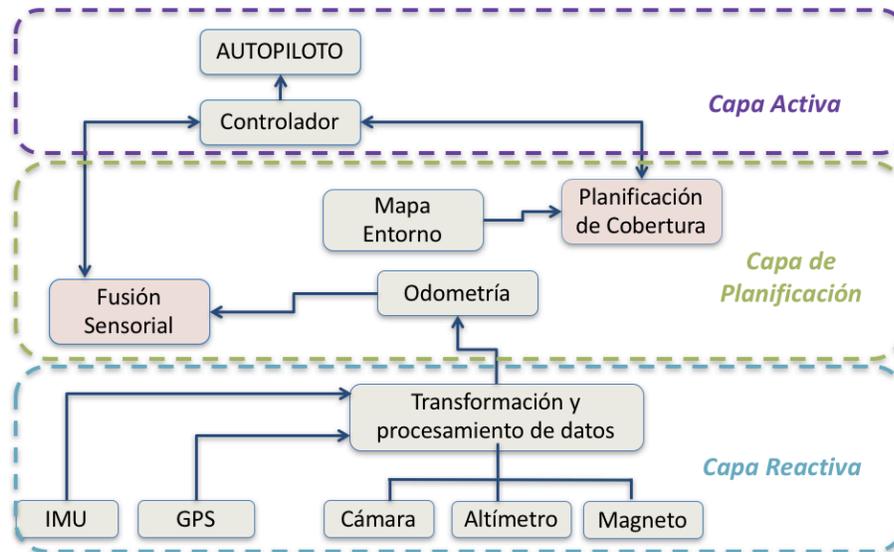


Figura 26. Arquitectura de control para un sistema de sensorio remoto basado en AR Drone

Para el caso de la arquitectura propuesta, la capa reactiva precisa los movimientos relativos del AR Drone para que la traslación y rotación correspondan con las decisiones de la capa deliberativa. Para tal objetivo, se utilizan los sensores del entorno como acelerómetros, giróscopos (IMU), GPS, cámaras, altímetro y magnetómetro para determinar la posición y orientación (odometría) del robot en un sistema de referencia local. La capa deliberativa, llamada en la arquitectura como capa de planificación, analiza los datos percibidos del entorno y la interpretación previa del ambiente que rodea el robot aéreo, para tomar decisiones de los próximos movimientos del AR Drone, considerando algoritmos de planificación de cobertura y algoritmos de fusión de datos para obtener un posicionamiento preciso. El resultado del procesamiento de la información se transforma en comandos de control que se ejecutaran en la capa activa, y son interpretados por los actuadores controlados por el autopiloto.

Los módulos en sombra roja de la figura 26, se han remarcado para declarar que son los principales componentes software desarrollados para el prototipo del sistema de *sensorio remoto* propuesto. El módulo de fusión de datos de la capa de planificación se encarga de tomar los datos formateados en ROS del AR Drone para calcular una odometría resultante. Finalmente, el módulo de planificación de cobertura, define los puntos de paso del robot aéreo sobre la zona a monitorizar, mediante algoritmos de planificación global. Las siguientes secciones describen en detalle la configuración e implementación de cada uno de los componentes declarados anteriormente.

5.1.2 Fusión Sensorial

Como se describió en el capítulo 3, el proyecto propone optimizar el posicionamiento del cuadricóptero AR Drone mediante una estrategia de fusión sensorial, conocida

filtro de Kalman extendidos, utilizando la información de la unidad inercial, sensores de altura (sonómetro y altímetro) y receptor GPS *Flight Recorder*. La comunicación entre el control de posicionamiento y el AR Drone se realiza mediante una red Wi-Fi con un retardo en la transmisión desde el cuadricóptero de 130 milisegundos y para el envío de comandos de 60 ms.

5.1.2.1 Implementación de Fusión de Datos en ROS

La implementación de la estrategia de posicionamiento se realiza utilizando el framework del sistema operativo ROS [97]. De acuerdo a lo anterior, el subsistema de posicionamiento implementado en la Raspberry Pi utiliza los nodos de la figura 27 relacionados en la tabla 5, los cuales tienen como centro el paquete ROS llamado *ardrone_autonomy*, que es el controlador del AR Drone (*ardrone_driver*) basado en el SDK oficial de Parrot. Este paquete además especifica los mensajes y tópicos para el intercambio de información con la IMU (*ardrone/imu*), los sensores de altura (*ardrone/navdata_altitude*), el receptor GPS (*ardrone/navdata_gps*), la cámara frontal (*ardrone/front/image_raw*), y con la estimación de posición y orientación (*ardrone/odometry*).

La frecuencia de actualización de la información es de 200 Hz, sin embargo, se realiza una relación de correspondencia con los datos del GPS (de acuerdo al timestamp), ya que este solo se actualiza cinco veces por segundo. La implementación de la estrategia planteada requirió formatear los tópicos de la altitud y posicionamiento GPS, ya que el paquete *ardrone_autonomy* no cuenta con el formato de los tópicos requeridos para la fusión de datos en ROS. Los tópicos que finalmente utiliza la estrategia son *ardrone/altitude* y *ardrone/gps*.

El nodo controlador *ardrone_driver*, interpreta los comandos del nodo de la misión *mission_node* mediante los tópicos *cmd_vel*, *ardrone/takeoff*, *ardrone/land* correspondientes con el control de la velocidad de los motores, despegue y aterrizaje del AR Drone.

Nodos	Tópicos Publicados	Tópicos Suscritos
<i>ardrone_driver</i>	<i>ardrone/imu</i> <i>ardrone/odometry</i> <i>ardrone/altitude</i> <i>ardrone/gps</i> <i>ardrone/front/image_raw</i>	<i>cmd_vel</i> <i>ardrone/takeoff</i> <i>ardrone/land</i>
<i>kf_localization</i>	<i>odometry/filtered</i> <i>tf</i>	<i>ardrone/odometry</i> <i>ardrone/imu</i> <i>ardrone/altitude</i> <i>tf</i> <i>tf_static</i>
<i>gps_transform</i>	<i>odometry/gps</i> <i>tf_static</i>	<i>ardrone/gps</i> <i>ardrone/imu</i> <i>odometry/filtered</i> <i>tf_static</i>

mission_node		ardrone/front/image_raw w cmd_vel
--------------	--	---

Tabla 5. Nodos ROS en la optimización del posicionamiento

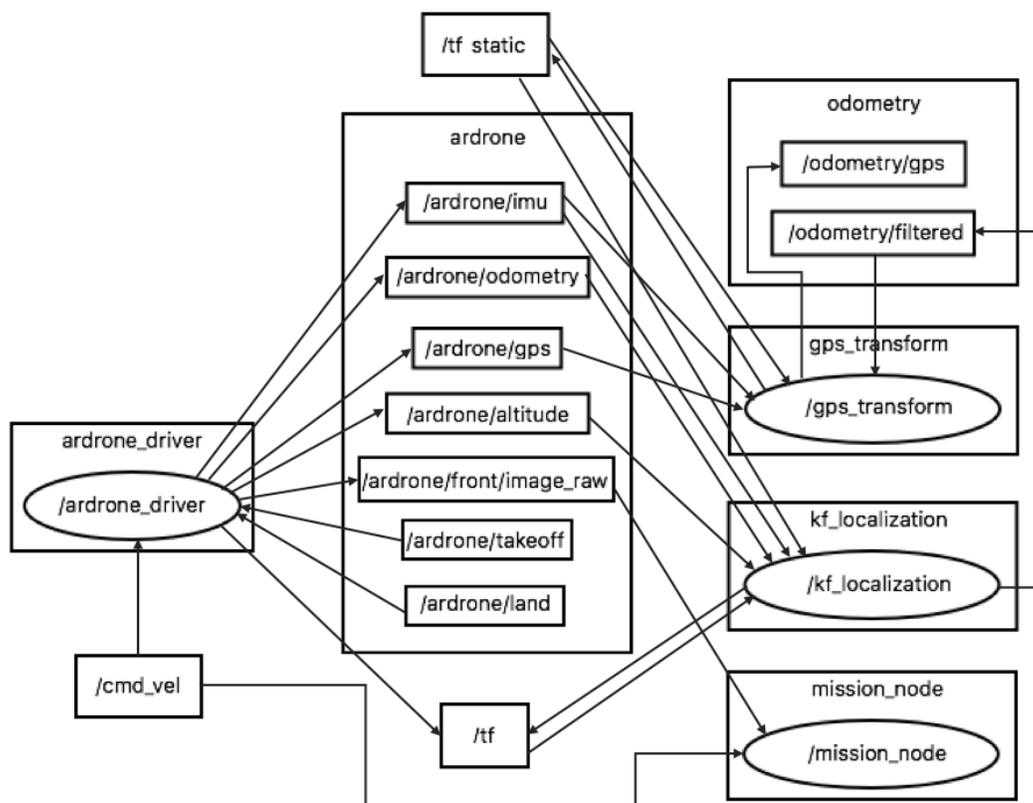


Figura 27. Grafo en ROS para estrategia de posicionamiento

El nodo de localización **kf_localization**, para posicionamiento basado en los algoritmos de Kalman, se suscribe a los tópicos del controlador **ardrone/odometry**, **ardrone/imu** y **ardrone/altitude** para calcular el primer ciclo del filtro de Kalman, y de esta manera obtener odometría filtrada, en el proceso fusión de datos es necesario realizar transformaciones entre los sistemas de referencia local (**odom**) y del cuerpo (**ardrone_base_link**), las cuales se suscribe y publican en los tópicos **tf_static** y **tf**. Para continuar la corrección del posicionamiento con el filtro de Kalman, se adiciona, la odometría obtenida desde el GPS **gps_transform**. Para lo cual se requiere transformar las coordenadas al sistema **tf_static** (sistema de referencia local) mediante los tópicos **ardrone/gps**, **ardrone/imu** y el resultado del primer ciclo de filtrado.

El nodo **mission_node**, integra una estrategia para capturar fotografías automáticamente y un mecanismo para ejecutar el modo autopiloto con GPS del AR Drone. La captura de fotografía se lleva a cabo mediante el nodo estándar de ROS **extract_images**, que hace parte del paquete ROS **image_view**, el cual se suscribe

al t3pico que contiene los mensajes de la c3mara frontal ardrone/front/image_raw. El modo autopiloto del AR Drone, no se encuentra implementado oficialmente como alternativa de navegaci3n, sin embargo, se implementa un nodo para gestionar el servicio en modo auto del paquete *ardrone_autonomy*. De esta manera, se llama secuencialmente el servicio *ardrone/setautoflight* y el servicio *ardrone/setgpstarget* para seguir la ruta de cobertura planificada [110].

5.1.2.2 Implementaci3n de filtros de Kalman

Con el objetivo de seleccionar la mejor estrategia de posicionamiento, el proyecto experimenta con dos filtros de Kalman extendidos mayormente difundidos en la literatura para modelamientos no lineales, FKE y FKU, como se describe en el cap3tulo 3. Los algoritmos implementados en el nodo (*kf_localization*) requieren el conocimiento de premisas como: la matriz de covarianza del error inicial P_0 , las matrices de covarianzas del ruido de la medida y del ruido de proceso Q y R , y la estrategia para calcular puntos sigma en el filtro de Kalman "Unscented":

1. Matriz de covarianza del error inicial

Los par3metros de la matriz de covarianza del error de proceso inicial est3n relacionados directamente con el Jacobiano de la funci3n de transferencia del vector de estado inicial, esta matriz cambia a medida que se actualiza la medida en el filtro.

$$P_0 = \text{diag}(J_f(x, y, z, \varphi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\varphi}, \dot{\theta}, \dot{\psi}, \ddot{x}, \ddot{y}, \ddot{z})) \quad (5.1.1)$$

La funci3n de transferencia F_K , es representada mediante la siguiente matriz:

	x	y	z	φ	θ	ψ	\dot{x}	\dot{y}	\dot{z}	$\dot{\varphi}$	$\dot{\theta}$	$\dot{\psi}$	\ddot{x}	\ddot{y}	\ddot{z}
x	1	0	0	0	0	0	a	b	c	0	0	0	$a/2$	$b/2$	$c/2$
y	0	1	0	0	0	0	d	e	f	0	0	0	$d/2$	$e/2$	$f/2$
z	0	0	1	0	0	0	g	h	i	0	0	0	$g/2$	$h/2$	$i/2$
φ	0	0	0	1	0	0	0	0	0	a	b	c	0	0	0
θ	0	0	0	0	1	0	0	0	0	d	e	f	0	0	0
ψ	0	0	0	0	0	1	0	0	0	g	h	i	0	0	0
\dot{x}	a	d	g	0	0	0	1	0	0	0	0	0	0.01	0	0
\dot{y}	b	e	h	0	0	0	0	1	0	0	0	0	0	0.01	0
\dot{z}	c	f	i	0	0	0	0	0	1	0	0	0	0	0	0.01
$\dot{\varphi}$	0	0	0	a	d	g	0	0	0	1	0	0	0	0	0
$\dot{\theta}$	0	0	0	b	e	h	0	0	0	0	1	0	0	0	0
$\dot{\psi}$	0	0	0	c	f	i	0	0	0	0	0	1	0	0	0
\ddot{x}	$a/2$	$d/2$	$g/2$	0	0	0	0.01	0	0	0	0	0	1	0	0
\ddot{y}	$b/2$	$e/2$	$h/2$	0	0	0	0	0.01	0	0	0	0	0	1	0
\ddot{z}	$c/2$	$f/2$	$i/2$	0	0	0	0	0	0.01	0	0	0	0	0	1

Donde los $a, b, c, d, e, f, g, h, i$ se describen en las siguientes expresiones:

$$\begin{aligned}
a &= \cos \psi \cos \theta \\
b &= \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi \\
c &= \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\
d &= \sin \psi \cos \theta \\
e &= \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi \\
f &= \sin \psi \sin \theta \cos \varphi + \cos \psi \sin \varphi \\
g &= -\sin \theta \\
h &= \cos \theta \sin \varphi \\
i &= \cos \theta \cos \varphi
\end{aligned}$$

2. Matrices de covarianza Q y R

Los parámetros de ruido para las matrices Q y R se seleccionaron con base en observaciones del rendimiento de los sensores en un estado estacionario. Para Q se fijaron los valores de acuerdo a la salida esperada de la IMU y la odometría calculado propia del AR Drone. Para R se consideró además el comportamiento en estado estacionario del receptor GPS y el sensor de ultrasonido, modificando los valores hasta encontrar el mejor rendimiento del filtro. Los parámetros seleccionados toman las desviaciones estándar de cada una de las variables de estado como se determina en las ecuaciones 5.1.2 y 5.1.3

$$Q = \text{diag} \left(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_\varphi^2, \sigma_\theta^2, \sigma_\psi^2, \sigma_\dot{\varphi}^2, \sigma_\dot{\theta}^2, \sigma_\dot{\psi}^2, \sigma_{\dot{x}}^2, \sigma_{\dot{y}}^2, \sigma_{\dot{z}}^2, \sigma_{\ddot{x}}^2, \sigma_{\ddot{y}}^2, \sigma_{\ddot{z}}^2 \right) \quad (5.1.2)$$

$$R = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2) \quad (5.1.3)$$

3. Cálculo de los puntos sigma en FKU

Considerando que la matriz de covarianza del error, P, es positiva, se define un factor de corrección llamado *Peso de Covarianza* considerando la descomposición Cholesky (LL') [111], específicamente las columnas de la matriz L. Este factor es adicionado y restado al estado predicho para formar un conjunto de 2N (siendo N = 15 variables de estado) puntos sigma. Dichos puntos sigma integran en su cálculo la media y la covarianza del sistema.

De acuerdo a lo anterior los puntos sigma para la etapa de predicción del filtro FKU se obtienen con la ecuación 5.1.4.

$$\begin{aligned}
\hat{x}_k^{(i)}_{FKU} &= F_k * (\hat{x}_{k-1} + \text{Peso de Covarianza}) \\
\hat{x}_k^{(i+N)}_{FKU} &= F_k * (\hat{x}_{k-1} - \text{Peso de Covarianza})
\end{aligned} \quad (5.1.4)$$

Y los puntos sigma para la etapa de corrección del filtro FKU se obtienen con la ecuación 5.1.5.

$$\begin{aligned}\hat{x}_{k \text{ FKU}}^{(i)} &= (\hat{x}_{k-1} + \text{Peso de Covarianza}) \\ \hat{x}_{k \text{ FKU}}^{(i+N)} &= (\hat{x}_{k-1} - \text{Peso de Covarianza})\end{aligned}\tag{5.1.5}$$

5.1.3 Planificación de Cobertura

La estación de tierra dispuesta para el proyecto es un ordenador portátil que corre el sistema operativo ROS soportado en Ubuntu 14.04 LTS. Fundamentada en ROS, se desarrolla un software que le permita al piloto encargado de la misión de vuelo cargar el mapa que incluye el área de estudio, calcular el mejor plan de vuelo según el número de puntos de paso definidos y variables externas, y por último realizar un protocolo de preparación de vuelo.

5.1.3.1 Drone Planner

El software denominado **DronePlanner**, está desarrollado con base en la plataforma QT Creator versión 4.2, en el lenguaje C++ y la metodología de desarrollo está basada en los estándares de ROS [112]. La interfaz gráfica está diseñada para pilotos con conocimientos en el manejo de ROS e imágenes georreferenciadas.

El código fuente del software comprende tres funcionalidades principales que siguen el orden de la figura 28. El paquete en ROS de **DronePlanner** que implementa las funcionalidades utiliza las librerías OpenCV (Open Source Computer Vision Library) y GDAL (Geospatial Data Abstraction Library) para el procesamiento del mapa digital en el cual el AR Drone desplegará la misión.

The image displays three screenshots from the DronePlanner software interface, illustrating its main functionalities:

- 1 Procesamiento del Mapa Digital (Raster):** Shows the 'Raster' window with input fields for 'No. Cells' (60), 'Start Position' (Manual/Auto), 'Position' (100), and 'Wind Direction' (North). A grid of numbered cells is overlaid on a satellite map.
- 2 Calcular rutas de cobertura (Planner):** Shows the 'Planner' window with a path overlaid on the map. A summary box indicates: 'Selected planner: _dijkstra.tif Start Position: 0', 'Flight Alt: 7m', 'Size Cell: 10.72x12.85 m', and 'SpF: 0.424'.
- 3 Verificar funcionamiento de AR Drone (Connect):** Shows the 'Connect' window with drone status data: 'Lat0: 0.00000000', 'Long0: 0.00000000', 'Alti0: -393.21', 'Batt: 82.00%', 'nSat: 0', and 'Press: 81654 Pa'.

Figura 28. Funcionalidades principales del código fuente desarrolladas

El *procesamiento del mapa* requiere que el usuario de **DronePlanner** pueda acceder al mapa digital georreferenciado (formato TIFF) de la zona que se desea cubrir y la posición inicial.

La posición inicial puede ser elegida manualmente o calculada por **DronePlanner**. Para el caso de selección manual, la interfaz muestra el índice identificador de las celdas en el mapa, para que el usuario escriba el índice en el campo **position** de la interfaz. Para el caso de la selección automática, se requiere especificar la dirección del viento en el campo de la interfaz **wind direction** (del norte, del sur, del este o del oeste) y con la convención especificada en sección 4.6.2 se declara una posición inicial en alguna esquina del mapa. Con las entradas del planificador fijadas, el primer paso es tomar la imagen en el procesamiento de la imagen es descomponerla en celdas (cantidad menor a 100) para obtener el número de nodos del grafo, descubriendo a su vez los nodos correspondientes a las zonas de poco interés (obstáculos) mediante la obtención del costo (valores de píxeles de las bandas roja, verde y azul) de cada celda, de acuerdo al pseudocódigo 5.

Después de preparar el mapa se procede a *calcular rutas de cobertura* considerando las celdas libres identificadas en la etapa anterior. La acción de esta funcionalidad está dada por el botón **Planner**, con el cual se calcula los tres algoritmos de cobertura basados en Dijkstra, Espirales y Frente de Onda, para luego visualizar el más eficiente. Cada algoritmo de planificación inicia con la creación del grafo de acuerdo a la configuración de vecinos Vonon Neuman o Moore, asignando a cada nodo (celda libre) sus vecinos respectivos. El siguiente paso es buscar nodos libres desde la posición inicial de acuerdo a la heurística de cada algoritmo, como lo indica el pseudocódigo 6. Cuando las rutas han sido calculadas, se pasa como parámetros a un procedimiento de evaluación (método *seleccionar_cobertura*), para determinar cuál es la mejor ruta. La selección de la ruta de cobertura se realiza por puntajes, dándole un punto a la ruta con menor distancia recorrida, un punto a la ruta con menos giros y un punto a la ruta con menos revisitas. En el caso que no haya cobertura total de alguna ruta, se le disminuye el puntaje en 1, y en el caso que hay puntuaciones iguales se decide por la ruta con menor distancia a recorrer. Este paso obtiene no solo la ruta de cobertura en el plano, sino también se calcula la altura optima de vuelo y el tiempo de captura, de acuerdo a la sección 4.6.1 del capítulo planeación de vuelo para *sensado remoto*.

Seguidamente, se toman las rutas calculadas y la imagen georreferenciada para obtener la posición geográfica de cada una de las celdas visitadas por la ruta en el sistema global WGS84 o el código EPSG 4326. Para realizar esta tarea se extraen las coordenadas geográficas de las esquinas y el centro de la imagen, con lo cual se realiza una transformación de cada pixel a coordenadas geográficas mediante interpolación lineal (método *Geotransformación*). El último paso es escribir un archivo de texto con las coordenadas geográficas para cada ruta y un archivo con la altura calculada (pseudocódigo 6), en un formato diseñado para ser legible en el software del sistema de procesamiento en vuelo, Raspberry Pi.

Pseudocódigo 5: Procesamiento del Mapa

Identificación de Celdas (Imagen I, Número de celdas requeridas m):

num_col: número de columnas de la imagen en pixeles

num_fil: número de filas de la imagen en pixeles

$tamaño_celda \leftarrow 5 * (num_col / m)$

$media_celda \leftarrow tamaño_celda / 2$

$x \leftarrow media_celda$

$y \leftarrow media_celda$

$num_nodos \leftarrow 0$

Inicio ciclo

Si $(x + media_celda \leq num_fil) \ \& \ (y + media_celda \leq num_col)$

$Nodos \leftarrow x * num_col + y$

$Costo \leftarrow costo_promedio(x, y)$

$x \leftarrow x + tamaño_celda$

$y \leftarrow y + tamaño_celda$

$num_nodos \leftarrow num_nodos + 1$

Sino
Fin ciclo
Retornar (num_nodos)

costo_promedio (Posición del pixel p_x, Posición del pixel p_y):

promedio $\leftarrow 0$
 cont_fil $\leftarrow p_x - media_celda$
 cont_col $\leftarrow p_y - media_celda$
Inicio ciclo
Si (cont_fil $\leq p_x + media_celda$) & (cont_col $\leq p_y + media_celda$)
 promedio $\leftarrow promedio + Intensidad_pixel(cont_fil, cont_col)$
Sino
Fin ciclo
Retornar (promedio/(tamaño_celda*tamaño_celda))

Pseudocódigo 6: Rutas de Cobertura

planificador_ruta (Imagen i, Origen s, Número de nodos n):

num_col: número de columnas de la imagen en pixeles
 num_fil: número de filas de la imagen en pixeles

col_descomposicion $\leftarrow num_col/tamaño_celda$
 nodo $\leftarrow 0$
Inicio ciclo
Si (nodo $\leq n$)
 vecino_derecha $\leftarrow nodo + 1$
 vecino_izquierda $\leftarrow nodo - 1$
 vecino_arriba $\leftarrow nodo - col_descomposicion$
 vecino_abajo $\leftarrow nodo + col_descomposicion$
 vecino_superior_izq $\leftarrow vecino_arriba - 1$
 vecino_superior_der $\leftarrow vecino_arriba + 1$
 vecino_inferior_izq $\leftarrow vecino_abajo - 1$
 vecino_inferior_der $\leftarrow vecino_abajo + 1$

 Si (vecino es parte del mapa)
 G $\leftarrow ady[nodo].push(vecino)$
Sino
Fin ciclo

Ruta_dijkstra \leftarrow Cobertura_Dijkstra (Grafo G, Origen s)
 Ruta_FrenteOnda \leftarrow Cobertura_FrenteOnda (Grafo G, Origen s)
 Ruta_Espiral \leftarrow Cobertura_Espiral (Grafo G, Origen s)

Ruta \leftarrow seleccionar_cobertura(Ruta_dijkstra, Ruta_FrenteOnda, Ruta_Espiral)
 Coordenadas_ruta \leftarrow Geotransformación(Ruta)

Crear y escribir archivo de texto con **Coordenadas_ruta (Longitud, Latitud)**
 Crear y escribir archivo de texto con **Altura optima**

seleccionar_cobertura (Ruta a, Ruta b, Ruta c):

$Puntuación_frenteonda \leftarrow 0$

$Puntuación_espiral \leftarrow 0$

$Puntuación_dijkstra \leftarrow 0$

Si (distancia_frenteonda es la menor distancia)

$Puntuación_frenteonda \leftarrow Puntuación_frenteonda + 1$

Sino Si (distancia_espiral es la menor distancia)

$Puntuación_espiral \leftarrow Puntuación_espiral + 1$

Sino Si (distancia_dijkstra es la menor distancia)

$Puntuación_dijkstra \leftarrow Puntuación_dijkstra + 1$

Si (giros_frenteonda es el menor número)

$Puntuación_frenteonda \leftarrow Puntuación_frenteonda + 1$

Sino Si (giros_espiral es el menor número)

$Puntuación_espiral \leftarrow Puntuación_espiral + 1$

Sino Si (giros_dijkstra es el menor número)

$Puntuación_dijkstra \leftarrow Puntuación_dijkstra + 1$

Si (revisitas_frenteonda es el menor número)

$Puntuación_frenteonda \leftarrow Puntuación_frenteonda + 1$

Sino Si (revisitas_espiral es el menor número)

$Puntuación_espiral \leftarrow Puntuación_espiral + 1$

Sino Si (revisitas_dijkstra es el menor número)

$Puntuación_dijkstra \leftarrow Puntuación_dijkstra + 1$

Si (nodosvisitados_frenteonda/nodos_libres < 1)

$Puntuación_frenteonda \leftarrow Puntuación_frenteonda - 1$

Sino Si (nodosvisitados_espiral/nodos_libres < 1)

$Puntuación_espiral \leftarrow Puntuación_espiral - 1$

Sino Si (nodosvisitados_dijkstra/nodos_libres < 1)

$Puntuación_dijkstra \leftarrow Puntuación_dijkstra - 1$

Si (las puntuaciones son puntuaciones son iguales)

Retornar ruta con menor distancia

Sino Si (Puntuación_frenteonda >= 2)

Retornar ruta b

Sino Si (Puntuación_espiral >= 2)

Retornar ruta c

Sino Si (Puntuación_dijkstra >= 2)

Retornar ruta a

Geotransformación (Imagen *i*, Ruta *r*):

Abrir imagen con Raster TIFF con GDAL

Obtener las coordenadas geográficas de las cuatro esquinas y el centro de la imagen

nodos_ruta $\leftarrow 0$

Inicio ciclo

Si (*nodos_ruta* \geq *numero de nodos en r*)

x \leftarrow *posición de nodo_ruta en la imagen*

y \leftarrow *posición de nodo_ruta en la imagen*

coordenadas_ruta \leftarrow *pixel_a_mundo*(cinco puntos de referencia,

x, y)

nodos_ruta \leftarrow *nodos_ruta* + 1

Sino

Fin ciclo

Retornar (*coordenadas_ruta*)

Con la ruta de cobertura calculada, es necesario *verificar el estado funcional* del cuadricóptero antes de ejecutar la misión, con el fin de seguir protocolos de seguridad. Para este propósito, se inicia el controlador ROS *ardrone_autonomy* y se conecta con la interfaz gráfica mediante el botón **Connect**, para obtener en detalle parámetros como:

- La posición geográfica de partida (Lat0, Lon0): La longitud y latitud inicial que registra el GPS da una idea de la cercanía con punto de partida seleccionado.
- La altitud inicial (Alti0): este parámetro es calculado con los sensores de ultrasonido, de presión y con la elevación registrada por el GPS; y debería ser cercano cero.
- El porcentaje de batería (Batt): con la batería se determina la autonomía del *drone*, por lo cual se establece que, para un sensado remoto con una carga útil como la Raspberry Pi, el porcentaje de batería debe ser superior 85%.
- El número de satélites que gestiona el GPS (nSat): para lograr una mejor precisión en el posicionamiento se debe contar con la recepción de al menos 10 satélites.
- La presión atmosférica percibida por el AR Drone (Press): la presión atmosférica es un parámetro de seguimiento de los vuelos planeados, que regularmente no excede 84000 Pascales.

Adicionalmente hay un **panel de control del AR Drone** en la parte final de la interfaz el cual envía comandos como despegar, aterrizar, evaluar respuesta del cuadricóptero en cada eje de movimiento y reiniciar la aeronave.

5.2 Plan de pruebas

En este apartado se presenta la programación de pruebas y las métricas que validan el rendimiento del sistema de *sensado remoto* implementado. Las pruebas realizadas

están enmarcadas en dos escenarios exteriores, en los cuales se despliega la estrategia de posicionamiento, planificación de cobertura y ejecución de misión de *sensado remoto*.

5.2.3 Escenarios de pruebas

Los escenarios de pruebas se seleccionaron considerando las condiciones reales de un cultivo agrícola de Colombia. Los dos escenarios a convenir se caracterizan por ser campos verdes extensos y despejados de edificaciones.

- **Escenario 1: Campus Deportivo Universidad del Cauca – CDU**

Ubicación: Latitud 2.4473757970566234 N, Longitud 76.59928804265127 O

Vientos locales: Menores a 1.2 m/s

Número de Satélites disponibles: 10

Pruebas: posicionamiento de AR Drone en ambientes abiertos



Figura 29. Escenario 1- CDU Universidad del Cauca - Popayán

- **Escenario 2: Cultivo de Pasto Gramillón – Norte de Popayán**

Ubicación: Latitud 2.4473757970566234 N, Longitud 76.59928804265127 O

Vientos locales: 1.2 m/s – 2.0 m/s

Número de Satélites disponibles: 11

Pruebas: algoritmos de cobertura y sensado remoto de cultivo



Figura 30. Escenario 2- Cultivo de pasto gramillón – Vereda Guacas, Popayán

5.2.4 Pruebas

Las pruebas para validar el funcionamiento y rendimiento del prototipo implementado están basadas en los principales aportes de la investigación: posicionamiento de AR Drone en espacios abiertos, rutas cobertura completa mediante algoritmos heurísticos y planeación de vuelo para *sensado remoto* de cultivos. La tabla 6 describe en detalle las pruebas programadas para el prototipo.

Modulo a Evaluar	Referencia	Escenario	Descripción de la Prueba
Posicionamiento en ambientes abiertos	P1.1	1	Pruebas generales para determinar los parámetros de configuración de los filtros de Kalman.
	P1.2	1	Comparación del comportamiento de los filtros de Kalman FKE y FKU para seleccionar el algoritmo con mejor rendimiento en exteriores
	P1.3	1	Evaluación de precisión del filtro de Kalman seleccionado para una ruta rectangular y una circular
	P2.1	1	Comparación del gasto energético de la plataforma en cada una de las posibles posiciones de partida

Rutas de cobertura completa	P2.2	2	Evaluación del rendimiento y porcentaje de cobertura de cada algoritmo en un ambiente real
	P2.3	2	Comparación de Mosaicos obtenidos con la determinación del software de planificación de rutas

Tabla 6. Plan de Pruebas

5.2.3 Métricas de Evaluación

La evaluación para el sistema de *sensado remoto* propuesto se basa en los dos módulos de desarrollo identificados en la tabla 6, los cuales corresponden al prototipo implementado:

Evaluación de posicionamiento del AR Drone en ambientes abiertos

La métrica de evaluación para el posicionamiento es el valor RECM, es decir, la raíz cuadrada el error cuadrático medio (ecuación 5.2.1), mediante el cual se establece la aproximación de los puntos de paso de las rutas experimentales y_i con respecto a los puntos de paso de las rutas esperadas \hat{y}_i , con n puntos de paso.

$$RECM = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (5.2.1)$$

Las rutas esperadas corresponden a recorrer rutas predefinidas que fijan un rectángulo y un círculo. Para la ruta rectangular se mide la precisión de las posiciones de cada lado del rectángulo y para la ruta circular se mide la precisión de cada posición de la ruta considerando la distancia horizontal.

Evaluación de la cobertura de las rutas calculadas con algoritmos heurísticos

Las métricas de evaluación de cobertura están basadas en parámetros determinantes en el consumo de batería:

- *Numero de giros*: una ruta de cobertura óptima debe tener la menor cantidad de giros ya que el cuadricóptero requiere mayor potencia para realizarlos.
- *Numero de revisitas*: una ruta de cobertura óptima debe tener la menor cantidad de revisitas ya que el cuadricóptero gasta más tiempo en retornar a las celdas ya visitadas.

- *Numero de celdas libres cubiertas*: una ruta de cobertura óptima debe cubrir cada celda libre para obtener información completa del área en estudio.
- *Distancia*: una ruta de cobertura óptima debe invertir alrededor de 10 minutos o menos por lo que la distancia a recorrer debe ser la más corta.
- *Mosaico*: evaluación cualitativa de los mosaicos de fotografías obtenidas durante la ejecución de las rutas de cobertura realizadas por el AR Drone.

La métrica principal para la evaluación del *sensado remoto* es el tiempo mínimo requerido para la ejecución de las rutas de cobertura desde las 4 esquinas del mapa a monitorizar mientras la plataforma está sometida a una intensidad de viento no superior a 2.5 m/s.

5.3 Resultados y Análisis

De acuerdo al plan de pruebas de la tabla 6 en la sesión 5.2.2, se describen los resultados de la puesta en funcionamiento del *sistema de sensado* desarrollado en el presente proyecto.

5.3.1 P1.1: Configuración de Filtros de Kalman

De acuerdo a la descripción del prototipo las variables incluidas para la fusión sensorial son obtenidas desde los sensores de la IMU, odometría calculada desde el AR Drone y el receptor GPS. Por tanto, en la implementación de los filtros de Kalman se consideró aumentar progresivamente el número de variables de estado y optimizar la frecuencia de actualización, para lograr la estabilidad de los filtros.

El primer paso para la configuración de los filtros de Kalman, es establecer la calibración de los sensores involucrados, ya que, con esto se obtienen los valores iniciales para las matrices Q y R, las cuales son independientes del filtro. Los valores iniciales de los sensores involucrados se obtienen de la observación del estado estacionario del *drone* durante 12 minutos aproximadamente como se observa en las figuras 31, 32 y 33.

Para Q se fijaron los valores de acuerdo a la salida esperada del acelerómetro (0 m/s^2 en el plano y 9.8 m/s^2 para la aceleración en el eje Z) y la salida esperada del giróscopo (0 rad/s alrededor de los tres ejes) en estado estacionario como se presenta en la figura 31. Como complemento se tienen los valores en estado estacionario de la odometría básica de AR Drone (figura 31), de la cual se espera una posición inicial de cero en los tres ejes de traslación (0 m en el plano y 0 m en el eje Z) y velocidades lineales en cero.

Para construir R, además de la odometría de la figura 31, se considera la posición en el plano del receptor GPS como se muestra en la figura 32, la cual se espera

mantenga una precisión alrededor de 2 m (de acuerdo a las especificaciones técnicas del fabricante).

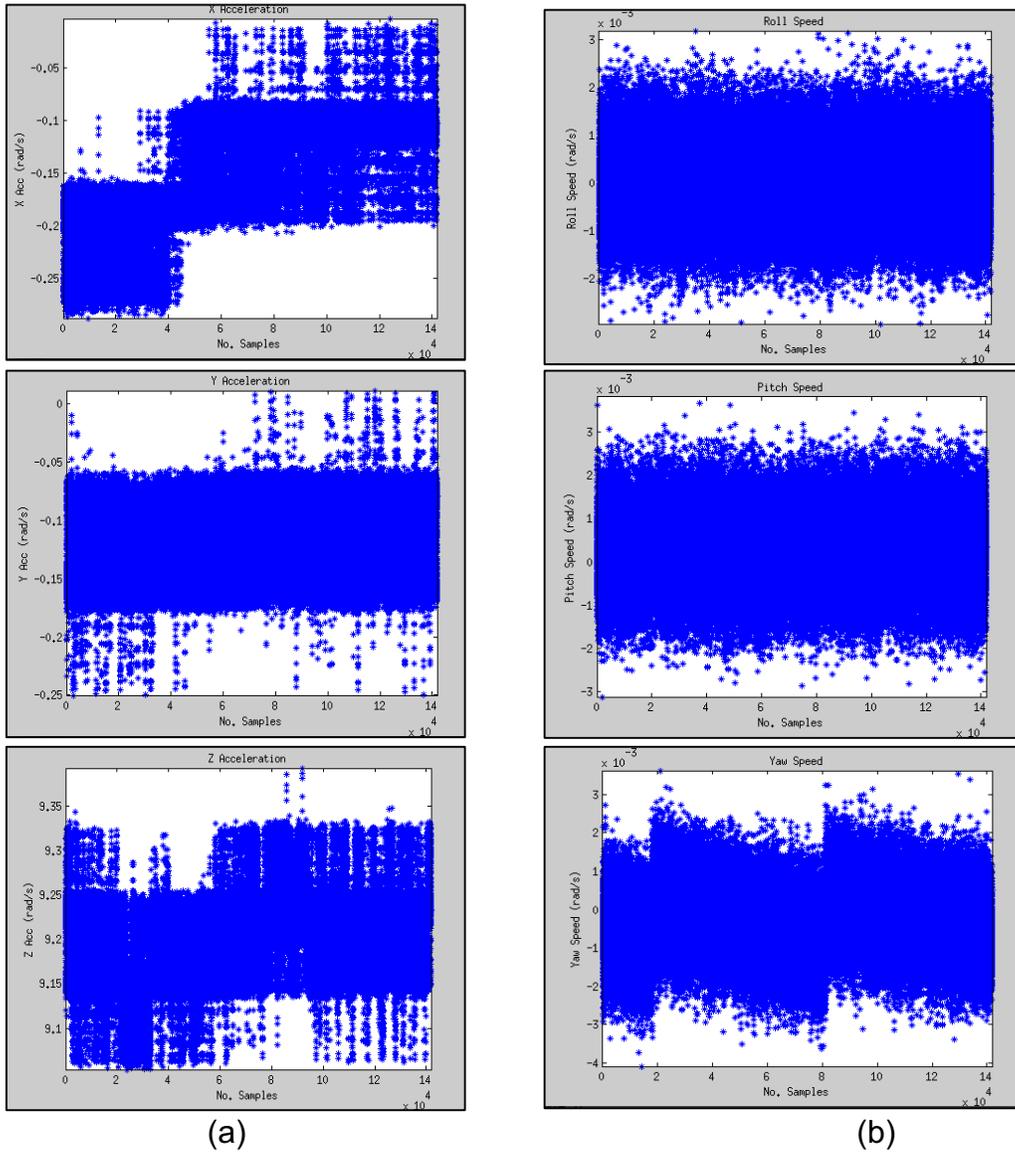


Figura 31. Estado estacionario de la IMU (a) Mediciones del acelerómetro y (b) Mediciones del giróscopo

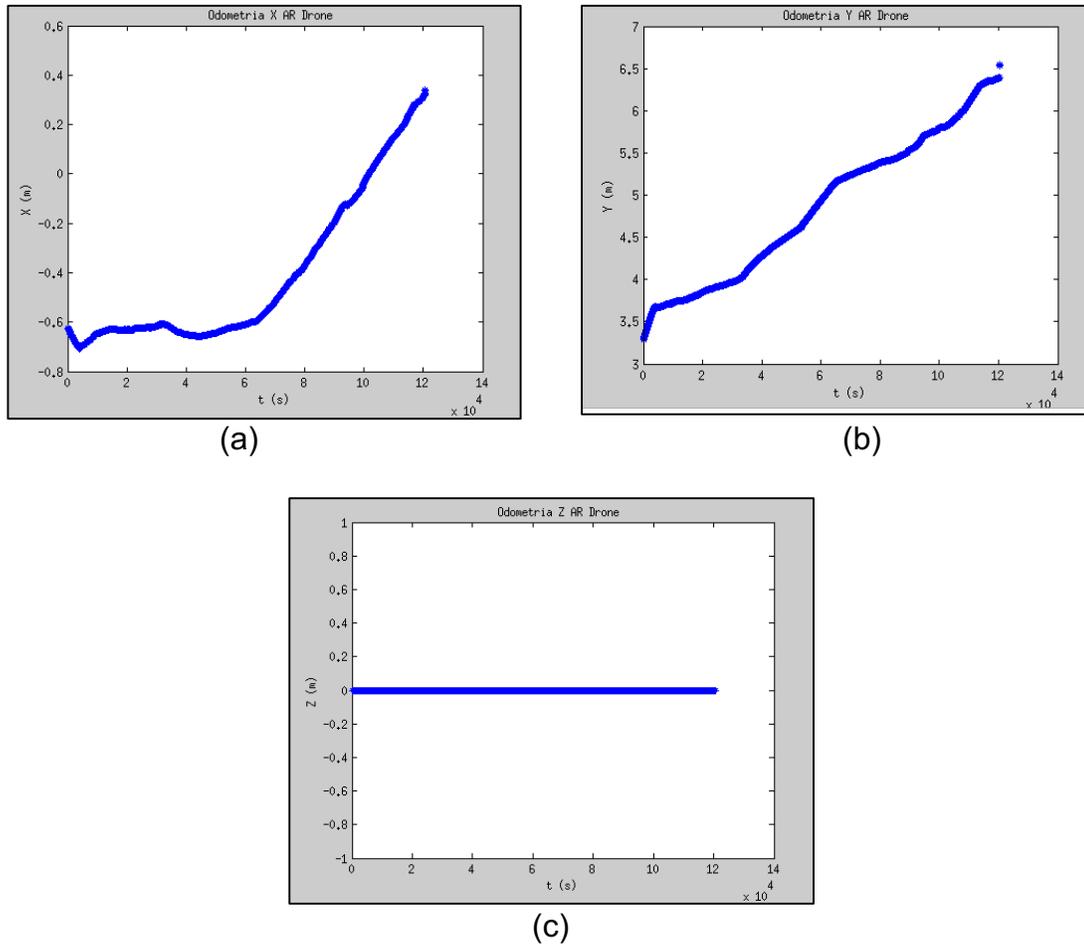


Figura 32. Estado estacionario de la odometría básica de AR Drone (a) Posición x, (b) Posición y (c) Posición z

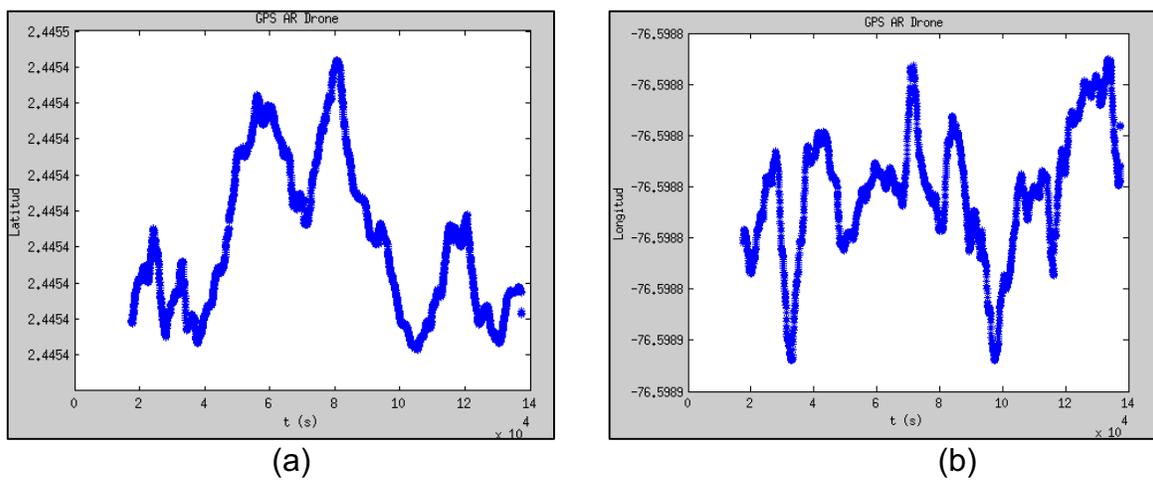


Figura 33. Estado estacionario del receptor GPS (a) Latitud y (b) Longitud

De acuerdo a los resultados anteriores los valores de los parámetros que conforman las matrices Q y R inicialmente se observan en las tablas 7 y 8.

σ_x^2	σ_y^2	σ_z^2
0.05	0.05	0.06
σ_φ^2	σ_θ^2	σ_ψ^2
0.03	0.03	0.06
$\sigma_{\dot{x}}^2$	$\sigma_{\dot{y}}^2$	$\sigma_{\dot{z}}^2$
0.06	0.025	0.025
$\sigma_{\dot{\theta}}^2$	$\sigma_{\dot{\theta}}^2$	$\sigma_{\dot{\psi}}^2$
0.0001	0.0005	0.0005
$\sigma_{\ddot{x}}^2$	$\sigma_{\ddot{y}}^2$	$\sigma_{\ddot{z}}^2$
0.01	0.01	0.1

Tabla 7. Covarianzas asociadas a Q

σ_x^2	σ_y^2	σ_z^2
0.01	0.01	0.00001

Tabla 8. Covarianzas asociadas a R

Partiendo de las matrices Q y R desde el estado estacionario del *drone*, se inicia un proceso de optimización de manera empírica aumentando de manera progresiva las variables de estado involucradas desde cada fuente de información (tabla 9).

	x	y	z	φ	θ	ψ	\dot{x}	\dot{y}	\dot{z}	$\dot{\varphi}$	$\dot{\theta}$	$\dot{\psi}$	\ddot{x}	\ddot{y}	\ddot{z}
IMU	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
Odometría	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
GPS	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 9. Variables de estado de cada fuente de información

Para lograr estabilizar los filtros con un RECM aceptable se inicia la fusión de datos entre IMU y odometría básica, y la segunda fusión considera IMU, odometría y GPS. Para las pruebas de las fusiones se tuvo en cuenta solo la ruta del plano, ya que, entre los filtros FKE y FKU tienen un comportamiento similar. Los resultados de las fusiones nombradas se muestran en las figuras 35 y 36; y son comparados con la odometría básica en la figura 34 (las figuras solo muestran los resultados con FKE). La tabla 10, resume los errores RECM encontrados en cada fusión con respecto a la ruta esperada la cual constituye un cuadrado de 5 metros de ancho por 4 metros de longitud y la conclusión reflejada en los cambios de las matrices Q y R.

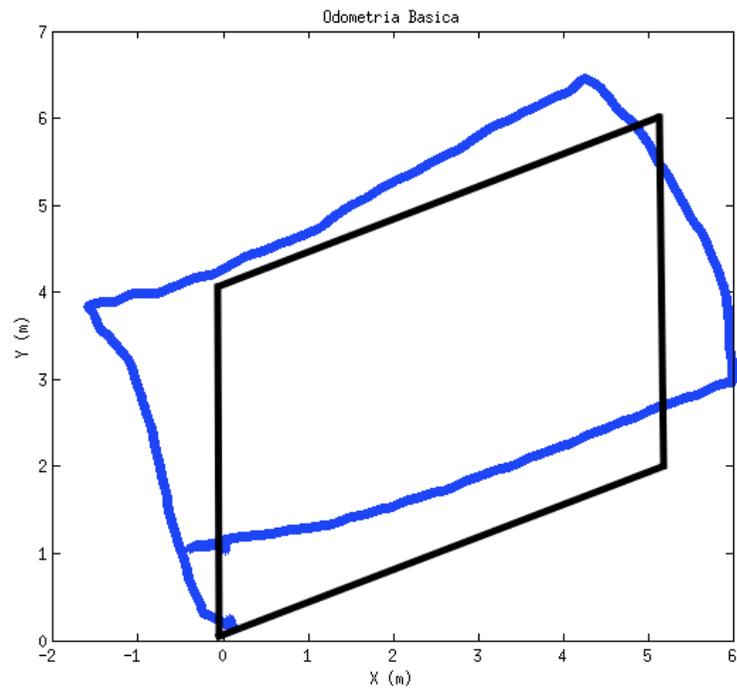


Figura 34. Odometría Básica (azul) y ruta esperada (negro)

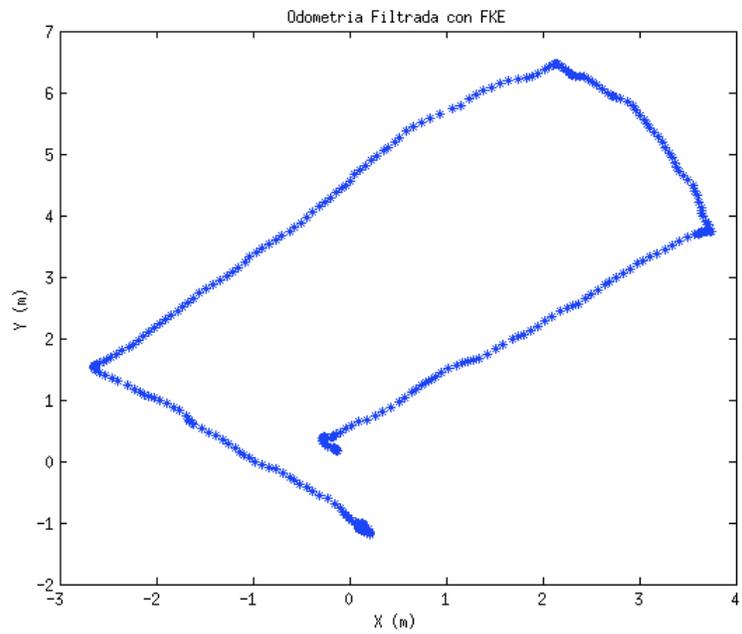


Figura 35. Fusión IMU y Odometría

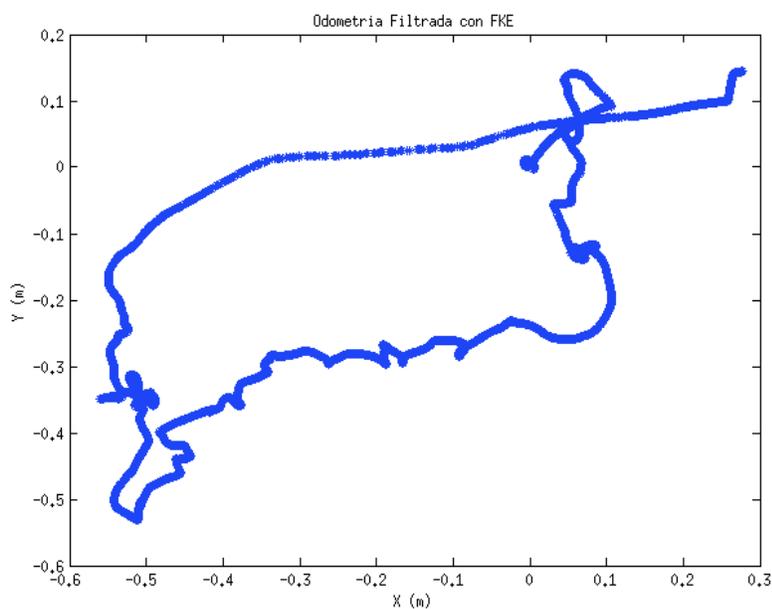


Figura 36. Fusión completa con GPS

	RECM (m)	Cambios en Q	Cambios en R
Odometría	1.23	$\sigma_y = 0.01$ $\sigma_{\dot{y}} = 0.01$	$\sigma_y = 0.05$
Fusión 1	1.75	$\sigma_y = 0.02$ $\sigma_{\ddot{x}} = 0.01$	$\sigma_x = 0.05$
Fusión 2	0.52	$\sigma_y = 0.02$ $\sigma_z = 0.03$	$\sigma_x = 0.05$ $\sigma_y = 0.03$

Tabla 10. Precisión de los filtros y cambios en las matrices de ruido

Como conclusiones relacionadas con esta sección de pruebas, se establecen las desviaciones estándar finales para conformar las matrices Q y R como se muestran en las tablas 11 y 12, para una frecuencia de actualización de 50 Hz. Además, es necesario destacar, algunos hallazgos de la experimentación expuesta en las figuras. El primero, es que, al agregar el GPS a la fusión, la odometría obtenida arroja valores de posición diferentes a las otras odometrías. Esta particularidad fue corregida, al notar que los ejes de traslación en el sistema local del AR Drone y del GPS tenían sentidos opuestos. El segundo hallazgo, se puede observar en la figura 36, donde se destaca que a pesar de que la ruta presenta derivas y algunos retrocesos, su orientación es similar a la ruta esperada y su error es cercano a 0.52 metros, lo cual representa una optimización destacada con respecto a la primera fusión. En cuanto a la primera fusión, se puede observar que no añade mayores cambios a la odometría básica del AR Drone, por el contrario, no optimiza el posicionamiento, sino que tiende a aumentar la desviación estándar.

σ_x^2	σ_y^2	σ_z^2
0.05	0.0004	0.0009
σ_φ^2	σ_θ^2	σ_ψ^2
0.03	0.03	0.06
$\sigma_{\dot{x}}^2$	$\sigma_{\dot{y}}^2$	$\sigma_{\dot{z}}^2$
0.06	0.0001	0.025
$\sigma_{\ddot{\theta}}^2$	$\sigma_{\ddot{\theta}}^2$	$\sigma_{\ddot{\psi}}^2$
0.0001	0.0005	0.0005
$\sigma_{\ddot{x}_i}^2$	$\sigma_{\ddot{y}}^2$	$\sigma_{\ddot{z}}^2$
0.01	0.01	0.1

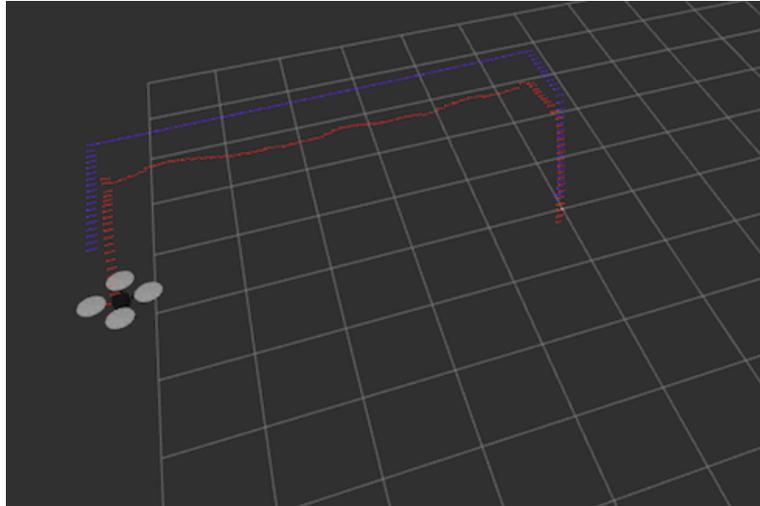
Tabla 11. Covarianzas asociadas a Q

σ_x^2	σ_y^2	σ_z^2
0.0025	0.0009	0.00001

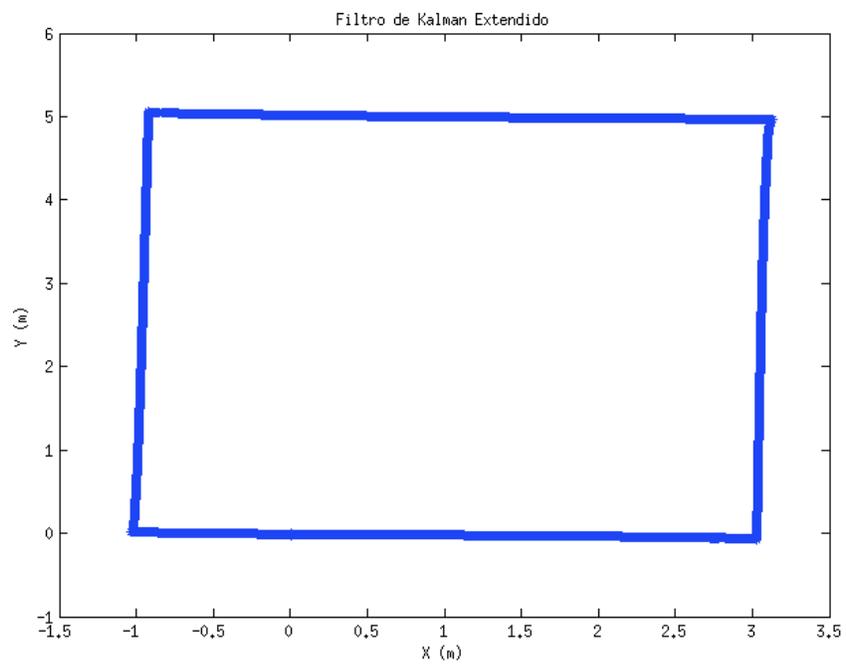
Tabla 12. Covarianzas asociadas a R

5.3.2 P1.2: Comparación de filtros FKE y FKU

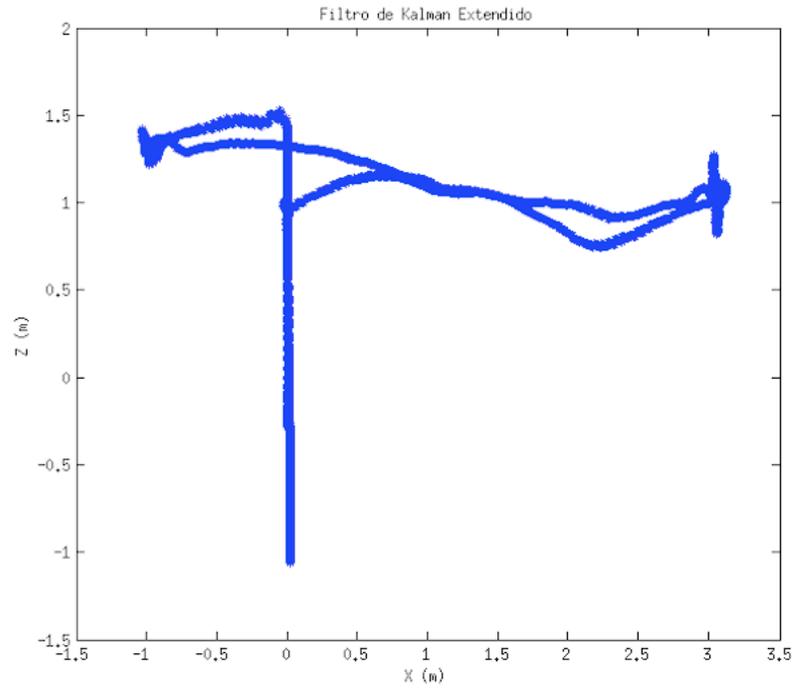
La figura 32 y 33 representan el comportamiento del filtro FKE y FKU en simulación para una ruta rectangular con 4 metros de ancho, 5 metros de largo y altura de 1.5 metros. Las figuras muestran la odometría filtrada. En la simulación (figuras 32(a) y 33(b)) se observa una diferencia de 0,3 m en altura con respecto a la ruta planeada (color azul), puesto que el filtro asume como sistema de referencia el suelo, y el modelo de simulación considera como referencia el centro de gravedad de la aeronave. En este experimento de simulación de rutas, se observa una maniobra ideal en el plano (figuras 32(b) y 33(b)) para ambos filtros con un error RECM similar de 0.020 metros con respecto a la ruta planeada; sin embargo, las figuras 32(c) y 33(c) evidencian una gran diferencia, ya que con el filtro FKE se obtiene una respuesta estable y con un error RECM apreciable de 0.37 metros, mientras que con el filtro FKU se obtiene una respuesta en altura inestable con saltos de información frecuentes y dispersos.



(a)

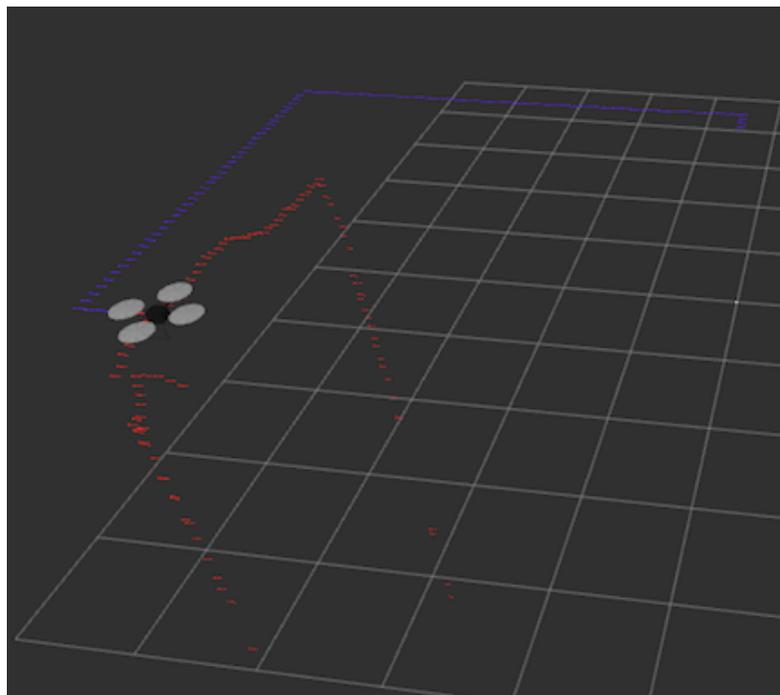


(b)

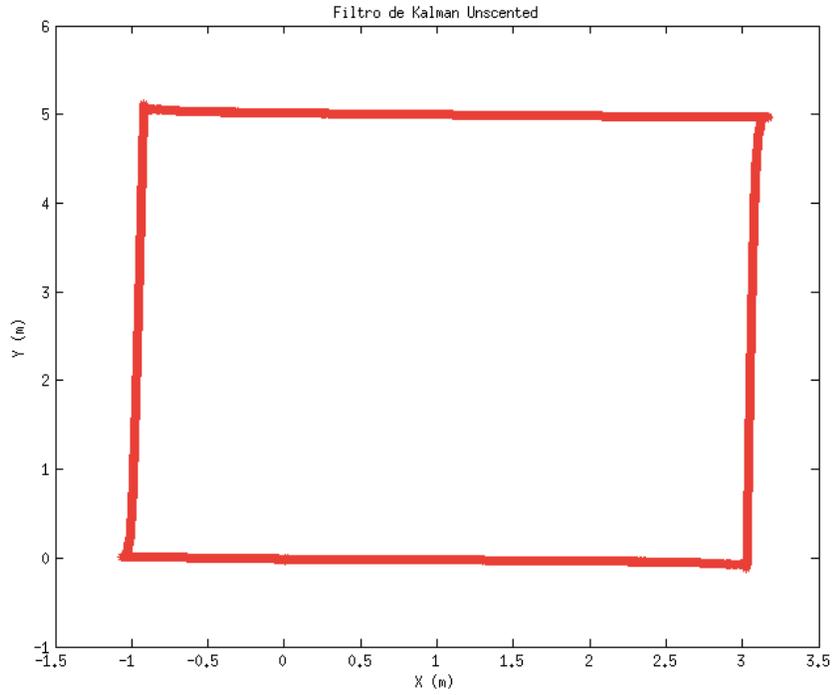


(c)

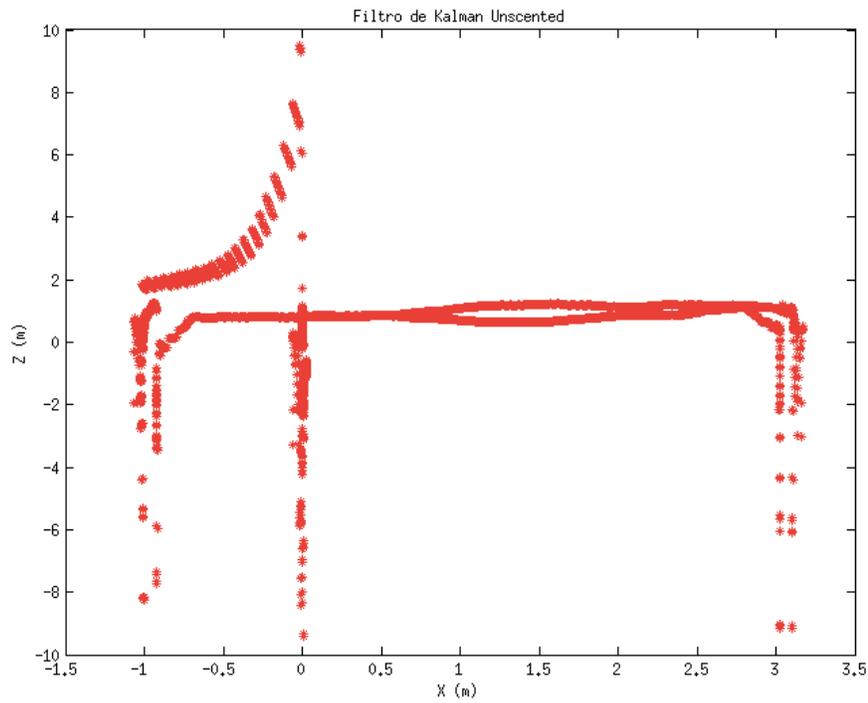
Figura 37. Simulación de una ruta rectangular con FKE (a) vista en Rviz, (b) ruta en el plano y (c) ruta en altura



(a)



(b)



(c)

Figura 38. Simulación de una ruta rectangular con FKU (a) vista en Rviz, (b) ruta en el plano y (c) ruta en altura

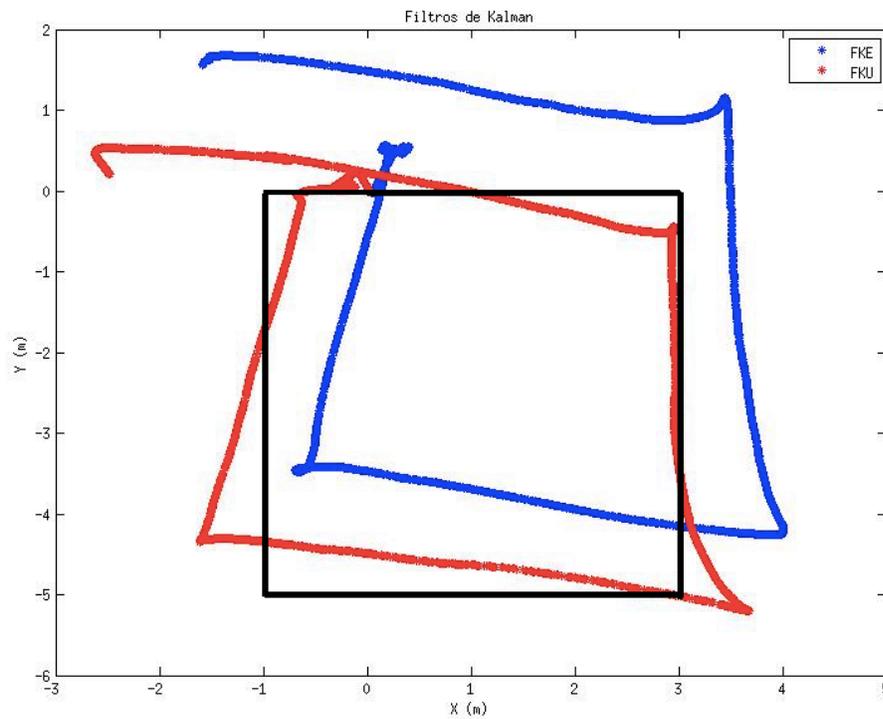


Figura 39. Ruta rectangular experimental FKE y FKU en el plano

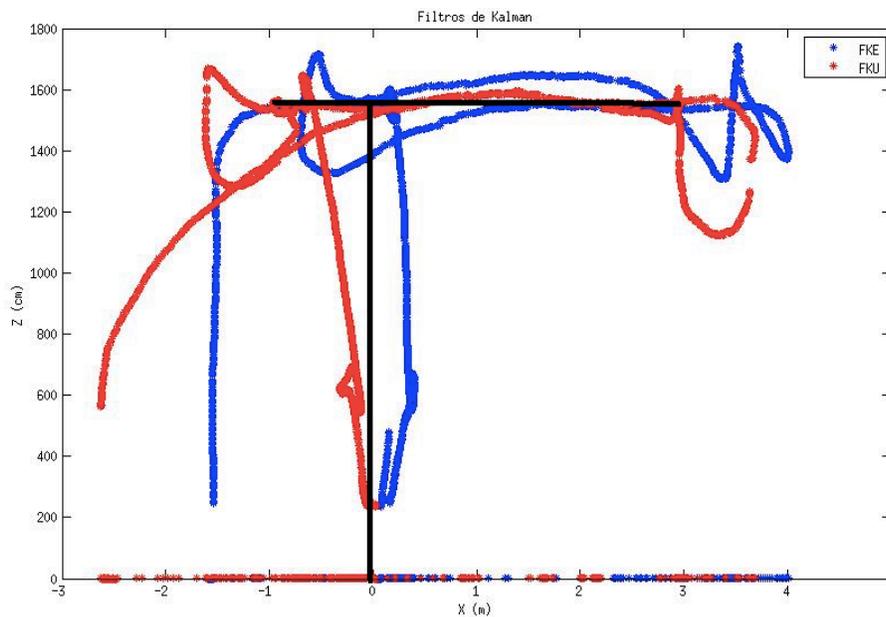


Figura 40. Ruta rectangular experimental FKE y FKU en altura

De acuerdo a las observaciones en simulación, se llevan a cabo pruebas que sustenten la selección definitiva del filtro de Kalman extendido. Las figuras 34 y 35

representan los resultados reales con un viento de 1.0 m/s de la ejecución de la ruta rectangular propuesta en simulación.

Los resultados evidencian que el filtro FKU (ruta de color rojo) tiene una mayor aproximación a la ruta esperada en el plano, con un RECM correspondiente al promedio del error de cada lado de la ruta rectangular de 0.1514 metros. Mientras que la ruta filtrada por FKE (ruta de color azul) arroja un error RECM de 0.6233 metros.

Consecuente con la ruta en el plano, se asume la respuesta de los filtros en el eje Z (figura 35), la cual evidencia un error RECM de 0.2864 metros para FKE y de 0.6415 metros para FKU, definiendo de esta manera, que la corrección del filtro FKE es más aproximada al comportamiento esperado en vuelo.

De acuerdo a los resultados anteriores se infiere una comparativa entre los filtros de Kalman, ya que si bien, FKU tiene un nivel de aproximación superior a FKE en el plano, no se obtiene el mismo rendimiento en la altura de vuelo, evidenciando saltos de información importantes, mayor inestabilidad que en la corrección del FKE, y una acumulación rápida y significativa del error, lo cual incurre en una mayor desviación con respecto a la ruta esperada.

De acuerdo a las observaciones anteriores, y considerando que para realizar una misión de *sensado remoto* es indispensable conservar una altura estable, se determina que el filtro FKE sea la estrategia óptima para posicionamiento del AR Drone en ambientes abiertos como los cultivos.

5.3.2 P1.3: Posicionamiento con FKE

De acuerdo al criterio de selección mencionado anteriormente, se realizó una serie de experimentos, mediante los cuales se acondicionaron las matrices de covarianza del filtro FKE para lograr optimizar la ruta y alcanzar un RECM en el plano mínimo de 0.5420 metros con respecto a la ruta con la odometría real (figura 36) y en altura (figura 37) un error mínimo de 0.1031 metros con respecto a la altura real.

Las figuras 36 y 37 representan una ruta rectangular de 2 metros de ancho, 3 metros de largo y 1.5 metros de altura. En ellas se aprecian los resultados de la ruta real y filtrada en el plano, fijando la ruta corregida con el filtro FKE, como la ruta óptima tanto en el plano como en altura. Alternativamente se considera desplegar el cuadricóptero para seguir una ruta circular (figura 38), con la cual se busca discriminar el RECM y la capacidad del filtro para incluir cambios de dirección en la corrección de la ruta. La ruta circular tiene un radio aproximado de 1.5 metros y conserva un RECM de 0.1690 metros con respecto a la ruta esperada.

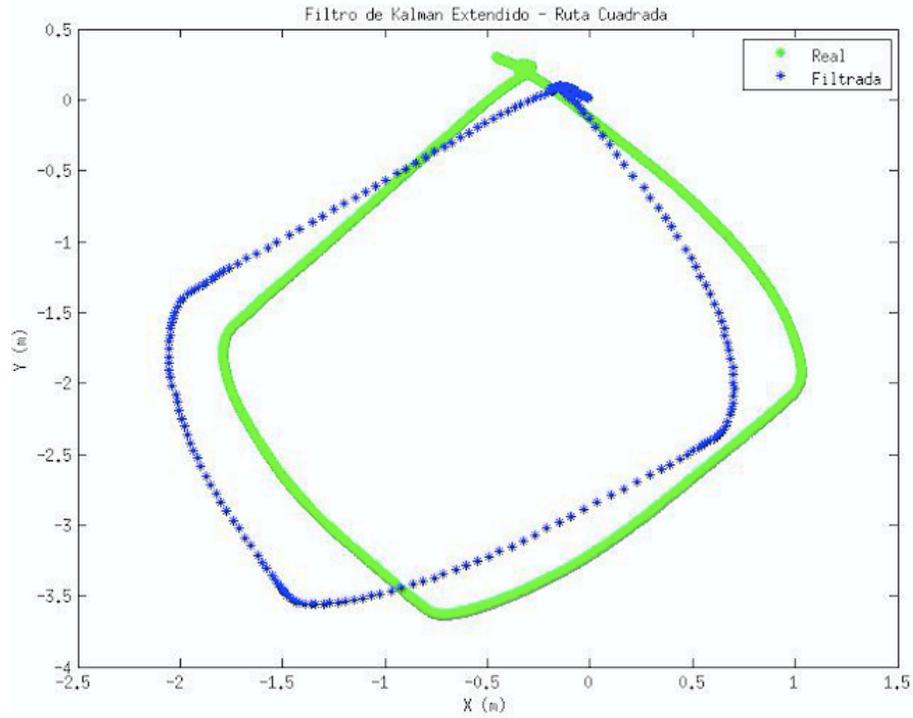


Figura 41. Ruta rectangular en el plano con FKE optimizado

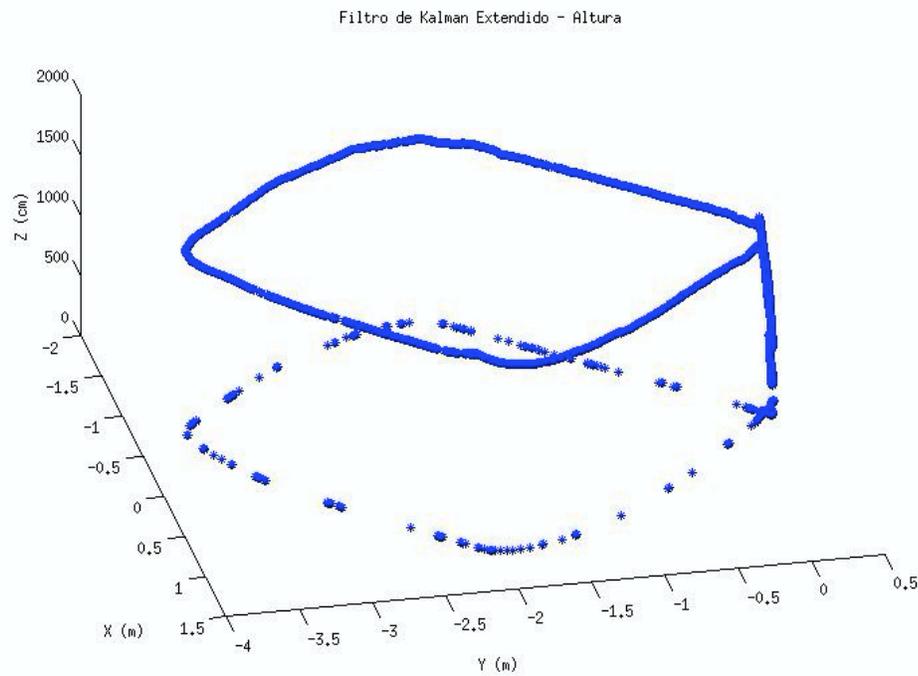


Figura 42. Ruta rectangular en altura con FKE optimizado

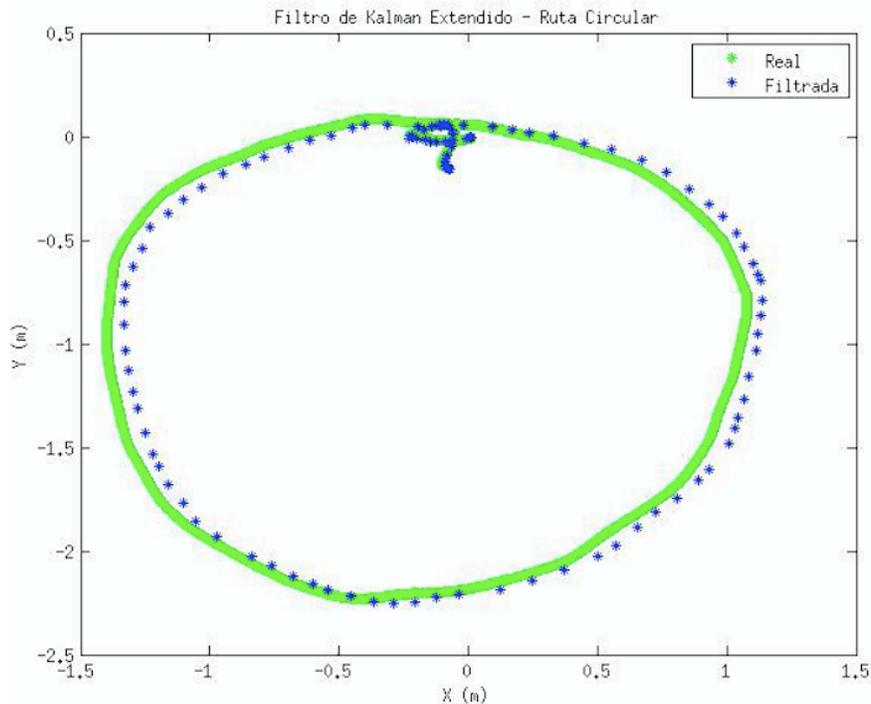


Figura 43. Ruta circular en el plano con FKE optimizado

De acuerdo a los resultados anteriores el filtro de Kalman FKE conserva una precisión en el plano de 0.54 metros aproximadamente con respecto a las rutas esperadas, el cual es un valor menor que la precisión especificada para el receptor GPS de 2.0 metros.

5.3.3 P2.1: Planeación de Vuelo

Uno de los retos del prototipo implementado es gestionar de manera eficiente la batería del cuadricóptero AR Drone, para este propósito se plantea una estrategia para maniobrar el *drone* ante la intensidad y dirección del viento local. La estrategia se basa en las siguientes pruebas.

- *Maniobra frente al viento:* para establecer cuál es la mejor estrategia de maniobra del vehículo aéreo no tripulado en ambientes expuestos a vientos locales, se programó una ruta en línea recta de aproximadamente 5 metros. El *drone* realizó la ruta programada con el viento a favor, en contra y trasversal al cuerpo, con un viento de 2.0 a 2.5 m/s.

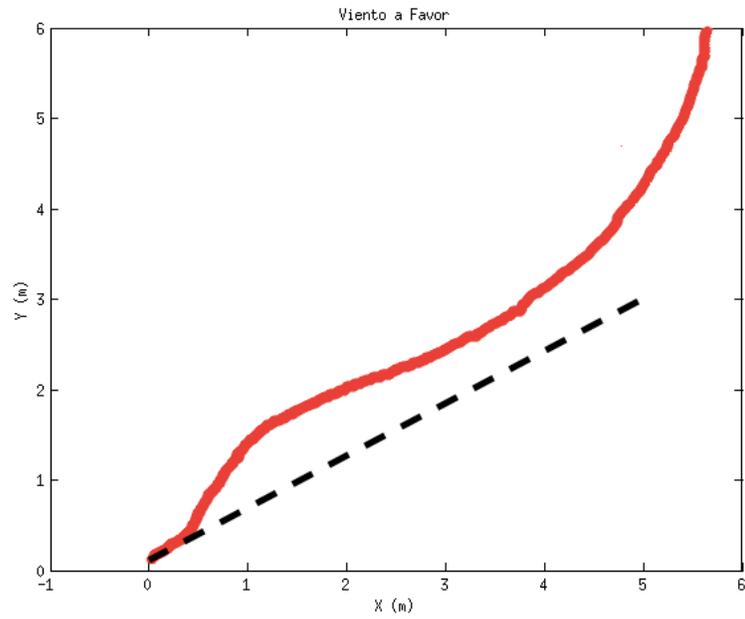


Figura 44. Deriva de AR Drone vs. Viento a Favor

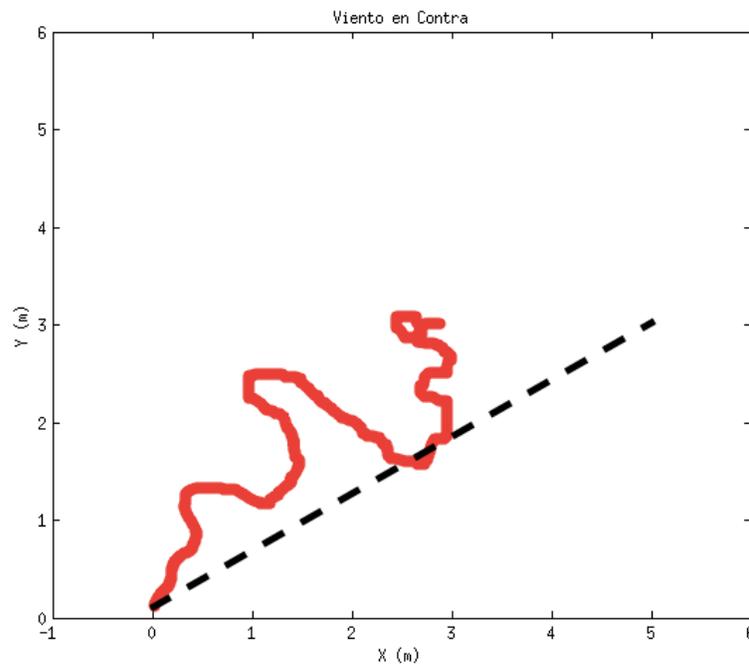


Figura 45. Deriva de AR Drone vs. Viento en Contra

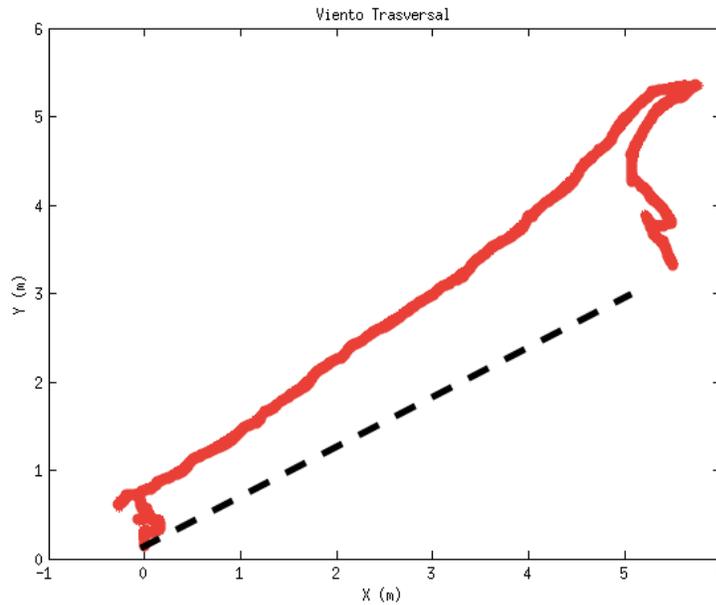


Figura 46. Deriva de AR Drone vs. Viento Trasversal

Las figuras 44, 45 y 46 evidencian el esfuerzo energético que debe hacer el robot aéreo maniobrando con viento en contra y la deriva maniobrando con el viento a favor. Por lo que se concluye que la maniobra que resulta eficiente, es cuando el AR Drone atraviesa la dirección del viento. Este resultado es de esperar, ya que, si se representa la velocidad del viento y la velocidad del AR Drone de acuerdo a la física de vectores (figura 47), un viento en contra requerirá mayor potencia del AR Drone para avanzar en la ruta, por lo que en la figura 45 se observa que el *drone* es frenado por el viento y no termina la ruta programada. Un viento a favor otorgará una cantidad de movimiento adicional que puede finalizar con la deriva de la plataforma, razón por la cual, en la figura 44 el *drone* alcanza la ruta esperada pero no se detiene hasta después de un tiempo. Mientras que un viento transversal, imprime en la plataforma una deriva aceptable en la precisión del posicionamiento.

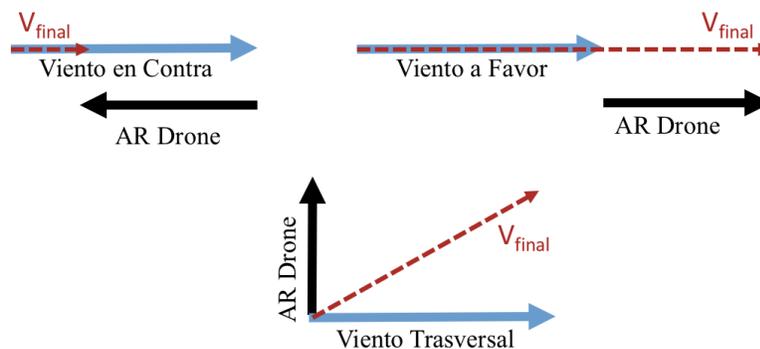


Figura 47. Vectores de Movimiento de AR Drone con respecto al viento

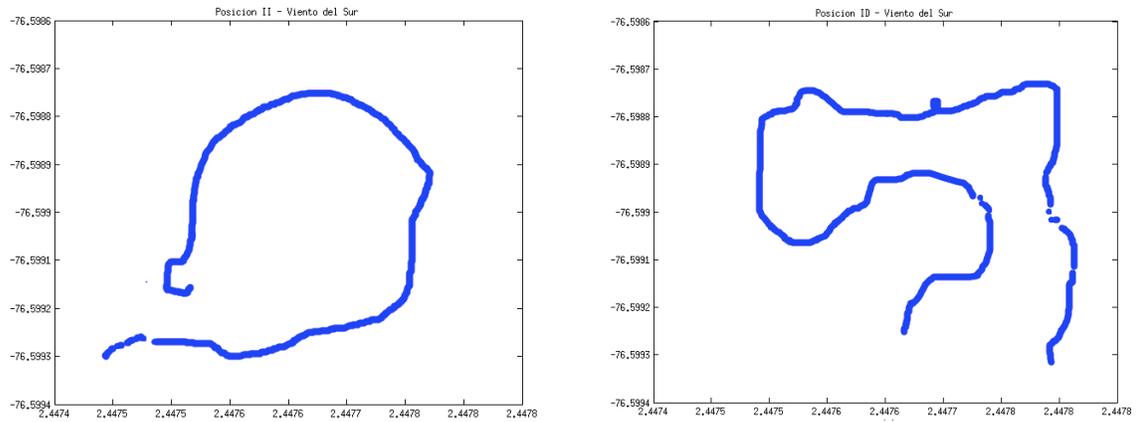


Figura 50. Rutas con viento del Sur

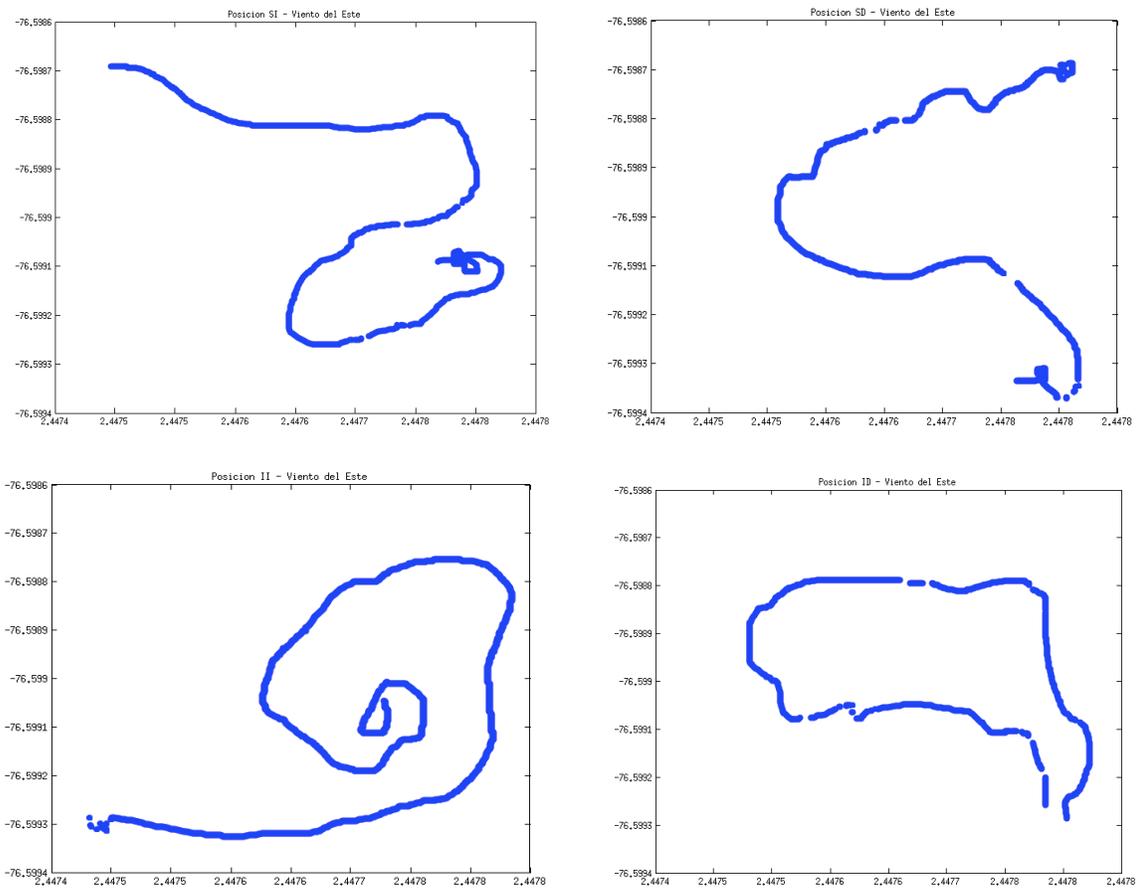


Figura 51. Rutas con viento del Este

Este	SD	7.3'	4.45
	II	7.5'	3.80
	ID	7.5'	6.52
Oeste	SI	7.4'	5.52
	SD	7.4'	3.22
	II	7.4'	6.21
	ID	7.4'	6.16

Tabla 13. Resultados de posición inicial vs. Tiempo de ruta

De acuerdo a la tabla 13, para un viento del Sur o del Oeste la ruta con mayor proximidad es la que inicia con desde la esquina superior derecha, sin embargo, es con respecto al Oeste que el *drone* pasa la mayor parte de tiempo en atravesando el viento. Las rutas con viento del Norte y del Este presentan una mayor cercanía a la ruta esperada, cuando se inicia en SI y II respectivamente. La cercanía con la ruta esperada no implica que se la posición con el mejor consumo energético, sin embargo, las posiciones resaltadas se encuentran en tiempos aceptables en los que se permite sortear una ruta de cobertura heurística. De acuerdo a lo anterior, se establecen la siguiente convención:

- Viento desde el norte: posición de partida SI
- Viento desde el sur: posición de partida ID
- Viento desde el este: posición de partida II
- Viento desde el oeste: posición de partida SD

Esta convención para la posición inicial, fue implementada en el software de planificación **DronePlanner**, para el caso en el que el piloto requiera elegir la posición inicial de manera automática.

5.3.4 P2.2: Rutas de Cobertura con Algoritmos Heurísticos

Las rutas de cobertura calculadas con el software **DronePlanner** para el área especificada en el escenario 2, se observan en la figura 53 y los parámetros calculados para un *sensado remoto* óptimo en la figura 54.



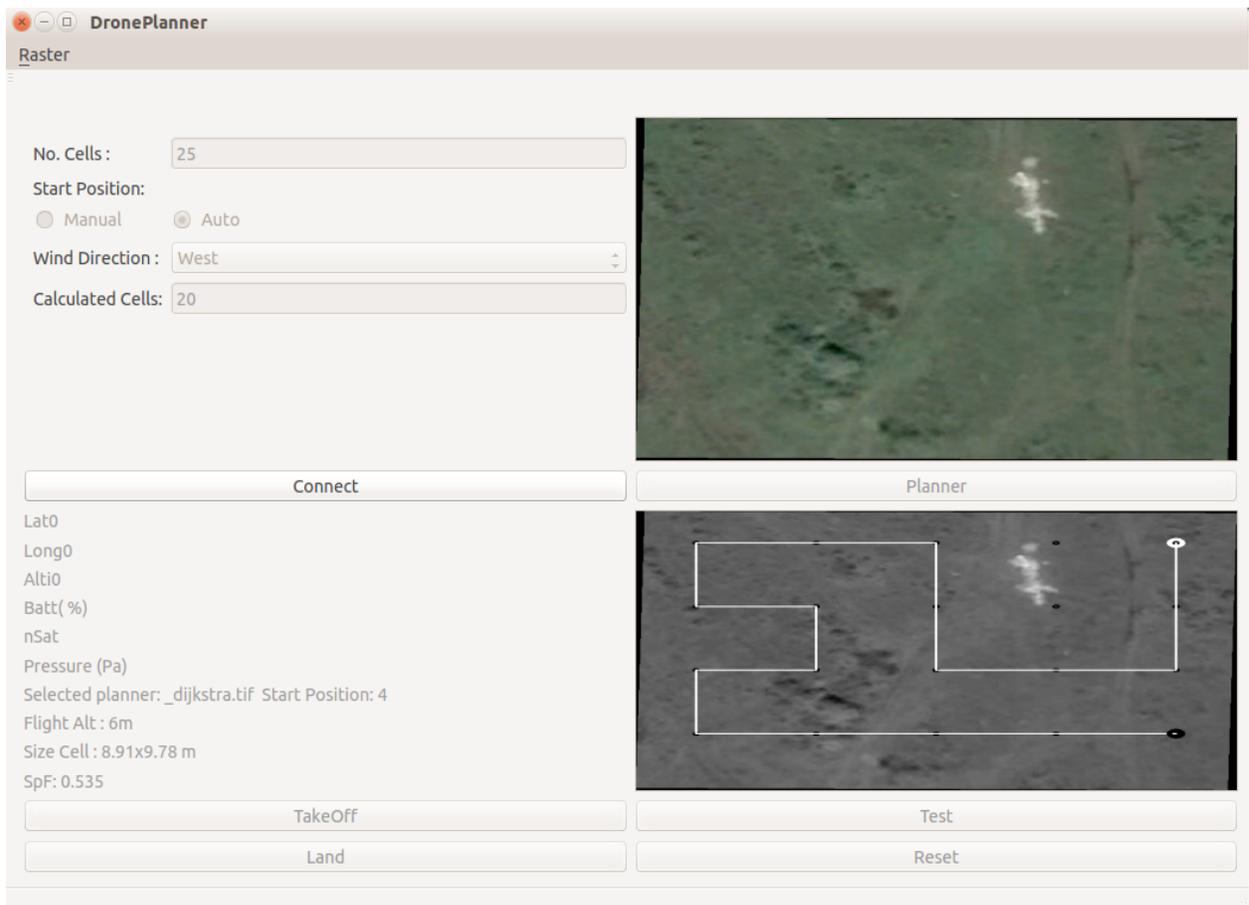


Figura 54. Parámetros resultantes calculados por DronePlanner para sensado remoto

Algoritmo basado en	Giros	Revisitas	Cobertura (%)	Distancia (m)
Frente de Onda	7	2	100	178.099
Espirales	7	1	100	170.614
Dijkstra	10	0	100	151.355

Tabla 14. Parámetros de evaluación de cobertura calculada con DronePlanner

De acuerdo a la figura 54, el software **DronePlanner** descompone el área en 20 celdas con un tamaño 8.91x 9.78 metros cuadrados y establece que la misión de *sensado remoto* se debe realizar:

1. A una altura de 6 metros
2. Con un tiempo de captura de fotografías de 0.535 segundos

3. Recorridos evitando las celdas numeradas como 3 y 8, las cuales representan la zona de poco interés.
4. Iniciar la misión en la celda 4, ya que se establece un viento desde el oeste.

En la tabla 14 se observa que la ruta calculada con el algoritmo basado en Dijkstra es la ruta más corta y con menos revisitas, por lo cual es seleccionada como la ruta de cobertura óptima para el cultivo definido en el escenario 2. Análogamente, la ruta de cobertura calculada con el algoritmo basado en propagación de frente de onda presenta la mayor cantidad de revisitas y la ruta más larga. En general las rutas calculadas aseguran un porcentaje total de la cobertura, es decir, visitan 18 celdas de las posibles 18 celdas libres.

Las rutas calculadas fueron ejecutadas con un viento promedio de 1.3 m/s al oeste, con 10 satélites capturados por el receptor GPS y con un tiempo de vuelo de 6 minutos aproximadamente. En las figuras 55, 56, y 57 se visualiza la comparación de las rutas estimadas con el GPS por defecto, y la ruta de cobertura corregida por el componente de optimización de posición desarrollado en Raspberry Pi.

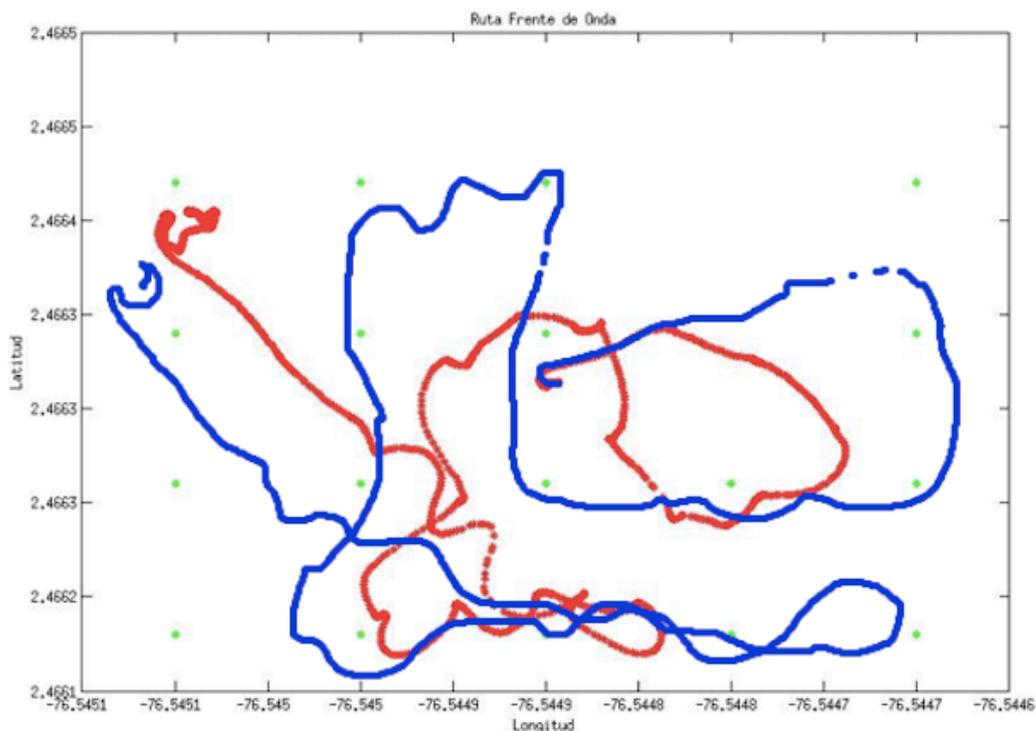


Figura 55. Ruta de cobertura basada en la propagación de frente de onda

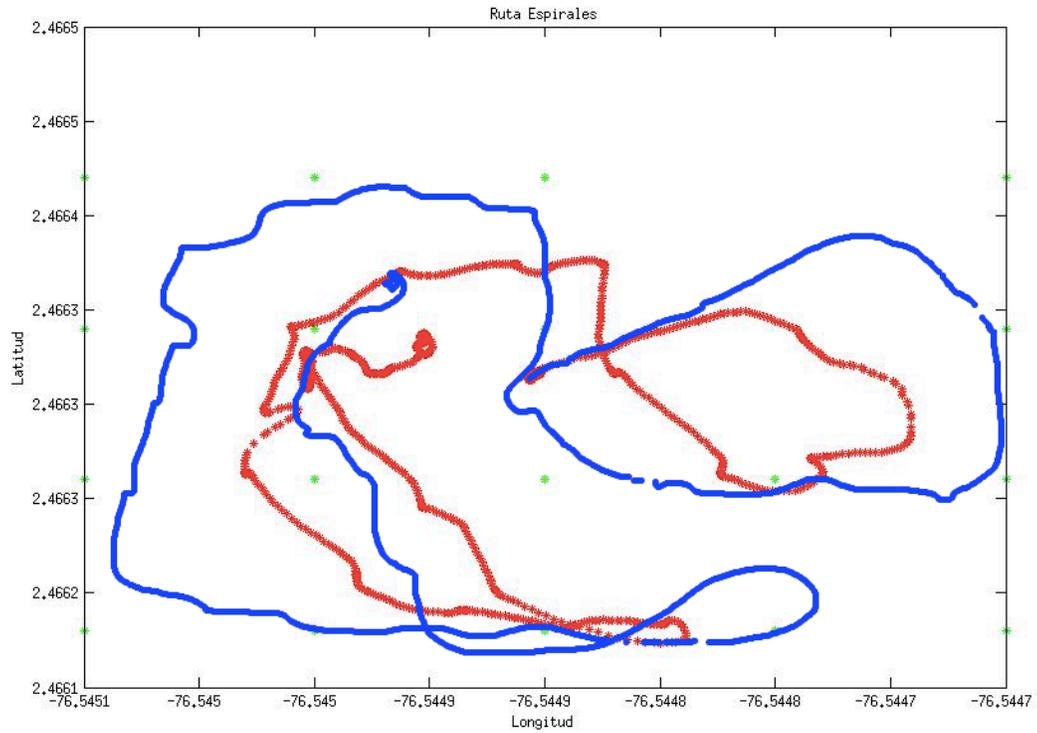


Figura 56. Ruta de cobertura basada en espirales

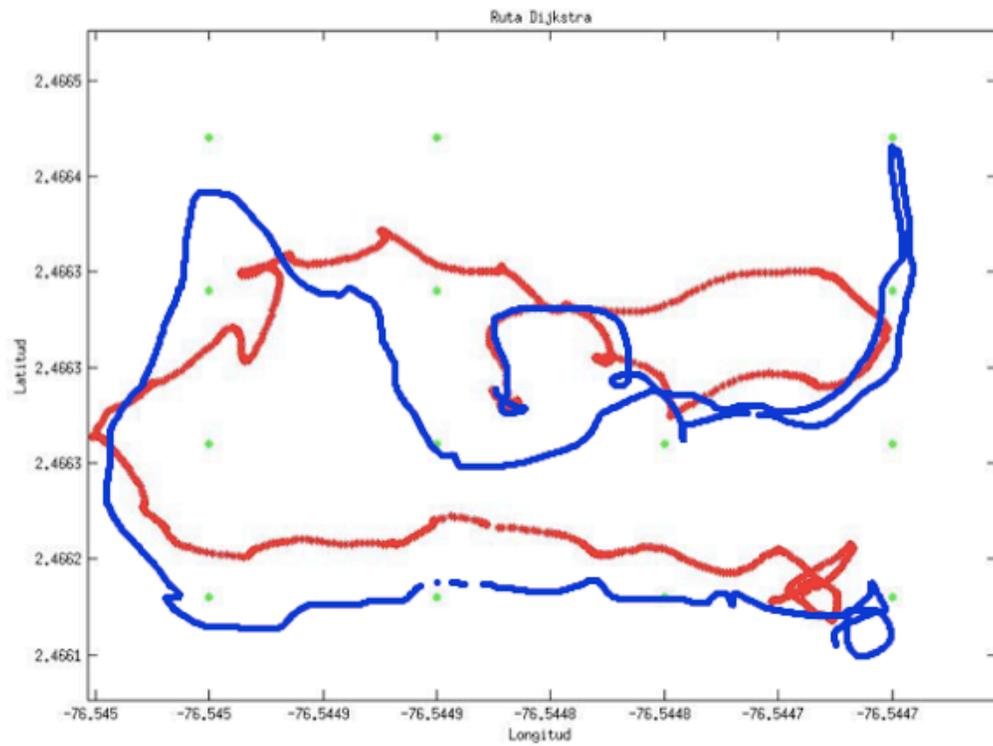


Figura 57. Ruta de cobertura basada en Dijkstra

Algoritmo basado en	Giros	Revisitas	Cobertura (%)	Distancia (m)	RECM
Frente de Onda	11	5	77.8	304.20	1.951
Espirales	15	4	83.3	289.42	1.987
Dijkstra	13	0	72.2	213.89	1.962

Tabla 15. Parámetros de evaluación de las rutas de cobertura real

Según los resultados de las figuras, las rutas reales conservan en gran manera la guía de la ruta calculada por el planificador, sin embargo, todas las rutas presentan ausencia de información en algunos tramos del recorrido, lo cual es un resultado probable de la pérdida de señal con los satélites. Una característica común entre las rutas de cobertura, es que ninguna de ellas cubre totalmente la celda inicial, y en la mayoría se cubre parcialmente las celdas de las esquinas. Además, todas las rutas conservan un RECM inferior a 2 metros (ver tabla 10). Aunque difiere del valor mínimo esperado con la corrección del filtro de Kalman, son valores admisibles ya que la plataforma está sometida a condiciones externas como cambios en el viento local o un menor porcentaje de batería.

Los resultados de cobertura son especificados en la tabla 15, la cual evidencia que la ruta basada en el algoritmo de Dijkstra, la cual fue seleccionada por **DronePlanner**, no cumple con el porcentaje de cobertura esperado (la cobertura real de una celda se establece si el cuadricóptero visita el rango del tamaño de la celda), sin embargo, es la ruta más corta y con menos revisitas. La ruta de cobertura basada en propagación de frente de onda, que fue catalogada con el menor rendimiento por **DronePlanner**, es la ruta con la menor cantidad de giros y la distancia más larga; y finalmente, la ruta de cobertura basada en espirales conserva el mayor porcentaje de cobertura entre las tres.

De acuerdo a lo anterior, se puede decir que, para el área de cobertura seleccionada, la ruta basada en propagación de frente de onda cumple con el objetivo del *sensado remoto*, sin embargo, el gasto energético mínimo en cuanto a tiempo y revisitas, es requisito para asegurar que el AR Drone, el cual tiene una autonomía limitada, complete la ruta sin riesgos de perder o dañar la plataforma. Por lo que se determina que la ruta de cobertura con el algoritmo basado en espirales, es la ruta óptima para el cultivo en estudio, ya que tiene el mayor porcentaje de cobertura, y conserva los valores medios entre en el rango de rendimiento de los parámetros de evaluación.

5.3.5 P2.3: Mosaicos finales del Sensado remoto con las rutas de cobertura

La última fase del protocolo de pruebas incluye la unión de las fotografías recolectadas por las rutas para formar el mosaico final de la información capturada. El mosaico es obtenido mediante el software especializado en fotogrametría aérea llamado Pix4D en su versión de prueba [113], propendiendo que el cocido de fotografías conserve la dirección en las que las fotografías fueron capturadas. Los

mosaicos de las rutas de cobertura desplegadas para el cultivo de pasto gramillón son apreciables en las figuras 58, 59 y 60.



Figura 58. Mosaico final con la ruta de cobertura basada en propagación de frente de onda

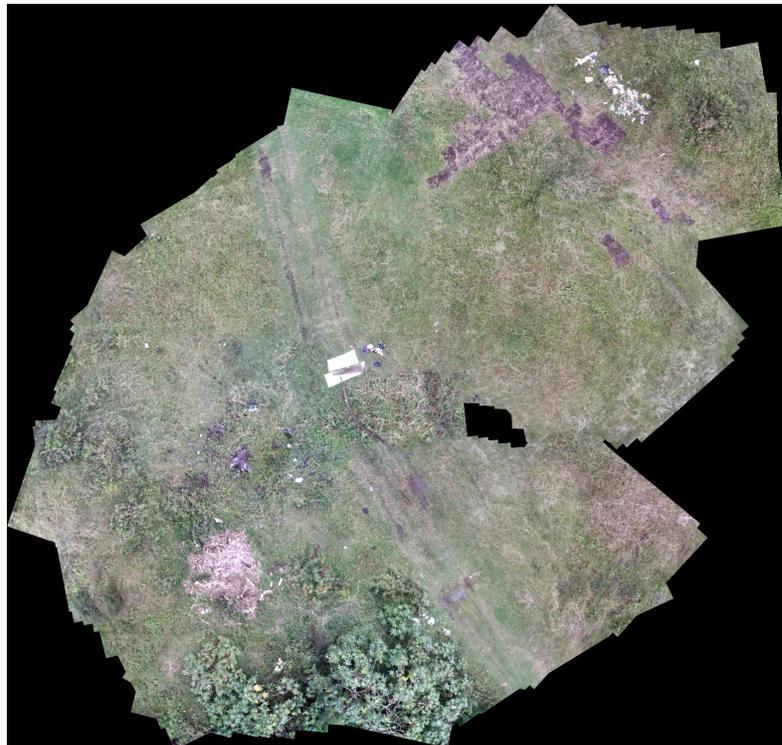


Figura 59. Mosaico final con la ruta de cobertura basada espirales

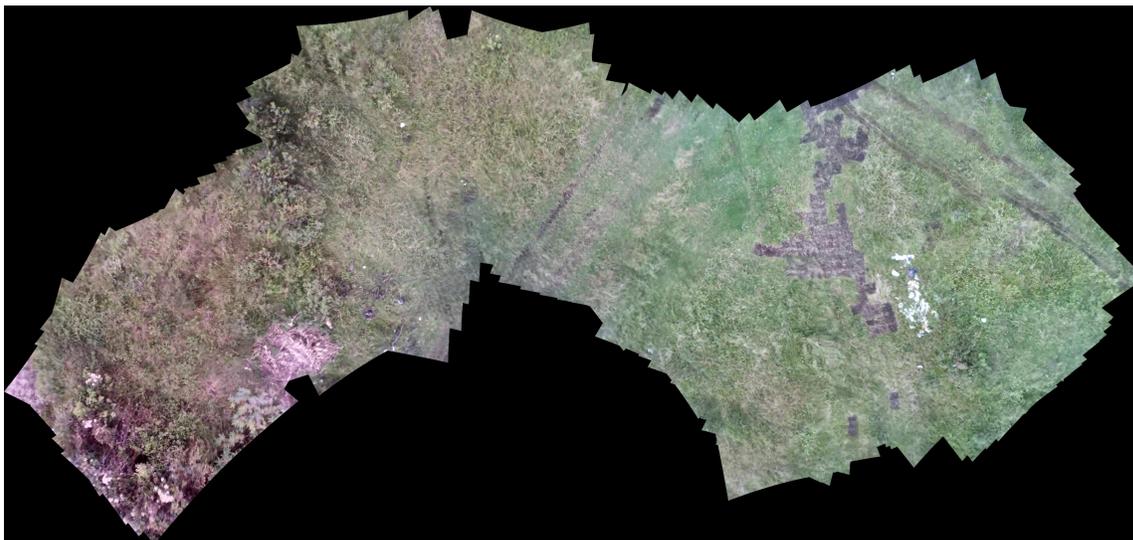


Figura 60. Mosaico final con la ruta de cobertura basada en Dijkstra

La ruta de cobertura basada en Dijkstra de la figura 60, evidencia los resultados de la tabla 15, ya que la cobertura denota un porcentaje bajo. La ruta de cobertura basada en la propagación del frente de onda de la figura 58, conserva la cobertura esperada, sin embargo, es la ruta basada en espirales de la figura 59, en la que se aprecia un porcentaje de cobertura mayor, a pensar de observar ausencia de información.

5.4 Resumen

El capítulo 5 describe en detalle los resultados obtenidos con la puesta en funcionamiento del sistema de *sensado remoto* propuesto. Los resultados siguen un orden establecido por el plan de pruebas el cual evidencia el alcance de cada objetivo de la propuesta de maestría. La descripción de resultados incluye la selección de la estrategia de posicionamiento del AR Drone para ambientes abiertos como los cultivos, basada en un posicionamiento. Seguidamente, describe los resultados obtenidos con el filtro de Kalman extendido (FKE) el cual evidencia un RECM mínimo de 0,54 metros en el plano y de 0,10 metros en altura. El siguiente modulo a validar son los algoritmos de cobertura, los cuales evidencian el éxito del *sensado remoto* con el cuadricóptero AR Drone considerando factores externos como el viento. En esta fase se coloca a prueba el rendimiento de los tres algoritmos, con lo cual se llega a que la ruta basada en espirales puede ser considerada la ruta óptima para el cultivo de pasto gramillón en estudio. Los mosaicos de fotografías capturadas con cada ruta de cobertura, evidencia la inferencia de que la ruta basada en espirales es óptima para el *sensado remoto* del cultivo, a pensar de observar pérdida de información en el mosaico final.

CONCLUSIONES Y TRABAJOS FUTUROS

El presente capítulo describe las conclusiones generales del trabajo desarrollado para la construcción de un sistema de *sensado remoto* basado en el cuadricóptero AR Drone, en ellas no solo se refleja la conclusión de las pruebas del capítulo 7, sino también de toda la experimentación realizada para lograr los objetivos propuestos. El capítulo finaliza con la descripción de los trabajos futuros, los cuales representan las brechas del trabajo desarrollado.

6.1 Conclusiones

- La fusión de datos mediante FKE para optimizar la posición y la robustez del cuadricóptero ante las condiciones de un entorno agrícola, mantiene márgenes mínimos de error cuadrático medio no superiores a 0.54 m en el plano y 0.10 m en altura, con respecto a las coordenadas planificadas. Sin embargo, existe una tendencia a deriva (perdida de información), ya que las señales del receptor GPS no aseguran una precisión en el tiempo.
- El software de planificación diseñado, **DronePlanner**, es una herramienta fundamental para el sistema de *sensado remoto*, ya que realiza la transformación de la ruta trazada en el mapa (offline) a la ruta seguida en un ambiente real. Su desarrollo central, está basado en imágenes georreferenciadas mediante las cuales se calcula la navegación del AR Drone en un sistema de posicionamiento global. **DronePlanner** compara el comportamiento de los algoritmos calculados y predice cuál de los tres tendrá mayor eficiencia energética, basado en el número de giros, visitas y distancia calculada.
- Con un posicionamiento confiable se despliegan algoritmos de cobertura los cuales son calculados en tierra mediante DronePlanner. La experimentación coloca a prueba tres algoritmos para descomposición de área aproximada, de los cuales sobresale la ruta de cobertura basada en espirales por cobertura completa y la ruta de cobertura basada en Dijkstra con la mínima distancia. Sin embargo, la presente investigación no define un único algoritmo para el sistema ya que la eficiencia en vuelo depende de factores externos como el viento y los obstáculos o áreas de poco interés del área de cobertura.
- El sistema de *sensado remoto* desarrollado gestiona la captura de imágenes con un solapamiento promedio no mayor a 63%, sin embargo, de acuerdo a

los mosaicos obtenidos, para lograr una mayor precisión en el mosaico de las fotografías se requiere de una plataforma más robusta para maniobrar de mejor manera los giros y las ráfagas de vientos locales, las cuales afectan drásticamente el comportamiento de las rutas de cobertura esperadas.

- A diferencia de los sistemas de *sensado remoto* conocidos comercialmente, el prototipo desarrollado en esta investigación está enfocado en estrategias para el uso de vehículos aéreos de bajo costo en aplicaciones de alto impacto regional como la agricultura. Además, el sistema de *sensado remoto* optimiza la autonomía de la plataforma aérea considerando una ruta basada en el viento, obstáculos (o zonas de poco interés) y el sensor portado (cámara frontal AR Drone).

6.2 Trabajos Futuros

- ***Replantear el método para captura de imágenes***

El prototipo desarrollado captura las imágenes y la almacena en la memoria disponible en la Raspberry Pi, por lo que el número de fotografías está sesgado a dicho espacio de memoria. Una posible solución es usar el protocolo MAVLINK para formatear las imágenes y almacenarlas en la estación en tierra.

- ***Ampliación del sistema de sensado remoto desarrollado para soportar otras plataformas aéreas con mayor autonomía y capacidad de carga útil.***

El sistema de *sensado remoto* implementado se centra en la plataforma aérea AR Drone, sin embargo, su autonomía y capacidad de carga útil no son adecuadas para las aplicaciones en agricultura, las cuales buscan la exploración y monitoreo de extensas áreas y el porte de cámaras especializadas para fotogrametría. Por tanto, se sugiere ampliar la investigación con otras plataformas de bajo costo.

- ***Validar nuevos algoritmos para optimizar cobertura completa.***

El software de planificación desarrollado ***DronePlanner***, está basado en el cálculo de rutas de cobertura globales, por lo cual es posible implementar más algoritmos de cobertura mientras ellos estén desarrollados usando búsqueda de grafos y se cuente con el mapa digital del área de cobertura. La integración de nuevos algoritmos contribuirá a encontrar la mejor estrategia de cobertura usando vehículos aéreos multirrotores.

- ***Realizar fotogrametría especializada con los mosaicos obtenidos***

El sistema de *sensado remoto* planteado llega hasta la primera fase de la agricultura de precisión, es decir la adquisición de información, sin embargo,

el objetivo final del *sensado remoto* es analizar la información recolectada para apoyar los procesos de fertilización, monitoreo y siembra de cultivos. Lo anterior sugiere que se realice un procesamiento de imágenes de los mosaicos obtenidos y analizarlos mediante el cálculo de índices de vegetación.

- ***Aumentar cobertura con un grupo de vehículos aéreos trabajando cooperativamente.***

La investigación de un sistema de *sensado remoto* basado en un robot aéreo multirroto, está limitada a la autonomía que tienen estas plataformas, por lo que se propone utilizar más de un vehículo con la idea de cubrir áreas más extensas sin temor a pérdida de información y/o de la plataforma.

BIBLIOGRAFÍA

- [1] T. Reardon y J. A. Berdegué, «La rápida expansión de los supermercados en América Latina: Desafíos y oportunidades para el desarrollo,» *Estudios Sociedade e Agricultura*, 2013.
- [2] J. V. Stafford, «Implementing precision agriculture in the 21st century,» *Journal of Agricultural Engineering Research*, pp. 267-275, 2000.
- [3] R. Bongiovanni y J. Lowenberg-Deboer, «Precision agriculture and sustainability,» *Precision Agriculture*, pp. 359-387, 2004.
- [4] N. Zhang, M. Wang y N. Wang, «Precision agriculture—a worldwide overview,» *Computers and electronics in agriculture*, pp. 113-132, 2002.
- [5] G. Joseph, *Fundamentals of remote sensing*, Universities Press, 2005.
- [6] A. N. Rencz y R. A. Ryerson, *Manual of Remote Sensing, Remote Sensing for the Earth Sciences*, John Wiley & Sons, 1999.
- [7] S. Liaghat y S. K. Balasundram, «A review: The role of remote sensing in precision agriculture,» *American Journal of Agricultural and Biological Sciences*, vol. 5, nº 1, p. 50, 2010.
- [8] A. S. Gibb, «DRONING THE STORY,» UNIVERSITY OF BRITISH COLUMBIA, BRITISH, COLUMBIA, 2013.
- [9] S. Mahmood, *Unmanned Aerial Vehicle (UAV) Communications*, South Australia, : University of South Australia., 2007.
- [10] S. G. Gupta, M. M. Ghonge y P. Jawandhiya, «Review of unmanned aircraft system (UAS),» *technology*, vol. 2, nº 4, 2013.
- [11] Ministerio de Defensa del Reino Unido, «UNMANNED AIRCRAFT SYSTEMS: TERMINOLOGY, DEFINITIONS AND CLASSIFICATION,» 2010.
- [12] H. Gao, S. L. Kong, S. Zhou, F. Lv y Q. Chen, «Automatic Extraction of Multi-Vehicle Trajectory Based on Traffic Videotaping from Quadcopter Model,» de *Applied Mechanics and Materials*, Trans Tech Publ, 2014, pp. 232-239.
- [13] H. A. Almurib, P. T. Nathan y T. N. Kumar, «Control and path planning of quadrotor aerial vehicles for search and rescue,» de *SICE Annual Conference*, IEEE, 2011, pp. 700-705.
- [14] J. Jiangtao, H. Feifei, H. Zhitao, D. Xinwu, L. Jianjun, Z. Zhihua, D. Mengmeng, H. Yakai y M. Junwei, «The Application of Quad-Rotor UAV in Farmland Information Acquisition,» *Journal of Agricultural Mechanization Research*, vol. 2, p. 3, 2013.

- [15] M. Anudeep, G. Diwakar y R. Katukam, «Design of A Quad Copter and Fabrication,» *International Journal of Innovations in Engineering and Technology (IJJET)*, vol. 4, pp. 59-65, 2014.
- [16] Y. Naidoo, R. Stopforth y G. Bright, «Quad-Rotor Unmanned Aerial Vehicle Helicopter Modelling & Control.,» *International Journal of Advanced Robotic Systems*, 2011.
- [17] A. Rodic y G. Mester, «The modeling and simulation of an autonomous quad-rotor microcopter in a virtual outdoor scenario,» *Acta Polytechnica Hungarica*, pp. 107-122, 2011.
- [18] K. Petersen, R. Feldt, S. Mujtaba y M. Mattsson, «Systematic mapping studies in software engineering,» de *12th International Conference on Evaluation and Assessment in Software Engineering*, 2008.
- [19] V. Kumar y N. Michael, «Opportunities and challenges with autonomous micro aerial vehicles,» *The International Journal of Robotics Research*, pp. 1279-1291, 2012.
- [20] P. Wallich, «Arducopter parenting,» *Spectrum, IEEE*, vol. 49, nº 12, pp. 26-28, 2012.
- [21] P. S. Bhatkhande, REAL TIME FUZZY CONTROLLER FOR QUADROTOR STABILITY CONTROL, Michigan Technological University, 2014.
- [22] I. Monzón Catalán, Desarrollo de un cuadricóptero, Universidad de Zaragoza, 2013.
- [23] P. McKerrow, «Modelling the Draganflyer four-rotor helicopter,» de *Robotics and Automation, 2004. Proceedings, IEEE*, 2004, pp. 3596-3601.
- [24] K. Lillywhite, D.-J. Lee, B. Tippetts, S. Fowers, A. Dennis, B. Nelson y J. Archibald, «An embedded vision system for an unmanned four-rotor helicopter,» de *Optics East 2006*, International Society for Optics and Photonics, 2006, pp. 63840G-63840G.
- [25] M. Nahon, I. Sharf, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok y T. Wang, «Ground effect experiments and model validation with Draganflyer X8 rotorcraft,» de *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, Orlando, FL, IEEE, 2014, pp. 1158 - 1166.
- [26] F. Neitzel y J. Klonowski, «Mobile 3D mapping with a low-cost UAV system,» *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 38, pp. 1-6, 2011.
- [27] M. Baumker y H. Przybilla, «Investigations on the accuracy of the navigation data of unmanned aerial vehicles using the example of the system Mikrokopter,» *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3822, nº 113, 2011.
- [28] M. Leichtfried, C. Kaltenriner, A. Mossel y H. Kaufmann, «Autonomous Flight using a Smartphone as On-Board Processing Unit in GPS-Denied Environments,» de *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, ACM, 2013, p. 341.
- [29] D. Tinker, «Adding Vision to a Quadrotor: A Design-Build-Test Adventure,» Western Kentucky University, Bowling Green, Kentucky, 2014.

- [30] M. Saska, T. Krajnik, J. Faigl, V. Vonasek y L. Preucil, «Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation,» de *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 4808-4809.
- [31] L. Jayatilleke y N. Zhang, «Landmark-based localization for unmanned aerial vehicles,» de *Systems Conference (SysCon), 2013 IEEE International*, IEEE, 2013, pp. 448-451.
- [32] J. Rodriguez, C. Castiblanco, I. Mondragon y J. Colorado, «Low-cost quadrotor applied for visual detection of landmine-like objects,» de *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, Orlando, FL, IEEE, 2014, pp. 83-88.
- [33] P. T. T. Ha, H. Yamamoto y K. Yamazaki, «Using Autonomous Air Vehicle in DTN Sensor Network for Environmental Observation,» de *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, Kyoto, IEEE, 2013, pp. 447-450.
- [34] K. Okutani, T. Yoshida, K. Nakamura y K. Nakadai, «Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter,» de *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Vilamoura, IEEE, 2012, pp. 3288-3293.
- [35] R. Williams, B. Konev y F. Coenen, «Multi-agent Environment Exploration with AR. Drones,» de *Advances in Autonomous Robotics Systems*, Springer, 2014, pp. 60-71.
- [36] K. Pfeil, S. L. Koh y J. LaViola, «Exploring 3d gesture metaphors for interaction with unmanned aerial vehicles,» de *Proceedings of the 2013 international conference on Intelligent user interfaces*, ACM, 2013, pp. 257-266.
- [37] S. Khatoon, M. Shahid y H. Chaudhary, «Dynamic modeling and stabilization of quadrotor using PID controller,» IEEE, 2014, pp. 746-750.
- [38] Y. Luo, M. J. Er, L. L. Yong y C. J. Chien, «Intelligent control and navigation of an indoor quad-copter,» de *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, IEEE, Ed., 2014, pp. 1700-1705.
- [39] L. M. Argentim, W. C. Rezende, P. E. Santos y R. A. Aguiar, «PID, LQR and LQR-PID on a quadcopter platform,» de *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*, Dhaka, 2013.
- [40] C. Ha, Z. Zuo y D. Lee, «Backstepping control of quadrotor-type UAVs and its application to teleoperation over the internet,» de *Intelligent Autonomous Systems 12*, 2013.
- [41] M. A. Mohd Basri, A. R. Husain y K. A. Danapalasingam, «Intelligent adaptive backstepping control for MIMO uncertain non-linear quadrotor helicopter systems,» *Transactions of the Institute of Measurement and Control*, vol. 37, nº 3, pp. 345-361, 2015.
- [42] M. K. Joyo, D. Hazry, S. F. Ahmed, M. H. Tanveer, F. A. Warsi y A. T. Hussain, «Altitude and Horizontal Motion Control of Quadrotor UAV in the Presence of Air Turbulence,» de *Conference on Systems, Process & Control (ICSPC2013)*, Kuala Lumpur, IEEE, 2013, pp. 16-20.

- [43] M. H. a. A. S. F. Tanveer, D. Hazry, F. A. Warsi y M. K. Joyo, «STABILIZED CONTROLLER DESIGN FOR ATTITUDE AND ALTITUDE CONTROLLING OF QUAD-ROTOR UNDER DISTURBANCE AND NOISY CONDITIONS.,» *American Journal of Applied Sciences*, 2013.
- [44] Y. Hong, X. Lin, Y. Zhuang y Y. Zhao, «Real-Time Pose Estimation and Motion Control for a Quadrotor UAV,» de *Intelligent Control and Automation (WCICA)*, 2014, IEEE, Ed., 2014, pp. 2370-2375.
- [45] M. Tailanian, S. Paternain, R. Rosa y R. Canetti, «Design and implementation of sensor data fusion for an autonomous quadrotor,» de *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, Montevideo, IEEE, 2014, pp. 1431-1436.
- [46] L. Kis y B. Lantos, «Orientation estimation of an outdoor vehicle using inertial, magnetic and CP-GPS sensors,» *Systems and Automatic Control*, 2013.
- [47] K. Chee y Z. Zhong, «Control, navigation and collision avoidance for an unmanned aerial vehicle,» *Sensors and Actuators A: Physical*, pp. 66-76, 2013.
- [48] E. Lopez, R. Barea, A. Gomez, A. Saltos, L. M. Bergasa, E. J. Molinos y A. Nemra, «Indoor SLAM for Micro Aerial Vehicles Using Visual and Laser Sensor Fusion,» de *ROBOT*, 2015, pp. 531-542.
- [49] E. Schnipke, S. Reidling, J. Meiring, W. Jeffers, M. Hashemi, R. Tan, A. Nemati y M. Kumar, «Autonomous navigation of uav through gps-denied indoor environment with obstacles,» de *AIAA Infotech@ Aerospace*, 2015.
- [50] N. Ellis, «Inspection of power transmission lines using UAVs,» *University of Southern Queensland*, 2013.
- [51] H. Choset, «Coverage for robotics--A survey of recent results,» *Annals of mathematics and artificial intelligence*, pp. 113-126, 2001.
- [52] K. Senthilkumar y K. Bharadwaj, «Spanning tree based terrain coverage by multi robots in unknown environments,» de *India Conference, 2008. INDICON 2008. Annual IEEE*, Kanpur, 2008.
- [53] K. Bharadwaj y K. Senthilkumar, «Multi-robot exploration and terrain coverage in an unknown environment,» *Robotics and Autonomous Systems*, pp. 123-132, 2012.
- [54] X. Qiu, S. Liu y S. X. Yang, «A rolling method for complete coverage path planning in uncertain environments,» de *Robotics and Biomimetics, 2004. ROBIO 2004*, Shenyang, 2004.
- [55] A. Dirafzoon, S. Emrani, S. A. Salehizadeh y M. B. Menhaj, «Coverage control in unknown environments using neural networks,» *Artificial Intelligence Review*, pp. 237-255, 2012.
- [56] M. Ozkan, A. Yazici, M. Kapanoglu y O. Parlaktuna, «A genetic algorithm for task completion time minimization for multi-robot sensor-based coverage,» de *Control Applications, (CCA) & Intelligent Control, (ISIC)*, Saint Petersburg, 2009.
- [57] M. A. Habib, M. S. Alam y N. H. Siddique, «Optimizing coverage performance of multiple random path-planning robots,» *Paladyn*, pp. 11-22, 2012.

- [58] P. Pignon y H. Choset, «Coverage path planning: The boustrophedon cellular decomposition,» *Field and Service Robotics*, pp. 203-209, 1998.
- [59] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra y C. Bustacara, «BSA: a complete coverage algorithm,» de *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005.
- [60] M. Morin, I. Abi-Zeid, Y. Petillot y C.-G. Quimper, «A hybrid algorithm for coverage path planning with imperfect sensors,» de *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Tokyo, 2013.
- [61] N. Agmon, N. Hazon y G. A. Kaminka, «Constructing spanning trees for efficient multi-robot coverage,» de *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, Orlando, FL, 2006.
- [62] D. Michel y K. Mclsaac, «New path planning scheme for complete coverage of mapped areas by single and multiple robots,» de *Mechatronics and Automation (ICMA), 2012 International Conference on*, Chengdu, 2012.
- [63] D. H. Kim, G. Hoang, M.-J. Bae, J. W. Kim, S. M. Yoon, T.-K. Yeo, H. Sup y S.-B. Kim, «Path tracking control coverage of a mining robot based on exhaustive path planning with exact cell decomposition,» de *Control, Automation and Systems (ICCAS), 2014 14th International Conference on*, Gyeonggi-do, South Korea, 2014.
- [64] L. Paull, S. Saeedi, H. Li y V. Myers, «An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar,» de *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, Toronto, ON, 2010.
- [65] A. Sonmez, E. Kocyigit y E. Kugu, «Optimal Path Planning for UAVs Using Genetic Algorithm,» de *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, IEEE, 2015, pp. 50-55.
- [66] Z. Wang y Z. Bo, «Coverage path planning for mobile robot based on genetic algorithm,» de *Electronics, Computer and Applications, 2014 IEEE Workshop on*, Ottawa, ON, 2014.
- [67] S.-W. Moon y D. H.-C. Shim, «Study on path planning algorithms for unmanned agricultural helicopters in complex environment,» *International Journal of Aeronautical and Space Sciences*, pp. 1-11, 2009.
- [68] M. Nieuwenhuisen, D. Droeschel, M. Beul y S. Behnke, «Obstacle detection and navigation planning for autonomous micro aerial vehicles,» de *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, Orlando, FL, 2014.
- [69] L. Rodriguez, J. Cobano y A. Ollero, «Efficient Local Path Planning for UAVs in unknown environments,» de *XV Workshop of physical agents: book of proceedings, WAF 2014, June 12th and 13th, 2014*, León, Spain, 2014.
- [70] R. He, S. Prentice y N. Roy, «Planning in information space for a quadrotor helicopter in a GPS-denied environment,» de *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, Pasadena, CA, 2008.

- [71] H. Bai, S. Shao y H. Wang, «A VTOL quadrotor platform for multi-UAV path planning,» de *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, Harbin, Heilongjiang, China, 2011.
- [72] C.-L. Lin, C.-S. Lee, Y.-J. Tsai y C.-H. Huang, «Flight path planning for mini rotor UAVs,» de *Control & Automation (ICCA), 11th IEEE International Conference on*, Taichung, 2014.
- [73] B. a. M. M. Sinopoli, G. Donato y T.-J. Koo, «Vision based navigation for an unmanned aerial vehicle,» de *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Seoul, Korea, 2001.
- [74] F. Adolf, F. Andert y J. G. F. Rocha, «Rapid On-Line Path Planning On Board A VTOL UAV,» *Proc of AIAA InfoTech@ Aerospace*, pp. 1-11, 2010.
- [75] B. MacAllister, J. Butzke, A. Kushleyev, H. Pandey y M. Likhachev, «Path planning for non-circular micro aerial vehicles in constrained environments,» de *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, Karlsruhe, 2013.
- [76] S. Lim y H. Bang, «Waypoint Planning Algorithm Using Cost Functions for Surveillance,» *International Journal of Aeronautical and Space Sciences*, pp. 136-144, 2010.
- [77] J. Valente, «An Aerial Robotic Framework to address Area Coverage in Precision Agriculture Practices,» Industriales, Madrid, 2011.
- [78] A. Kushleyev, D. Mellinger, C. Powers y V. Kumar, «Towards a swarm of agile micro quadrotors,» *Autonomous Robots*, pp. 287-300, 2013.
- [79] T. Stirling, J. Roberts, J.-C. Zufferey y D. Floreano, «Indoor navigation with a swarm of flying robots,» de *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, Saint Paul, MN, 2012.
- [80] H. Duan y P. Qiao, «Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning,» *International Journal of Intelligent Computing and Cybernetics*, vol. 7, n° 1, pp. 24-37, 2014.
- [81] V. Roldao, R. Cunha, D. Cabecinhas, C. Silvestre y P. Oliveira, «A leader-following trajectory generator with application to quadrotor formation flight,» *Robotics and Autonomous Systems*, 2014.
- [82] A. A. Bandala, E. P. Dadios, R. R. P. Vicerra y L. A. G. Lim, «Swarming Algorithm for Unmanned Aerial Vehicle (UAV) Quadrotors--Swarm Behavior for Aggregation, Foraging, Formation, and Tracking--,» *Journal of Advanced Computational Intelligence Vol*, 2014.
- [83] S. Oweis, S. Ganesan y K. C. Cheok, «Server based control flocking for aerial-systems,» de *Electro/Information Technology (EIT), 2014 IEEE International Conference on*, Milwaukee, WI, 2014.
- [84] G. Pajares, «Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs),» *Photogrammetric Engineering & Remote Sensing*, vol. 81, n° 4, pp. 281-329, 2015.

- [85] C. Holness, T. Matthews, K. Satchell y E. C. Swindell, «Remote Sensing Archeological Sites through Unmanned Aerial Vehicle (U.A.V.) Imaging,» de *Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 6695-6698.
- [86] A. Tamminga, C. Hugenholtz, B. Eaton y M. Lapointe, «Hyperspatial remote sensing of channel reach morphology and hydraulic fish habitat using an unmanned aerial vehicle (UAV): a first assessment in the context of river research and management,» *River Research and Applications*, vol. 31, n° 3, pp. 379-391, 2015.
- [87] K. Zainuddin, N. Ghazali y Z. M. Arof, «The Feasibility of Using Low-Cost Commercial Unmanned Aerial Vehicle for Small Area Topographic Mapping,» de *Aerospace Electronics and Remote Sensing Technology (ICARES)*, 2015, pp. 1-7.
- [88] J. a. D. G. S. F. a. F. E. a. G. L. a. L. E. a. M. A. a. V. F. P. Primicerio, «A flexible unmanned aerial vehicle for precision agriculture,» *Precision Agriculture*, vol. 13, n° 4, pp. 517-523, 2012.
- [89] D. a. E. S. Anthony, A. Lorenz y C. Detweiler, «On crop height estimation with UAVs,» de *Intelligent Robots and Systems (IROS 2014)*, 2014 *IEEE/RSJ International Conference on*, Chicago, IL, IEEE, 2014, pp. 4805-4812.
- [90] R. Jannoura, K. Brinkmann, D. Uteau, C. Bruns y R. G. Joergensen, «Monitoring of crop biomass using true colour aerial photographs taken from a remote controlled hexacopter,» *Biosystems Engineering*, vol. 129, pp. 341-351, 2015.
- [91] D. Gómez-Candón, A. De Castro y F. López-Granados, «Assessing the accuracy of mosaics from unmanned aerial vehicle (UAV) imagery for precision agriculture purposes in wheat,» *Precision Agriculture*, vol. 5, n° 1, pp. 44-56, 2014.
- [92] F. Mesas-Carrascosa, I. Clavero Rumbao, J. Torres-Sánchez, A. García-Ferrer, J. Peña y F. López Granados, «Accurate ortho-mosaicked six-band multispectral UAV images as affected by mission planning for precision agriculture proposes,» *International Journal of Remote Sensing*, pp. 1-16, 2016.
- [93] J. Navia, I. Mondragon, D. Patino y J. Colorado, «Multispectral mapping in agriculture: terrain mosaic using an autonomous quadcopter UAV.,» de *Unmanned Aircraft Systems (ICUAS)*, 2016 *International Conference on*, IEEE, 2016, pp. 1351-1358.
- [94] G. S. Avellar, G. A. Pereira, L. C. Pimenta y P. Iscold, «Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time,» *Sensors*, vol. 15, n° 11, pp. 27783-27803, 2015.
- [95] Parrot S.A., «Parrot,» 2012. [En línea]. Available: <http://ardrone2.parrot.com/ardrone-2/specifications/>. [Último acceso: 1 03 2016].

- [96] P.-J. Bristeau, F. Callou, D. Vissiere y N. Petit, «The navigation and control technology inside the ar. drone micro uav,» de *18th IFAC World Congress*, 2011, pp. 1477-1484.
- [97] Open Source Robotics Foundation, «ROS,» 2007. [En línea]. Available: <http://www.ros.org/about-ros/>. [Último acceso: 2 Marzo 2017].
- [98] E. Kuantama y E. R. Tarca, «QUADCOPTER ATTITUDE AND THRUST SIMULATION BASED ON SIMULINK PLATFORM.,» *Robotica & Management*, vol. 20, nº 2, pp. 40-44, 2015.
- [99] J. Engel, J. Sturm y D. Cremers, «Camera-based navigation of a low-cost quadcopter,» de *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Vilamoura, IEEE, 2012, pp. 2815 - 2821.
- [100] Parrot, «Ar Drone España,» 2013. [En línea]. Available: <http://www.arronespain.com/blog/2013/09/13/manual-de-manejo-de-ar-drone-2-con-qgroundcontrol/>. [Último acceso: 01 02 2016].
- [101] GNU Comunity, «Paparazzi: the free autopilot,» 2015. [En línea]. Available: http://wiki.paparazziuav.org/wiki/AR_Drone_2. [Último acceso: 01 03 2016].
- [102] R. Negenborn, «Kalman Filter Extensions,» de *Robot localization and Kalman filters*, Utrecht , Utrecht University, 2003.
- [103] F. L. Lewis, L. Xie y D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory*, vol. 29, CRC press, 2007.
- [104] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng y S. Thrun, «Discriminative Training of Kalman Filters,» de *Robotics: Science and systems*, vol. 2, Stanford, 2005, p. 1.
- [105] M. M. Flood, «The traveling-salesman problem,» *Operations Research*, vol. 4, nº 1, pp. 61-75, 1956.
- [106] E. W. Dijkstra, «A note on two problems in connexion with graphs,» *Numerische mathematik*, vol. 1, nº 1, pp. 269-271, 1959.
- [107] A. Zelinsky, R. Jarvis, J. Byrne y S. Yuta, «Planning paths of complete coverage of an unstructured environment by a mobile robot,» *Proceedings of International Conference on Advanced Robotics*, p. 533–538, 1993.
- [108] E. Chuvieco, «Fundamentos de teledetección espacial,» *Estudios Geográficos*, vol. 52, nº 203, p. 371, 1991.
- [109] Manual de Vuelo, «Navegación Aerea,» 2000. [En línea]. Available: <http://www.manualvuelo.com/NAV/NAV71.html>. [Último acceso: 2 Febrero 2017].
- [110] GitHub Community, «ardrone_autonomy,» GitHub, 2014.
- [111] J. H. Wilkinson, *The algebraic eigenvalue problem*, vol. 87, 1965.
- [112] Open Source Developers, «Qt Creator/es,» 3 Junio 2016. [En línea]. Available: https://wiki.qt.io/Qt_Creator/es. [Último acceso: 4 Marzo 2017].
- [113] EPFL, «Pix4D,» 2011. [En línea]. Available: <https://pix4d.com/>. [Último acceso: 20 Febrero 2017].