



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“Diseño de un sistema integral de planificación de rutas de vuelo para un dron agrícola multifuncional en cultivos de arroz”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:
INGENIERO EN COMPUTACIÓN

Narcisa Fabiola Colcha Melendrez
Bryan Dario Tumbaco Moreira

GUAYAQUIL – ECUADOR

AÑO: 2019

AGRADECIMIENTOS

Agradezco a mi familia y amigos que me apoyaron a lo largo de esta gran etapa de mi vida...

Narcisa Colcha

Agradezco a Dios por bendecir mi vida a lo largo de todos estos años de estudio, por la paciencia y amor a la carrera que me permitió culminarla.

A mis padres, hermanos y amigos por su comprensión y apoyo incondicional en todo tiempo; fueron el motor para alcanzar nuevos desafíos.

A mis maestros de la facultad por sus enseñanzas académicas, lecciones aprendidas y sobre todo por sus acertados consejos de vida cuando los necesitaba.

Bryan Tumbaco

DEDICATORIA

Dedico este logro a mis padres...

Narcisa Colcha

Dedico este logro a Dios pues de ÉL es la gloria y la alabanza, quien a lo largo de mi vida ha sido guía, soporte y refugio, sin apartarse ni un momento de su fidelidad para conmigo.

Bryan Tumbaco

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Narcisa Colcha Melendrez

Bryan Tumbaco Moreira

RESUMEN

El presente proyecto es un punto de partida para la adaptación de tecnología de código libre a las actividades relacionadas con el sector agrícola del país. En vista de que, para la producción arrocerá se gasta más de lo necesario en insumos agrícolas que son desperdiciados por el uso de bombas, como práctica tradicional de fumigación, utilizadas por agricultores que sin precisión no cubren de forma homogénea el cultivo. Además, están expuestos a riesgos laborales ocasionados por intoxicaciones con el producto químico que manipulan. Por lo tanto, la implementación de nuevas tecnologías en el agro busca reducir afectaciones humanas y garantizar una producción de calidad del arroz. Para esto se propuso el diseño e implementación de un sistema integral de locomoción y planificación de rutas albergado por un dron con aplicaciones agrícolas, como la fumigación de plaguicidas y aspersión de fertilizantes, con el fin de ganar eficiencia con un mecanismo multifuncional que opera en su totalidad mediante el uso de *software* libre. Esto se logró con el análisis del proyecto *ArduPilot*, al utilizar la controladora de vuelo *Pixhawk* para un dron *small* de 4 rotores y vuelos planificados con *Mission Planner*, que luego fueron implementados sobre la estructura de un dron comercial de uso agrícola, para finalmente integrar los sistemas de fumigación y aspersión de fertilizantes. El dron agrícola debía ejecutar de forma automática un vuelo de 20 minutos a 4 metros por segundo y cubrir 2 hectáreas a 2 metros de altura del terreno con una sola carga de batería. De lo cual, según simulaciones, el dron pudo abarcar 1.10 hectáreas en el peor de los casos, 1.94 hectáreas en el mejor de los casos, por lo tanto, en promedio se logró 1.52 hectáreas en 20 minutos y 4.56 hectáreas por hora.

Palabras Clave: dron, fumigación, fertilización, tecnología libre.

ABSTRACT

This current project is a starting point for the adaptation of open source technology to activities related to the country's agricultural sector. In view of that, for rice production, more than necessary is spent on agricultural inputs that are wasted due to the use of pumps, as a traditional fumigation practice, used by farmers who do not accurately cover the crop in a homogenous way. In addition, they are exposed to occupational hazards caused by poisoning with the chemical they handle. Therefore, the implementation of new technologies in agriculture seeks to reduce human affectations and ensure quality rice production. For this purpose, the design and implementation of an integral system of locomotion and route planning put on a drone with agricultural applications, such as pesticide spraying and fertilizer sprinkling, was proposed in order to gain efficiency with a multifunctional mechanism that operates in its whole through the use of free software. This was achieved with the analysis of the ArduPilot project, by using the Pixhawk flight controller for a small drone of 4 rotors and flights planned with Mission Planner, which were then implemented on the structure of a commercial drone for agricultural use, to finally integrate the fertilizer sprinkling and spraying systems. The agricultural drone was to automatically execute a 20 minutes flight at 4 meters per second and cover 2 hectares at 2 meters high with a single battery charge. Of which, according to simulations, the drone could cover 1.10 hectares in the worst case, 1.94 hectares in the best case, therefore, on average 1.52 hectares were achieved in 20 minutes and 4.56 hectares per hour.

Keywords: drone, fumigation, fertilization, free technology.

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT	II
ÍNDICE GENERAL.....	III
ABREVIATURAS	VI
SIMBOLOGÍA.....	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	IX
CAPITULO I	10
1. INTRODUCCIÓN	10
1.1 Descripción del Problema.....	10
1.2 Justificación	11
1.3 Objetivos.....	12
1.3.1 Objetivo General	12
1.3.2 Objetivos Específicos.....	12
1.4 Marco Teórico	12
1.4.1 Dron agrícola.....	12
1.4.2 Fumigación y aspersión	13
1.4.3 Dron agrícola comercial y de código abierto.....	13
1.4.4 Situación actual del arroz en Ecuador y el mundo	15
1.4.5 Mejoras en la fumigación agrícola con el uso de drones.....	15
CAPÍTULO II	16
2. METODOLOGÍA Y SOLUCIÓN	16
2.1 Metodología.....	17
2.1.1 Método de investigación	17
2.1.2 Modelo de desarrollo de software.....	18
2.2 Requerimientos de hardware	18
2.2.1 Autopiloto <i>Pixhawk</i>	18
2.2.2 Componentes electrónicos.....	19
2.2.3 Componentes mecánicos	20
2.2.4 Componentes para el sistema de fumigación y fertilización	20
2.2.5 Estructura de montaje	21
2.2.6 <i>Raspberry Pi 3</i>	22
2.3 Recursos de <i>software</i>	22

2.3.1	<i>ArduPilot</i>	23
2.3.2	<i>Mission Planner</i>	23
2.3.3	<i>Software en el bucle: SITL</i>	23
2.3.4	<i>Microsoft Visual Studio Community 2017</i>	24
2.3.5	<i>Microsoft .NET</i>	24
2.3.6	<i>GMAP.NET</i>	24
2.3.7	<i>PostgreSQL</i>	25
2.4	Diseño de la solución.....	25
2.4.1	Mantener altura	28
2.4.2	Esquivar obstáculos.....	29
2.4.3	Simulación de pruebas de vuelos críticos	29
2.4.4	Adquisición de datos de vuelo.....	30
2.5	Plan de ejecución de vuelos	30
CAPÍTULO III		32
3.	IMPLEMENTACIÓN DE LA SOLUCIÓN	32
3.1	Instalación de hardware libre	32
3.1.1	Configuración	32
3.1.2	Calibración	34
3.1.3	Puesta en marcha.....	34
3.2	Integración de sistemas.....	35
3.2.1	Sistema de fumigación de plaguicidas.....	36
3.2.2	Sistema de aspersión de fertilizantes.....	37
3.3	Identificación del sistema de respuesta de frecuencia.....	37
3.4	Marcación de Obstáculos	40
3.5	Ejecución de vuelos	42
3.5.1.	Registros de vuelos almacenados	43
CAPÍTULO IV		45
4.	ANÁLISIS DE RESULTADOS	45
4.1	Evaluación de rutas de vuelo	45
4.1.1	Vuelos simulados	46
4.1.2	Vuelos del dron <i>small</i>	48
4.1.3	Vuelos del dron agrícola multifuncional.....	53
CAPITULO V.....		55
5.	CONCLUSIONES, RECOMENDACIONES Y TRABAJO A FUTURO	55
5.1	Conclusiones	55
5.2	Limitaciones	55

5.3 Recomendaciones	56
5.4 Trabajos a futuro	56
BIBLIOGRAFÍA.....	57
ANEXOS.....	61

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
CIDIS	Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales
GEA	Granja Experimental Agrícola
GCS	Ground Control Station
FC	Flight Control
SBC	Single Board Computer
PDB	Power Distribution Board
FMU	Flight Management Unit
RAM	Random Access Memory
PWM	Pulse Width Modulation
GPS	Global Positioning System
USB	Universal Serial Bus
ESC	Electronic Speed Controller
BEC	Battery Elimination Circuit
RC	Remote Control
UAV	Unmanned Aerial Vehicle
UART	Universal Asynchronous Receiver/Transmitter
I2C	Inter-Integrated Circuit
GPIO	General-Purpose Input/Output
UDP	User Datagram Protocol
UgCS	Estación de Control de Tierra Universal
EKF	Extended Kalman Filter
RC	Radio Control
SITL	Software in the Loop
RTL	Return to Launch
PID	Proporcional, Integral y Derivativo

SIMBOLOGÍA

m	Metro
mm	Milímetro
ha	Hectárea
mi	Milla
t	Tonelada
tm	Tonelada métrica
K	Kilo
seg	Segundo
min	Minuto
h	Hora
Hz	Hercio
MHz	Megahercio
GHz	Gigahercio
Kbps	Kilo bits por segundo
A	Amperio
mA	Miliamperio
V	Voltio
C	Coulomb
mAh	Miliamperio por hora
m/s	Metro por segundos
LiPo	Litio Polímero
#S	Número de celdas de una batería
Rx	Receptor
Tx	Transmisor
m ²	Metros cuadrados

ÍNDICE DE FIGURAS

Figura 2.1. Panel web de Trello con el flujo de trabajo del tablero Kanban.....	17
Figura 2.2. Muestra de componentes electrónicos conectados al Pixhawk 2.4.8.....	20
Figura 2.3. Vista panorámica del modelo 3D de un dron comercial	22
Figura 2.4. Raspberry Pi 3 model B V1.2	22
Figura 2.5. Arquitectura SITL [22]	24
Figura 2.6. Dron small	25
Figura 2.7. Dron agrícola multifuncional	26
Figura 2.8. Esquema general de la solución del proyecto	27
Figura 2.9. Desglose de funciones que el sistema integral de planificación de rutas de vuelo debe cumplir.	28
Figura 3.1. Diagrama de orden de los motores en configuración Multirotor en x Hexa. CW: giro en sentido horario. CCW: giro en sentido anti-horario.	33
Figura 3.2. Entrada de la tasa de balanceo	39
Figura 3.3. Entrada de la tasa de cabeceo	39
Figura 3.4. Diagrama de la tasa de guiñada.....	39
Figura 3.5. Software Open Source Mission Planner con ícono para marcar obstáculos	40
Figura 3.6. Aplicación de código libre <i>Mission Planner</i> : marcación del posible obstáculo.....	41
Figura 3.7. Software Open Source Mission Planner con recorrido esquivando el obstáculo.....	41
Figura 3.8. Software Open Source Mission Planner con simulación.	42
Figura 3.9. Visualizador del mapa de Mission Planner mientras el dron small realiza un vuelo automático con polígono definido por el usuario.....	43
Figura 3.10. Diagrama Entidad-Relación de la base de datos drone_flight.....	44
Figura 4.1. Porcentajes del error relativo entre altura real y altura desea	50
Figura 4.2. Porcentajes del error relativo de la diferencia entre la altura deseada y la real.	51
Figura 4.3. Gráfica de altura deseada y calculada durante el vuelo con referencia relativa. Mapa de ruta ejecutada.....	52
Figura 4.4. Gráfica de altura deseada y calculada durante el vuelo con referencia absoluta. Mapa de ruta ejecutada	52
Figura 4.5. Gráfica de altura deseada y calculada durante el vuelo con referencia respecto al terreno. Mapa de ruta ejecutada	53
Figura 4.6. Gráfica del desfase entre ángulo pitch deseado y real.....	54
Figura 4.7. Gráfica del desfase entre el ángulo roll deseado y real.....	54

ÍNDICE DE TABLAS

Tabla 1.1. Software comercial para controladoras y planificadores de vuelo.	14
Tabla 1.2. Software libre para controladoras y planificadores de vuelo.	14
Tabla 1.3. Porcentajes de la superficie y producción total sembrada y cosechada de cultivos de arroz en Guayas y Los Ríos en el 2018 [9], [10].	15
Tabla 1.4. Comparación de métodos tradicionales de fumigación vs. dron fumigador [3].	16
Tabla 2.1. Requisitos del prototipo final	28
Tabla 2.2. Planificación de vuelos para un dron small.	31
Tabla 2.3. Planificación de pruebas de vuelo.	31
Tabla 3.1. Parámetros de entrada para el barrido de la frecuencia	38
Tabla 4.1. Parámetros globales de simulaciones de vuelo con 6 metros de separación entre las líneas del recorrido.	46
Tabla 4.2. Parámetros de las simulaciones de vuelo a 3 m/s de velocidad	46
Tabla 4.3. Parámetros de las simulaciones de vuelo a 4 m/s de velocidad	46
Tabla 4.4. Parámetros de las simulaciones de vuelo a 5 m/s de velocidad	47
Tabla 4.5. Propiedades de portátil usada para las pruebas de vuelo simuladas y reales.	47
Tabla 4.6. Resultados de vuelos simulados estimados al tiempo de duración de las baterías que es 20 min.	48
Tabla 4.7. Resultados de registros de datos de vuelos controlados de forma manual acerca de la actitud, altitud y potencia de motores.	49

CAPITULO I

1. INTRODUCCIÓN

En esta sección se describen los antecedentes, problemas y soluciones que han impulsado a realizar la investigación y optimización del plan de vuelo para drones agrícolas, tanto del área tecnológica como del área agrícola arroceras debido a que son los pilares fundamentales del proyecto. Además, se detalla las funciones de fumigar, rociar y esquivar obstáculos; capacidades que tiene el dron incorporadas al hacer uso del software libre.

Actualmente se usan en gran medida métodos tradicionales que son: las avionetas, tractores, helicópteros y/o personas, para fumigar y rociar los cultivos de arroz. Sin embargo, estas soluciones son poco eficientes y costosas para pequeños y medianos productores, puesto que, la distribución de los plaguicidas o fertilizantes que se realizan no es uniforme y se desperdician insumos.

Este proyecto ofrece un preámbulo de cómo la sociedad puede colaborar en el desarrollo de *software* libre, teniendo como objetivo aumentar la adaptación de la tecnología en el sector agrícola mediante drones.

1.1 Descripción del Problema

La implementación de nuevas tecnologías en el agro busca reducir afectaciones humanas y medio ambientales, causadas por la mala aplicación de insumos sobre los cultivos, para lograr una mayor producción y rendimiento del producto sembrado, con la menor cantidad posible de plaguicidas y fertilizantes. Sin embargo, los métodos tradicionales para la fumigación y aplicación de fertilizantes son poco eficientes, lo cual genera costos elevados de producción, mala calidad y bajo rendimiento del cultivo.

A su vez, los pequeños y medianos productores de la gramínea apuestan por realizar este procedimiento de forma manual, en particular con bomba manual, sin la protección adecuada, por lo cual están expuestos comúnmente a intoxicaciones por vía inhalatoria o cutánea, causando enfermedades que

pueden llegar a ocasionarles hasta la muerte. Como resultado, en países en desarrollo 3 millones de personas sufren envenenamiento, de las cuales se estima que 370,000 mueren cada año por la ingesta inintencionada de plaguicidas, según la Organización Mundial de la Salud (OMS) [1]. Inclusive, en Ecuador, el Centro de Información y Asesoramiento Toxicológico (CIATOX) del Ministerio de Salud Pública (MSP) registró 5,283 casos de intoxicaciones por plaguicidas de uso doméstico y agrícola, durante el período 2015 - 2017 [2].

Además, la fumigación que realiza el agricultor no es homogénea durante la jornada laboral, puesto que, el cansancio, insolación y la fatiga provocan disminución en la intensidad con la que empieza a laborar, lo que genera que el plaguicida se concentre más en ciertas zonas de cultivos que en otras. Igualmente, los tractores para fumigación utilizan mala atomización del líquido y desperdician hasta un 70% del plaguicida mezclado con grandes cantidades de agua. Aunque, las avionetas utilizan atomización a bajo volumen, debido al sobrevuelo de mínimo 30 metros sobre el cultivo, y por la velocidad de la aeronave, se aumenta el área de fumigación en los campos y el plaguicida no alcanza a ser rociado en toda la planta [3][4]. Entonces, no es recomendable llegar a fumigar hasta los bordes de las plantaciones, ya que el químico afectará a terrenos aledaños donde residen personas. Esto evidencia que cuando el hombre controla directamente la fumigación, ya sea con bombas manuales o vehículos, no hay precisión al momento de arrojar el plaguicida. En efecto, cada intento por perfeccionar esta práctica trae consigo otros problemas aún mayores, como la contaminación ambiental.

1.2 Justificación

Por todo lo expuesto anteriormente se evidencia que, las prácticas tradicionales de fumigación y aplicación de fertilizantes no garantizan una producción de arroz de calidad, porque se gasta más de lo necesario en insumos agrícolas que luego son desperdiciados con estas prácticas. Además, la fumigación tradicional conlleva a la acción de riesgos laborales y medio ambientales,

puesto que aparte de no ser homogénea, dispersa inadecuadamente el plaguicida lejos de la planta.

Por lo tanto, es de vital importancia automatizar el proceso de fumigación mediante el uso de tecnología amigable con el medio ambiente, sin requerir la intervención directa del ser humano. Más bien, se prescinde de naves autónomas que sobrevuelan a pocos metros de altura sobre los extensos cultivos de arroz recorriendo rutas sin peligro de colisionar con obstáculos, a fin de aumentar la eficiencia y practicidad en la fumigación y, no menos necesaria, fertilización del cultivo.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar un sistema integral de locomoción y planificación de rutas para un dron agrícola, logrando una mayor eficiencia en la fumigación y aspersión de fertilizantes sólidos, en cultivos de arroz, mediante el uso de software libre.

1.3.2 Objetivos Específicos

Evaluar rutas de vuelo por medio de tecnologías de código abierto.

Integrar el sistema de fumigación y aspersión de fertilizantes sólidos al dron agrícola.

Acoplar al dron, la instrucción de esquivar obstáculos, sin perder la posición actual de la ruta de vuelo.

1.4 Marco Teórico

1.4.1 Dron agrícola

Son vehículos aéreos no tripulados, creados para mejorar el sector agrícola al monitorear y aumentar la producción de los cultivos. Estos robots autónomos están equipados con distintos sensores y/o herramientas, los cuales permiten

que el dron agrícola ejecute diferentes funciones como: monitoreo de áreas protegidas, medición áreas y del terreno, salud del cultivo, inventario del cultivo, fumigación, aspersión, entre otras [5].

Se pueden manejar de manera manual o automática por medio del uso de *software* de planificación de vuelos y en su gran mayoría son comerciales.

1.4.2 Fumigación y aspersión

La fumigación permite combatir plagas, minimizar el uso de agroquímicos y la manipulación de estos con los seres humanos. A pesar de que los agroquímicos solo representan del 20% al 25% del costo total de la producción [6], es necesario que se implemente esta tecnología para ayudar al medio ambiente, aumentar la productividad de los cultivos en general y cuidar la salud de los agricultores.

Por su parte, la aspersión permite nutrir y hacer crecer las plantas mediante el riego superficial del cultivo. En efecto, el presente proyecto dirigido al cultivo de arroz pretende esparcir fertilizantes en estado sólido.

Ambos sistemas son montados al dron y puede ser utilizado uno a la vez, generando un sistema auto mecanizado de aspersión, como lo describen en [7].

1.4.3 Dron agrícola comercial y de código abierto

Existen drones comerciales y de software libre, en el ámbito comercial (ver Tabla 1.1) el área agrícola y tecnológica es liderada a nivel mundial por la empresa china *DJI*, la cual crea drones profesionales, de uso civil y sus propias aplicaciones para controlarlos; por otro lado, los de *software* libre (ver Tabla 1.2) son liderados por la empresa *Ardupilot* con su controladora de vuelo *Pixhawk* y *AMP*, las cuales permiten el uso de programas estación de control de tierra también libres, que son *Mission Planner*, *QGroundControl* y *APM Planner*, creadas por la misma empresa de software libre, como se puede ver en la Tabla 1.2.

Comerciales	Nombre de Empresa/Software	País	Descripción	Modelos/ Versiones
Controladora de Vuelo y Firmware	<i>DJI</i>	China	<i>Hardware y software.</i>	<i>Nasa, Phantom, AGRAS, T16</i>
	<i>Parrot</i>	Francia	<i>Hardware y software.</i>	Drones para mapeo.
	<i>Erle Robotics</i>	España	<i>Hardware y software.</i>	<i>ErleBrain: 2 y 3</i>
	<i>Yuneec</i>	China	<i>Hardware y software.</i>	Drones para mapeo.
Estación de control de tierra	<i>DJI Ground Station</i>	China	<i>Software.</i>	Actual V4.0
	<i>UAV Pilot</i>	China	<i>Software.</i>	Actual V3.0
	<i>UgCS</i>	Letonia	<i>Software.</i>	Actual V3.2

Tabla 1.1. Software comercial para controladoras y planificadores de vuelo.

Código Libre	Nombre de Empresa/ Software	Descripción	Lenguaje de Programación	Modelos/ Versiones
Controladora de Vuelo y Firmware	<i>ArduPilot</i>	<i>Hardware y software libre.</i>	C++	<i>ArduPilot, APM 1, APM 2, APM 2.5/2.6, Pixhawk Actual V3.6.10</i>
	<i>PX4 Firmware</i>	<i>Software libre.</i>	C++	Actual V3.2.1
Estación de control de tierra	<i>Mission Planner</i>	<i>Software libre.</i>	C#	Actual V1.3.68
	<i>APM Planner</i>	<i>Software libre</i>	C++	Actual V2.7.1
	<i>QGroundControl</i>	<i>Software libre</i>	C++	Actual V3.5.4

Tabla 1.2. Software libre para controladoras y planificadores de vuelo.

El mercado comercial está cubierto por empresas nacionales e internacionales, entre las nacionales que ofrecen servicios de fumigación están: ECOBIS, AGROSCAN, DRON CROP y AGRONEC; también existen empresas nacionales que se dedican al campo agrícola con drones, pero solo al análisis de los cultivos por medio de la toma de imágenes aéreas: TOPDRONES, AGROPRECIÓN, ECOBIS, AGROSCAN, GEOKONCEPT, AGROSOFT y LAS; esta última marca una gran diferencia al crear sus propios drones, mientras que el resto usa drones de la marca comercial DJI para ofrecer sus servicios.

Entre las internacionales sobresalen los países de China, España, Estados Unidos y Suiza; por ejemplo; dedicadas a la fumigación: ASTA, empresa de China; empresas dedicadas al procesamiento de imágenes aéreas: GEOTEDICS empresa de Estados Unidos, SENSEFLY empresa de Suiza, muy conocida por crear drones de ala fija con tiempo de vuelo de hasta 90 minutos.

1.4.4 Situación actual del arroz en Ecuador y el mundo

El arroz es el segundo producto con mayor extensión de cultivos a nivel nacional y el segundo cereal más consumido a nivel mundial tras el maíz, según La Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) [8]. Solo en la región costa, en Guayas y Los Ríos se concentra el 95% de la superficie cosechada de todo el país, ver Tabla 1.3. La superficie con labor agropecuaria en 2018 abarcó 5.3 millones de hectáreas, con el 15.1% en cultivos transitorios y barbechos, de donde el arroz alcanza el 32.1% de la superficie total sembrada a nivel nacional, por debajo de los cultivos de maíz duro seco con el 40.7%, contrario a las cifras de 2016 donde el arroz se mantenía superior. En vista de que, en ese año su producción alcanzó 1.534,537 toneladas métricas, pero al siguiente registró un decrecimiento del 30.5%. De todas formas, en 2018 con 1.350,093 toneladas métricas no superó la producción de arroz de 2016 [9].

Provincias	Superficie Total (Ha.)		Producción Nacional (Tm.)
	Sembrada	Cosechada	
Guayas	69.7%	69%	71%
Los Ríos	25.7%	26%	23%

Tabla 1.3. Porcentajes de la superficie y producción total sembrada y cosechada de cultivos de arroz en Guayas y Los Ríos en el 2018 [9], [10].

1.4.5 Mejoras en la fumigación agrícola con el uso de drones

Si bien es cierto, los drones actualmente pueden abarcar entre 40 y 50 hectáreas de fumigación al día, con cerca de 10 horas de vuelos. Lo cual no puede competir con las 200 a 400 hectáreas que una avioneta o helicóptero puede cubrir en un par de horas. Pero cabe destacar que, un dron gana en

eficiencia con respecto al uso de plaguicidas, que sin utilizar demasiada agua y con atomización de bajo volumen, rocía lo suficiente las hojas del cultivo, puesto que puede volar a escasos metros de altura sobre las plantaciones, sin afectar el cultivo ni desperdiciar el líquido. Lo opuesto, genera elevados costos de producción, afectaciones al hombre y al medio ambiente, lo que ya provocan los métodos tradicionales de fumigación, ver Tabla 1.4.

Indicadores	Drones de Fumigación	Métodos de Fumigación Tradicionales		
		Bomba de Fumigación Manual	Tractores elevados de fumigación	Helicópteros y Avionetas de Fumigación
Eficiencia	40-53 ha/día	0.8 – 1.3 ha/día	66.7 - 80 ha/día	200 - 400 ha /día
Adaptabilidad	Alta	Baja	Baja	Media/Baja
Eficiencia en uso de pesticidas	> 85%	30%	70%	85%
Consumo de agua por hectárea	Atomización de bajo volumen	Más uso de agua por hectárea	Líquido desperdiciado en el suelo	Bajo volumen, pero aumenta área de fumigación
Seguridad	Sin riesgo laboral	Alto riesgo de intoxicación	Riesgo ocasional de intoxicación	Alto riesgo de accidente aéreo

Tabla 1.4. Comparación de métodos tradicionales de fumigación vs. dron fumigador [3].

CAPÍTULO II

2. METODOLOGÍA Y SOLUCIÓN

En esta sección se describe la metodología de desarrollo del sistema a diseñar, especificando los diferentes niveles de implementación para la realización del

proyecto. Luego se detallarán los requerimientos de hardware y software, y los cambios que se propondrán en el mismo. Finalmente, se explicará cómo llevar a cabo el diseño y ejecución de la solución con el uso de tecnologías de código abierto.

2.1 Metodología

Para el desarrollo del presente proyecto se utilizó la metodología ágil *Kanban*, implementada en sus inicios como una técnica de los sistemas de producción en *Toyota*, y adaptada a la ingeniería de software a finales del siglo XX [11]. Cuyos resultados son altamente positivos, puesto que mantiene un flujo constante de la información, limita el trabajo en progreso y ayuda a visualizar problemas en el proceso de desarrollo [12]. Además, *Kanban* es una palabra japonesa que significa letrero [13], por lo que es necesario utilizar un tablero físico donde se coloquen tarjetas visuales por cada actividad para cumplir un básico ciclo de vida de desarrollo de software.

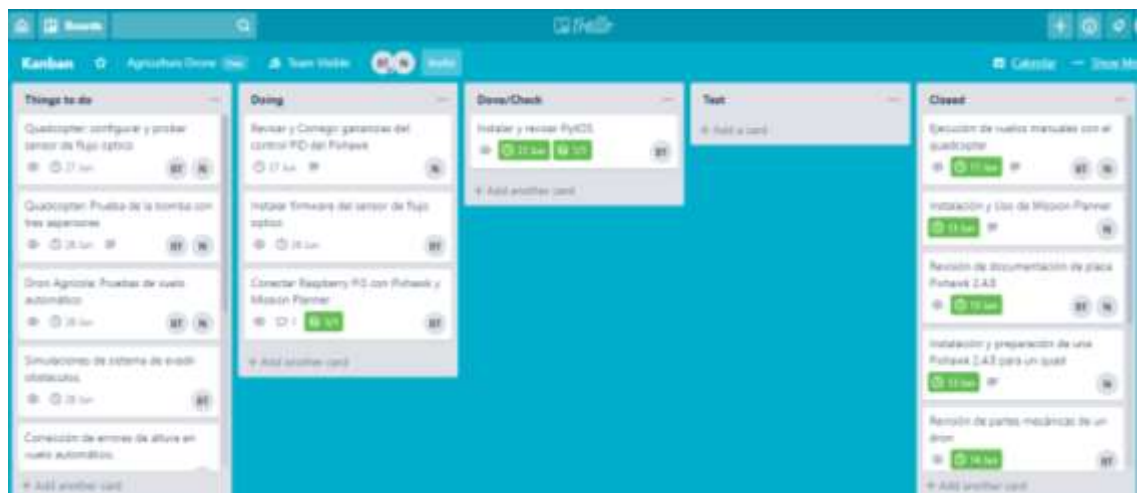


Figura 2.1. Panel web de Trello con el flujo de trabajo del tablero Kanban

2.1.1 Método de investigación

Durante todo el proceso de diseño y posterior desarrollo de la solución se persiguió un método analítico-sintético, ya que, se partió de la descomposición de un sistema integral en cada uno de sus subsistemas para estudiarlos de forma individual (análisis), que luego fueron anexados para estudiarlos con un enfoque holístico (síntesis) [14].

De esta manera la presente investigación cubre tanto el análisis de los requerimientos de hardware y software fundamentales, como el diseño y desarrollo de la solución sobre estas bases tecnológicas, las cuales serán descritas en una sección posterior.

2.1.2 Modelo de desarrollo de software

Para el caso específico del uso de técnicas de ingeniería de *software*, se utilizó un modelo de desarrollo incremental, el cual se basa en entregas escalables conocidas como incrementos que describen funcionalidades del sistema que el usuario puede validar [15]. Cada incremento fue planificado sobre el tablero *Kanban* organizado por columnas. Tuvimos así actividades fluyendo en las siguientes columnas: cosas por hacer, haciendo, listo para revisión, testeo, simulación y cierre.

Entonces, en vías de digitalizar el proceso y visualizar desde cualquier lugar el flujo de trabajo se utilizó una herramienta de colaboración llamada *Trello*¹, la cual opera como un organizador de proyectos que emplean metodologías ágiles como *Kanban*, ver Figura 2.1. Desde ahí se monitoreaba el estado de cada actividad, planificada con fecha y hora de finalización a un miembro del equipo de trabajo.

2.2 Requerimientos de hardware

Para el desarrollo de la solución se prescindió del uso de piezas de hardware, en su mayoría de código abierto, que permitieron modelar la arquitectura del sistema final. A continuación, se describen cada uno de los componentes físicos y electrónicos con especial detalle en las características que fueron más relevantes para el proyecto.

2.2.1 Autopiloto *Pixhawk*

Pixhawk [16] es una controladora de vuelo, del inglés *Flight Control* (FC) que proporciona hardware abierto como estándar para modelos de pilotos automáticos de alta calidad y de bajo costo, aprovechado por comunidades

¹ <https://trello.com/en>

académicas, aficionados y desarrolladores. Combina una unidad de administración de vuelo, del inglés *Flight Management Unit* (FMU), y la placa *PX4I/O* en un solo componente. Además, utiliza procesadores de 32 bits *ARM CortexM4* de alto rendimiento, ejecutando el sistema operativo *Nutts RTOS*, y soporta tanto la pila de vuelo *PX4* en su instalación mediante *QGroundControl*, como la pila de vuelo *APM* en su instalación por medio de *Mission Planner* o *APM Planner*, todas estaciones de control en tierra, del inglés *Ground Control Station* (GCS).

El kit *Pixhawk 2.4.8*, que corresponde a la serie *mRo Pixhawk*², viene con un zumbador para emitir sonidos de estado, un botón de seguridad para permitir el armado del vehículo, y una tarjeta *microSD* que almacena datos del vuelo. Otras características principales son: Microprocesador de 32 bits *STM32F427 Cortex M4* con frecuencia de 168MHz y 256K de RAM, giroscopio digital de 3 ejes *L3GD20*, acelerómetro/magnetómetro de 3 ejes *LSM303D*, acelerómetro/magnetómetro de 6 ejes *MPU6000*, barómetro de precisión *MS5611*. Las interfaces disponibles más importantes son UART, CAN, I2C, entrada para micro USB externo, entrada compatible con receptor de señal PPM y satelital *Spektrum*, entrada y salida compatible con *Futaba S-BUS*, y pines para salida de señales PWM [17].

2.2.2 Componentes electrónicos

Para armar el sistema RC por radiofrecuencia se utilizó el receptor (Rx) *RadioLink R12DS* que opera en la banda de 2.4 para comunicación con el transmisor (Tx); el enlace con la FC se puede establecer de dos formas, con PWM de 11 canales o con S-BUS de 12 canales [18]. Además, este Rx es compatible con transmisores *RadioLink*: *AT9*, *AT9S*, *AT10* y *AT10 II*; de donde se escogió el Tx *AT9S* habilitado para salidas de 12 canales. Este transmisor maneja dos tipos de espectro ensanchado DSSS y FHSS con una frecuencia de 2.4GHz, y corriente de operación de 90mA a 12V en promedio [19].

En el caso de la conexión por telemetría entre la FC y una GCS, utilizamos el kit de telemetría FPV, el cual opera en la banda de 915MHz, con velocidades

² https://docs.px4.io/en/flight_controller/mro_pixhawk.html

de datos en aire de hasta 250Kbps y rango aproximado de 1mi con antenas suministradas [20]. El módulo en tierra fue conectado al GSC por USB, mientras que el módulo en aire fue conectado al puerto UART de *TELEM 1* de la FC.

A su vez, para dotar al dron con datos de posicionamiento global y poder realizar vuelos autónomos, utilizamos el módulo GPS U-blox M8N con exactitud de 2m, conectado al puerto GPS e I2C del FC.

Por otro lado, utilizamos motores sin escobillas conectados a controladores electrónicos de velocidad, del inglés *Electronic Speed Controller* (ESCs), de la marca *FlyColor* modelo *FlyDragon Opto* que operan a 100A y 50.4V para baterías de Litio Polímero (LiPo) de 5 a 12 celdas (5-12S). Ver Figura 2.2.

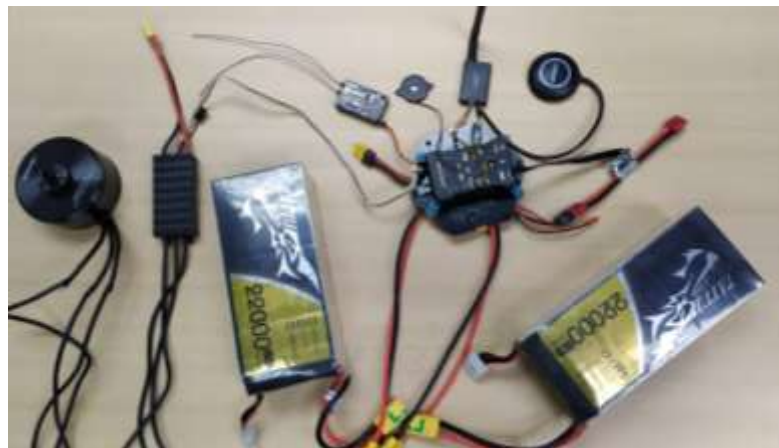


Figura 2.2. Muestra de componentes electrónicos conectados al Pixhawk

2.4.8

2.2.3 Componentes mecánicos

- 4 baterías LiPo de 6 celdas (6S) de 22.2V que operan a 22000mAh y carga/descarga a 25C.
- 6 brazos y 3 parejas de hélices en fibra de carbono
- Patas con tubos de aluminio de 19mm de diámetro y 2.5mm de espesor.

2.2.4 Componentes para el sistema de fumigación y fertilización

- 5 boquillas para fumigación
- Tanque plástico de 15lt para plaguicidas
- Tanque plástico de 15lb para fertilizantes sólidos
- Motor de bomba que opera a 12V

- Motor de aspersión que opera a 12V
- Dispensador de sólidos que opera a 24V
- Microcontrolador de la marca *Black Wings* modelo HC2412 manufacturado en China, con entrada de 17-32V a 5A, salida de 12V a 60W y entrada de señal de control por cable tipo Circuito de Eliminación de Batería, del inglés *Battery Elimination Circuit* (BEC), el cual fue conectado a la salida auxiliar 1 (*AUX OUT 1*) del Pixhawk para enviar por ese medio señales PWM al microcontrolador y encender o apagar el motor de la bomba.

2.2.5 Estructura de montaje

A la fecha de la realización del proyecto, el CIDIS contaba con un dron comercial para uso agrícola manufacturado en China, que incluía todo el sistema de control de vuelo para la agricultura *N3-AG*³, como un típico *DJI Agras MG-1*⁴. Sin embargo, el resto de las piezas mecánicas y electrónicas como los brazos, ESCs, tanque de fumigación, motor de bomba y aspersores, no correspondían a ningún modelo *DJI* específico. En todo caso, el centro de investigación realizó su modelamiento 3D para un estudio de análisis mecánico que nada tienen que ver con la realización del presente proyecto, pero a modo de muestra se puede apreciar una vista panorámica del modelo en la Figura 2.3.

Entonces, se desmontó el sistema *N3-AG* y en su lugar se colocó la *Pixhawk* conectada al Rx, el módulo *GPS Ublox M8N*, el módulo de telemetría en aire y el zumbador incluido en el kit de la FC, con la finalidad de utilizar sólo hardware libre sobre la estructura del dron provista.

³ <https://www.dji.com/agriculture-solution>

⁴ <https://www.dji.com/mg-1?site=brandsite&from=nav>



Figura 2.3. Vista panorámica del modelo 3D de un dron comercial

2.2.6 Raspberry Pi 3

Por último, se utilizó la conocida SBC a bordo *Raspberry Pi 3* modelo B versión 1.2, conectada al puerto *TELEM 2* de la FC por medio de un cable TTL a *UART* desde los pines de la *GPIO*, ver Figura 2.4. Esta SBC debía permitir ejecutar un *script* en *Python* después de cada vuelo, con la intención de almacenar los registros del vuelo en la base de datos de un servidor remoto.



Figura 2.4. Raspberry Pi 3 model B V1.2

2.3 Recursos de software

En su mayoría se utilizó *software* de código abierto para el control, planificación, simulaciones y la base de datos; y unos cuantos sistemas comerciales en su versión gratuita, como por ejemplo *Microsoft Visual Studio Community 2017*, entre los otros sistemas comerciales que se usaron fue: *PostgreSQL*.

2.3.1 ArduPilot

Proyecto de software y firmware del mismo nombre, creado en el año 2007 para vehículos no tripulados: drones, carros, helicópteros, avionetas, botes y submarinos de piloto automático, así también para rastreadores de antena. Actualmente la comunidad de desarrollo de este proyecto ofrece actualizaciones continuas, un código optimizado [20] y están encargados de otros diversos proyectos también de código abierto, como las estaciones de control de tierra: *Mission Planner* y *APM Planner*, el sistema operativo *ChibiOS*, el firmware *PX4Firmware*, entre otros, como lo podemos ver en su repositorio [21].

2.3.2 Mission Planner

Actualmente una de las estaciones de control de tierra más completas y compatible con los sistemas operativos: *Windows*, *Mac OS X* y *Linux*, creada por Michael Osborne, para controladoras de vuelo que integren el proyecto de *software* libre *Ardupilot*. Esta estación de control de tierra solo se queda a atrás del *software* Estación de Control de Tierra Universal o más conocido por su acrónimo en inglés UgCS, el cual permite la conexión de controladoras comerciales y de código libre.

Entre las características más destacables de *Mission Planner* se encuentran:

- Planificación de rutas.
- Calibrar y cambiar parámetros de controladora de vuelo y de todos los elementos conectados a la misma.
- Veintiún modos distintos de vuelo.
- Descargar y analizar los registros de vuelo.
- Simulación de distintos vehículos por medio del SITL: drones multirrotores, de ala fija, helicópteros y carros.

2.3.3 Software en el bucle: SITL

Simulador que permite ejecutar nativamente *Ardupilot* en cualquiera de los modelos de vehículos de *Plane*, *Rover*, *Multirrotor* o *Helicopter*, sobre *Mission Planner* sin necesidad de ningún hardware. Hace uso de un compilador del

lenguaje de programación C++, en la Figura. 2.5 podemos apreciar su arquitectura [22].

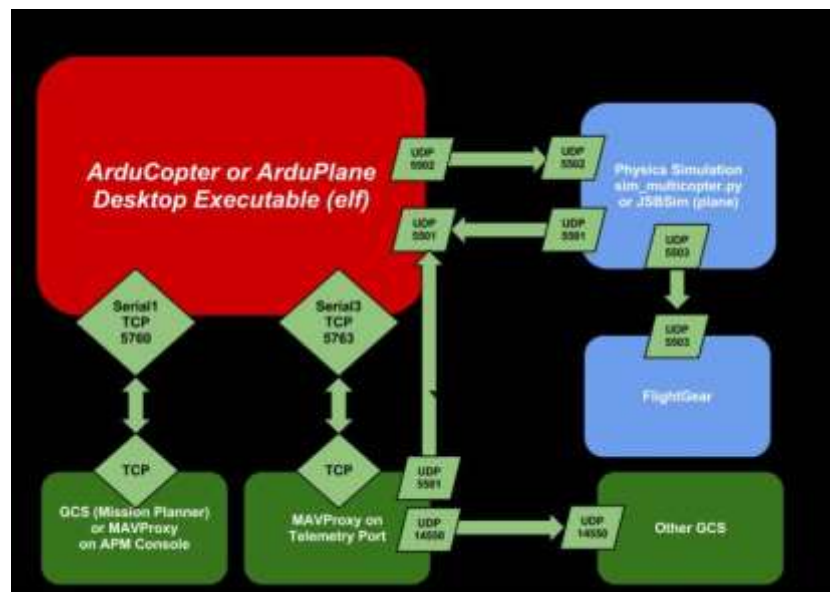


Figura 2.5. Arquitectura SITL [22]

2.3.4 Microsoft Visual Studio Community 2017

La versión gratuita del entorno de desarrollo integrado que puede ser ejecutado en *Windows*, creado por *Microsoft*, permite el desarrollo de aplicaciones multiplataforma de escritorio, webs, móviles y en la nube, conexión con bases de datos y rápida depuración del código; por medio de *JavaScript*, *ASP.NET*, *Node.js*, *TypeScript*, *Python*, los marcos de trabajo *Express*, *Angular*, *jQuery*, entre otros [23] y la plataforma *.NET*.

2.3.5 Microsoft .NET

Plataforma de desarrollo de software de código abierto, que permite crear aplicaciones con los lenguajes de programación *C#*, *F#* o *Visual Basic*, ejecución del código en forma nativa [24] y sobre todo permite el uso del paquete gratuito, multiplataforma para el uso de mapas *GMAP.NET*.

2.3.6 GMAP.NET

Componente de código abierto de la plataforma *.NET* que permite la creación de marcadores y polígonos; el enrutamiento de direcciones y mapas de

Google, Bing, Yahoo!, CloudMade, ArcGIS, Maps.it, WikiMapia, entre otros, para implementarlos en las aplicaciones de escritorio, webs y móviles.

2.3.7 PostgreSQL

Sistema gestor de base de datos relacional de código libre altamente extensible y basado en estándares técnicos. Con su herramienta visual por defecto, *pgAdmin 4*, se puede crear, diseñar y administrar bases de datos relacionales basadas en el lenguaje de consulta estructurada, del acrónimo inglés *SQL*. Además, ofrece la administración de los usuarios, la configuración del servidor y es compatible con los sistemas operativos: *Windows, Mac OS X y Linux*.

2.4 Diseño de la solución

La solución fue diseñada sobre el código fuente de *Mission Planner* a manera de modificación para satisfacer las necesidades del proyecto, en conjunto con simulaciones de prueba con SITL antes de ejecutar vuelos autónomos que fueron realizados primero con un dron de 4 rotores pequeño el cual en este proyecto fue denominado “dron *small*” como se puede ver en la Figura 2.6 y segundo con el dron real el cual fue denominado “dron agrícola multifuncional” como se puede ver en la Figura 2.7 para cumplir operaciones de fumigación y aspersión de fertilizantes sobre cultivos de arroz, a dos metros de altura puesto que el arroz es un cultivo que crece hasta máximo 1.5 metros.



Figura 2.6. Dron *small*



Figura 2.7. Dron agrícola multifuncional

A continuación, se detalla un esquema general de la función del hardware y software descritos en la sección anterior, como punto de partida para luego describir cada parte del sistema integral de planificación de rutas de vuelo. Tenemos así que:

- El dron agrícola lleva a bordo la *Pixhawk 2.4.8* y la *Raspberry Pi 3* conectadas desde el GPIO del SBC al puerto *TELEM 2* del FC, para establecer conexión a través del protocolo *MAVLink*.
- El FC será conectado a *Mission Planner* también por medio del protocolo *MavLink*, y con el uso de la pila de vuelo APM desarrollados todos por *ArduPilot*.
- El control del motor de la bomba para fumigación estará conectado al FC, quien encenderá o apagará el motor por medio del envío de señales PWM.

En la Figura 2.8 se observa el esquema general planteado para la solución del proyecto, en donde se describen 2 módulos diferenciados. El primero que trata sobre el control y planificación de rutas de vuelo se encuentra plenamente implementado por *ArduPilot* en el planificador de vuelo *Mission Planner*, pero por requisitos extras para el sistema final se modificará dicho código fuente. El segundo módulo se encarga de la adquisición de datos correspondientes a cada vuelo para su posterior almacenamiento en la base de datos de un servidor remoto para uso del CIDIS.

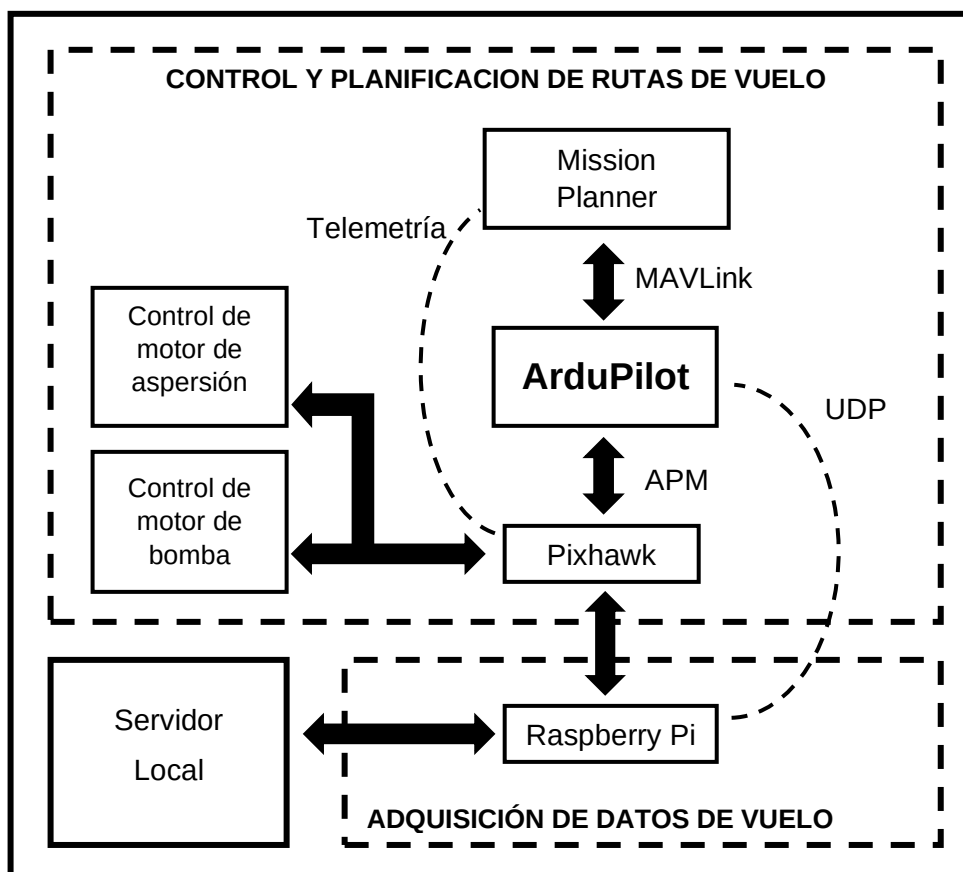


Figura 2.8. Esquema general de la solución del proyecto

Sin embargo, para el sistema integral de planificación de rutas de vuelo el módulo de adquisición de datos de vuelo es el inicio de futuros trabajos de investigación a partir del estudio de dichos datos. Es decir, el problema principal a resolver se centra en el módulo de control y planificación de rutas de vuelo sobre cultivos de arroz, el cual debe:

- Activar o desactivar el circuito controlador que acciona el motor de la bomba de fumigación o el mecanismo para esparcir fertilizantes sólidos,
- Mantener la altura deseada del dron agrícola durante un vuelo autónomo,
- Marcar un obstáculo detectado sobre una ruta de vuelo planificada,
- Evitar dicho obstáculo mediante la reestructuración del plan de vuelo antes de realizarse, y
- Continuar con el plan de vuelo trazado desde el comienzo, ver Figura 2.9.

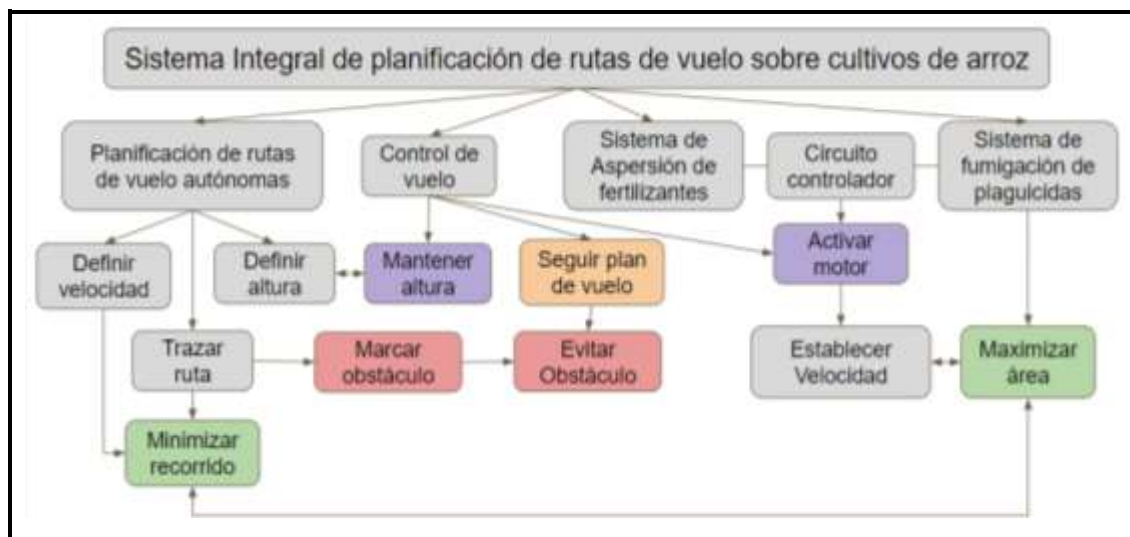


Figura 2.9. Desglose de funciones que el sistema integral de planificación de rutas de vuelo debe cumplir.

En general, el diseño final de esta metodología incluyó la implementación de un dron agrícola multifuncional a manera de prototipo, que contempla las características descritas en la Tabla 2.1, y consideró la siguiente hipótesis:

- Durante un vuelo de 20 minutos a 4m/s, el dron debe cubrirá 2 hectáreas a 2 metros de altura.

Parámetro	Valor
Autonomía de vuelo mínima	20min
Altura máxima	2m
Velocidad máxima	4m/s
Cobertura máxima	2ha
Alcance mínimo de la fumigación	6m
Alcance máximo de la aspersión de fertilizantes	10m

Tabla 2.1. Requisitos del prototipo final

2.4.1 Mantener altura

El controlador proporcional, integral y derivativo (PID) utilizado en procesos industriales con circuitos de retroalimentación, garantiza un alto rendimiento del control de actitud de un UAV en general. Para lo cual, se deben ajustar los correspondientes parámetros de control también conocidos como ganancias.

De ahí que, con la ganancia proporcional se puede disminuir el tiempo que le toma a la nave alcanzar la altura deseada; mientras que la ganancia derivativa es utilizada para amortiguar el exceso de impulso y alcanzar una pronta estabilización; y, la ganancia integral permite eliminar errores de estado estacionario.

El correcto ajuste de los parámetros de control asegura un buen rendimiento del controlador PID. Sin embargo, este procedimiento, conocido como sintonización y descrito en la documentación de *ArduPilot* [25], es un verdadero cuello de botella debido a que, el ajuste de los parámetros por lo general depende de la experiencia de un operador tras varias pruebas en modos de vuelo manuales [26].

Por lo tanto, se han realizado varios estudios para obtener el modelo del sistema de un UAV a partir de técnicas de identificación del sistema, empleando para tal propósito redes neuronales [26], lógica difusa [27] y el barrido de la frecuencia [28], [29]. Así que, para el presente proyecto se empleó la metodología de [29] debido a la complejidad de las demás y que esta fue la más accesible.

2.4.2 Esquivar obstáculos

Se realizó la compilación del código de *Mission Planner* en *Visual Studio Community 2017*, sobre el sistema operativo *Windows 10*, para implementar la marcación de obstáculos y la planificación de la ruta de vuelo alrededor del mismo, y así lograr un vuelo automático esquivando obstáculos previamente detectados.

2.4.3 Simulación de pruebas de vuelos críticos

Para la simulación de vuelos autónomos, ajustes del controlador PID, y prueba del mecanismo para marcar y evitar obstáculos se hizo uso de simulación SITL en *Windows*. En concreto, una instancia de *ArduCopter* fue lanzada desde el simulador y se conectó por UDP al planificador de vuelos *Mission Planner*, cumpliendo con lo descrito en la documentación de *ArduPilot* [30]. De esta manera, se probaron escenarios críticos y específicos que el dron agrícola

debía cumplir para garantizar su operación automática sin contratiempos en el campo. Así, fueron desarrolladas las siguientes actividades:

- Simulación de vuelos de rutas automáticas con especificaciones de tiempo de vuelo, velocidad y área de alcance máximo.
- Simulación de vuelos para obtener el barrido de la frecuencia.
- Simulación de comprobación de la actitud de vuelo tras ajuste del controlador PID.
- Simulación de planes de vuelo con marcación de obstáculos

2.4.4 Adquisición de datos de vuelo

Al final de cada vuelo un script en *Python* implementado en la *Raspberry Pi 3* debía tomar los siguientes datos del vuelo:

- Parámetros del FC
- Plan de vuelo
- Registros de vuelo
- Fecha, hora, duración y área recorrida del vuelo

Los cuáles son enviados al servidor donde se aloja una base de datos local de *PostgreSQL*.

2.5 Plan de ejecución de vuelos

Para llevar a cabo la evaluación de rutas de vuelo, fue necesario realizar un plan de actividades de vuelos, con distintos tipos de pruebas a realizar para garantizar que el dron agrícola multifuncional opere de forma automática sin contratiempos para el cliente final. Teniendo en cuenta que en una sección anterior ya se detallaron las actividades realizadas con simulación SITL. Ahora se describirán las pruebas de vuelos ejecutadas primero con el dron *small* de 4 rotores (ver Tabla 2.2), y luego las pruebas de vuelos del dron agrícola multifuncional que posee 6 rotores, ver Tabla 2.3.

Pruebas de vuelo para el dron <i>small</i>	
Actividad	Número de vuelos
Vuelo con radio control y altura variable	4
Despegue y aterrizaje a 2m de altura	4
Ruta automática con puntos referenciales	4
Ruta automática con área definida	5
Ruta automática prolongada o cíclica para medir tiempo de vuelo	2
Ruta automática a prueba de fallos de desconexión de radio control	2
Ruta automática a prueba de fallos de desconexión estación de control de tierra	2
Ruta automática a prueba de fallos de la batería	2
Ruta automática para evitar obstáculos	0
Total	25

Tabla 2.2. Planificación de vuelos para un dron *small*.

Pruebas de vuelo Para el dron multifuncional			
Actividad	Sin carga	Con sistema de fumigación y sistema de aspersión	Total
Límite de altura a 2 metros.	2	1	3
Límite de altura a 4 metros.	2	1	3
De un punto a otro punto.	2	2	4
Ruta automática a prueba de fallos de desconexión de radio control.	2	1	3
Ruta automática a prueba de fallos de desconexión estación de control de tierra (<i>Mission Planner</i>).	2	1	3
Ruta automática de área pequeña.	2	1	3
Ruta automática prolongada o cíclica para medir tiempo de vuelo.	2	1	3
Ruta automática a prueba de fallos de baterías.	2	1	3
Ruta automática con corrección de pérdida de altura.	2	2	4
Ruta automática con esquivar un obstáculo marcado.	2	2	4
Ruta automática de una hectárea.	2	2	4
Total	24	16	40

Tabla 2.3. Planificación de pruebas de vuelo.

CAPÍTULO III

3. IMPLEMENTACIÓN DE LA SOLUCIÓN

En esta sección se explica el desarrollo de los componentes del sistema propuesto, de acuerdo con la metodología presentada en el capítulo anterior. De ahí que, empieza con la integración de los sistemas de fumigación y aspersión de fertilizantes para automatizar su funcionamiento, tras la puesta en marcha del dron agrícola. Luego, describe el mecanismo de ajuste del control PID, adecuado para drones que deben garantizar estabilidad durante el vuelo. A su vez, presenta la funcionalidad de marcar zonas para evadir los obstáculos presentes en ellas, durante la planificación de rutas. Finalmente se ejecutaron dichas rutas de vuelo fundamentadas en la necesidad de testear los sistemas y funciones implementados.

3.1 Instalación de hardware libre

Primero que todo se debe hacer mención del cambio de hardware comercial por electrónica de bajo costo y de hardware libre, como la controladora de vuelo *Pixhawk*. El cambio busca mostrar que es posible diseñar e implementar una solución menos costosa de drones para uso agrícola, que además es posible añadir nuevas funcionalidades, de acuerdo con las necesidades que apremian en el mercado. Por lo cual, En las siguientes secciones se detalla la implementación del hardware y software específico para un dron agrícola que debe fumigar y fertilizar sembríos de arroz a 2 metros de distancia sobre el suelo.

3.1.1 Configuración

De manera general se realizaron varias configuraciones sobre el hardware tanto para alimentación como para establecer comunicación entre ellos. Así, se editaron parámetros tras instalación del firmware de *ArduCopter 3.6.9* para el *Pixhawk* alimentado por cable USB. Por otra parte, se precisó de la conexión de una PDB a los ESCs y al *Pixhawk* mediante un *Power Module* con 30V de entrada y 5V de salida. Además, como respaldo por alguna caída demasiado brusca de voltaje, una batería extra de 3S 2200mAh fue conectada a un BEC y

este a un par de pines de alimentación y tierra (positivo y negativo) del carril de los servomotores, ya sean las salidas principales (*MAIN OUT*) o las auxiliares (*AUX OUT*) del carril.

Elegido Hexa como tipo de marco (parámetro *FRAME_TYPE*). De ahí mismo, se precisó el sentido de giro de los motores, ver Figura 3.1.

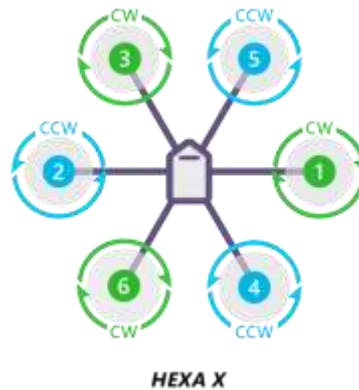


Figura 3.1. Diagrama de orden de los motores en configuración *Multirotor en x Hexa*. CW: giro en sentido horario. CCW: giro en sentido anti-horario.

El radio control *AT9S* debía estar habilitado para 12 canales antes de emparejar con receptor *R12D*, el cual soporta 12 canales. De lo contrario, se emparejan de forma fallida porque el receptor no recibe ninguna señal desde el radio control.

A su vez, se conectó la *Raspberry Pi 3* al *Pixhawk* con una configuración distinta a la prevista, como lo propone [31], debido a que no se solucionaron diferentes errores tras conexión desde la GPIO del SBC al puerto *TELEM 2* del FC. Por tal motivo, se utilizó en su lugar un cable USB conectado desde un puerto USB de la *Raspberry Pi* hacia la entrada micro-USB del *Pixhawk*. Para esto, debimos conectar vía *MAVlink* en 115200 baudios en lugar de 57600 como estaba previsto para la primera configuración. Además, el puerto de comunicación pasó de ser *ttyAMA0* a *ttyACM0*. De esta manera, como usuario *root* ejecutamos *mavproxy.py* con los siguientes parámetros:

```
mavproxy.py --master=/dev/ttyACM0--baudrate 115200 --aircraft dron-ag
```

3.1.2 Calibración

Todas las calibraciones que fueron realizadas incluyen: acelerómetro, compás, ESCs, radio control y módulo de poder. Para lo cual se siguieron las siguientes pautas:

- Antes de la calibración del compás fue necesario asegurarse de que tanto el compás del GPS y el compás interno del FC hayan sido reconocidos y habilitados. En realidad, se pueden calibrar hasta tres compases (uno interno y dos externos) o incluso solo uno (externo) si acaso el compás interno tuviere problemas para asimilar el campo magnético generado por el paso de la corriente desde las baterías, que son de gran tamaño en este tipo de drones. Luego de la calibración se evidenció el nuevo ajuste de las compensaciones para el campo magnético registrado en esa zona de la tierra.
- Se realizaron calibraciones del acelerómetro y compás antes de cada vuelo puesto que las condiciones geográficas cambian en diferentes sitios.
- Se volvió a realizar la calibración de los ESCs cada cuando se evidenciaba un desbalance en la potencia de los motores. Previo a esto se calibró el radio control.

3.1.3 Puesta en marcha

Se elaboró una lista de verificación (*CheckList*) de todas las herramientas necesarias, conexiones del dron en buen estado, configuraciones y calibraciones correctas, antes de ejecutar cualquier plan de vuelo, la cual está descrita en el Anexo 1.

Además, se implementó un protocolo de seguridad para realizar vuelos dentro y fuera el campus universitario, que incluye los siguientes puntos:

- Las pruebas de vuelo se deben realizar en espacios abiertos y sin la presencia de civiles ajenos a las actividades del presente proyecto.
- Coordinar con a las autoridades de Seguridad Ocupacional y Seguridad Laboral de la institución, la fecha y hora en que se realicen los vuelos, en caso de utilizar las instalaciones de la institución.
- Hacer uso de cascos, guantes anticorte y gafas.

- Operar el dron desde una distancia mínima de 20m desde la ubicación de despegue y aterrizaje del dron.

3.2 Integración de sistemas

Tanto el sistema de fumigación como el de fertilización incluyeron el mismo circuito microcontrolador que, mediante el envío de pulsos PWM permitió accionar los motores de ambos sistemas, a través de un cable conectado a una de las salidas auxiliares del *Pixhawk (AUX OUT)*. El procedimiento implementado se detalla a continuación:

Para probar el funcionamiento del microcontrolador, se procedió a utilizar los siguientes elementos:

- Microcontrolador con 17-32V de entrada y 12V de salida, con cable para recibir señal PWM.
- Fuente de voltaje
- *Arduino UNO r3*
- Aplicación ejecutable de *Arduino* para *Windows 10*
- Cables puente
- Multímetro digital

El procedimiento seguido para conocer el voltaje de operación del motor de la bomba fue el siguiente:

- Ajustar a 17.5V fuente de voltaje en corriente continua.
- Conectar puente entre BEC de microcontrolador y placa *Arduino*. Cable de señal fue colocado en la salida 11.
- Escritura de código sobre aplicación *Arduino*.
 - Se asignó a pin 11 como de tipo salida.
 - El incremento cada 3 segundos de una variable tipo *byte* es escrito a la salida analógica del pin 11.
- Conexión de fuente de poder a entrada de microcontrolador
- Conexión de placa *Arduino Uno* al computador.

- Ejecución del código anterior sobre placa *Arduino*.
- Medir el voltaje de salida del microcontrolador con un multímetro.
 - o A partir de las lecturas PWM 144 hasta 254, se mantenía el voltaje en 12.49V, y el microcontrolador encendido
 - o Mientras que para valores de 0 a 143, y 255, el voltaje caía a 0 y el microcontrolador se apagaba.

3.2.1 Sistema de fumigación de plaguicidas

Un sistema básico de fumigación se compone de un motor de bomba de riego que se encarga de hacer circular el líquido con la suficiente presión para que los aspersores rocíen agua de manera homogénea. Para esto, fue necesario que el microcontrolador con 12V de salida, antes probado, fuera conectado directo con el motor de la bomba de 7-14V de entrada, en lugar del multímetro; se comprobó que la velocidad de operación del motor era proporcional a la variación de los pulsos recibidos cada 3 segundos.

Luego, se procedió a configurar la fumigación con motor de la bomba desde el GCS conectado al FC, de la siguiente manera:

- En *Mission Planner* -> *Config/Tuning* -> *Full Parameter List* se editaron los siguientes parámetros

<i>SPRAY_ENABLE</i>	<i>1</i>
<i>SERVO9_FUNCTION</i>	<i>22</i> (definida para salida PWM en <i>AUX OUT 1</i>)
<i>CH7_OPT</i>	<i>15</i> (habilita el <i>spray</i> desde entrada de señal RC en el canal 7)
<i>SERVO9_RESERVED</i>	<i>1</i> (de lo contrario señal de salida disminuye)
<i>SERVO9_MAX</i>	<i>2200</i> (PWM máximo disponible)
<i>SERVO9_MIN</i>	<i>1100</i> (PWM mínimo disponible)
<i>SERVO9_TRIM</i>	<i>1200</i> (Recortador de la señal, por defecto corresponde al valor central del rango entre el PWM mínimo y máximo.)

- En *Mission Planner* -> *Flight Data* -> *Quick* se activaron las siguientes vistas:
 - *ch7in*: Para observar señal de entrada del canal 7 desde el radio control.
 - *ch9out*: Para observar señal de salida al controlador del motor de la bomba.

3.2.2 Sistema de aspersión de fertilizantes

El sistema de aspersión propuesto por tesistas de la Facultad de Ingeniería Mecánica y Ciencias de la Producción (FIMCP), consistía en dos motores: un dosificador de 24V para permitir o denegar el paso al fertilizante y un motor de 12V para esparcirlo. De modo que, el funcionamiento de ambos debía de ser paralelo e independiente uno del otro. Por tal motivo, se procedió a realizar los siguientes ajustes:

- Conexión en paralelo de la alimentación de los motores a la salida del microcontrolador.
- Uso de la misma configuración en *Mission Planner* utilizada para activar la bomba. Es decir, el microcontrolador solo debe ser desconectado del motor de bomba para conectar los motores de fertilización, sin alterar ningún parámetro ni vistas del GCS, como la entrada del canal 7 o la salida hacia el servomotor 9.

3.3 Identificación del sistema de respuesta de frecuencia

A continuación, se describe la implementación realizada para llevar a cabo el ajuste del control PID del FC, mediante la metodología de identificación del sistema.

El barrido de la frecuencia para el cabeceo (*pitch*), balanceo (*roll*) y guiñada (*yaw*), se obtuvieron tras simular vuelos en modo *Guided* para un dron de 4 rotores. En general, se implementó un script en *Python* con uso de librerías de *Dronkit* para controlar la actitud de vuelo del *ArduPilot* enviando comandos a través de una instancia de *MAVProxy*. De esta manera, se obtuvo una respuesta

dinámica por cada señal. Acorde con [29] y [28] se hizo uso de las siguientes ecuaciones para generar el barrido de la frecuencia:

$$(3.1) \quad \delta_{sweep} = A \sin[\theta(t)]$$

$$(3.2) \quad \theta(t) = \int_0^{T_{Rec}} \omega(t) dt$$

$$(3.3) \quad \omega = \omega_{min} + K(\omega_{max} - \omega_{min})$$

$$(3.4) \quad K = C_2 [\exp\left(\frac{C_1 t}{T_{Rec}}\right) - 1]$$

$$(3.5) \quad C_1 = 4$$

$$(3.6) \quad C_2 = 0.0187$$

Donde ω_{min} , ω_{max} corresponden con la frecuencia mínima y máxima para un UAV tipo *Multirotor*, T_{Rec} es el tiempo y A es la amplitud. Estas constantes fueron predefinidas en el código con los valores de la Tabla 3.1.

Parámetros		<i>Pitch</i>	<i>Roll</i>	<i>Yaw</i>
Amplitud de barrido	A	0.06	0.06	0.06
Frecuencia mínima (Hz)	ω_{min}	0.4	0.4	0.4
Frecuencia máxima (Hz)	ω_{max}	5	5	5
Longitud de registro de barrido (seg)	T_{Rec}	20	20	20

Tabla 3.1. Parámetros de entrada para el barrido de la frecuencia

Luego, se obtuvieron los registros del vuelo de donde solo fueron importante los datos sobre el control de actitud (*ATT*), que estima la orientación del dron, de donde se encuentran registros de la tasa de cambio de los ángulos para el cabeceo, balanceo y guiñada con respecto al tiempo, ver Figura 3.2, 3.3, y 3.4. A partir de ahí, no se continuó con el procedimiento para ajustar el PID a través de la metodología de identificación del sistema, puesto que no se obtuvieron las herramientas necesarias para realizar dicha tarea.

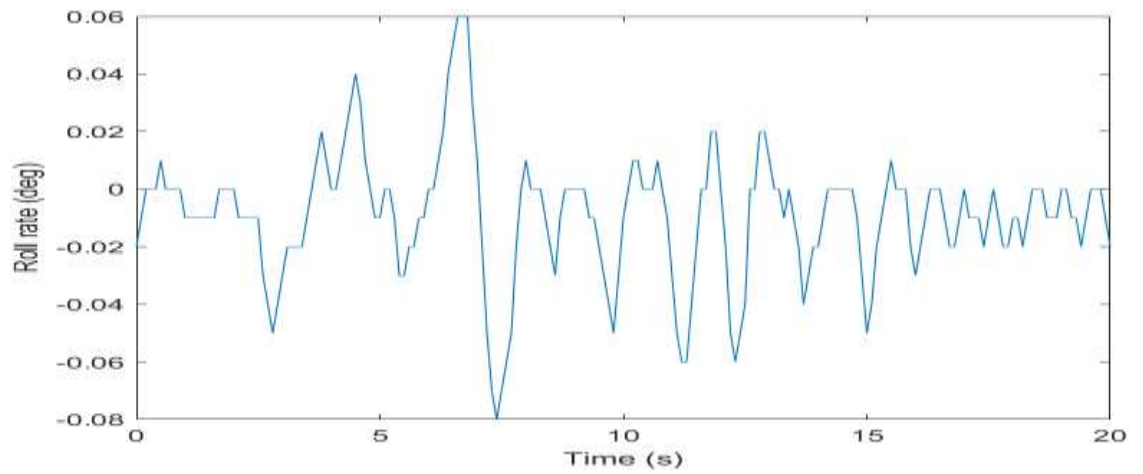


Figura 3.2. Entrada de la tasa de balanceo

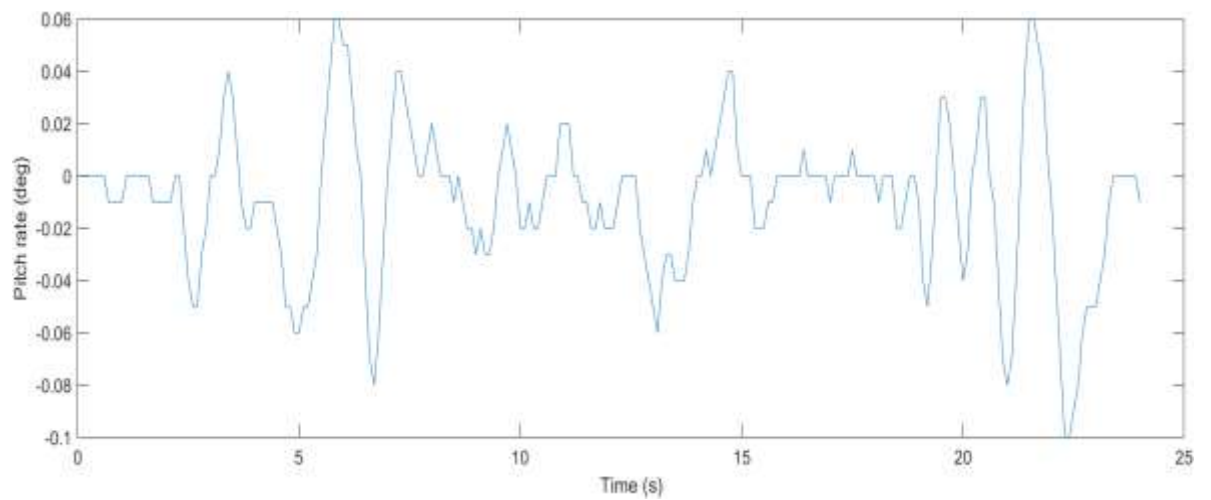


Figura 3.3. Entrada de la tasa de cabeceo

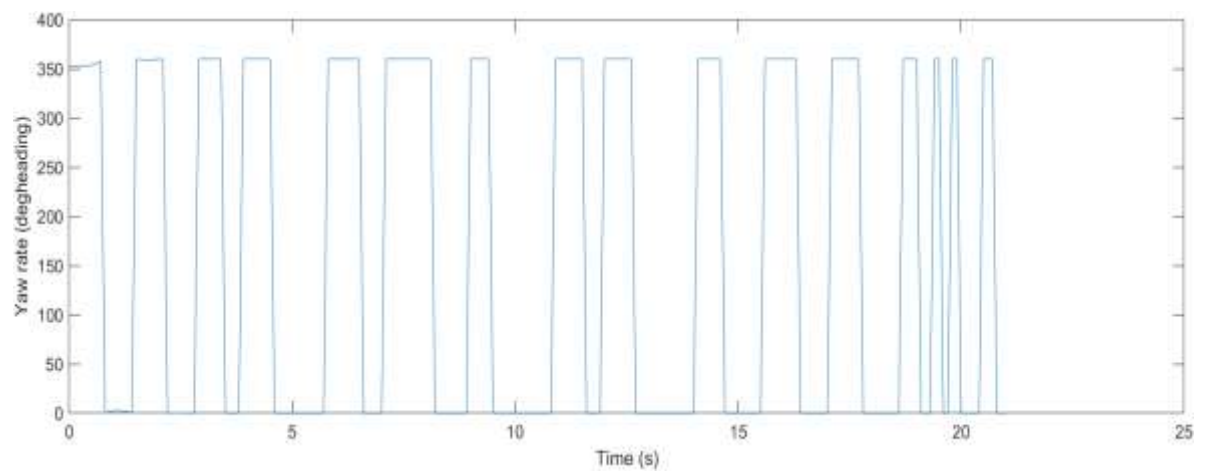


Figura 3.4. Diagrama de la tasa de guiñada

3.4 Marcación de Obstáculos

Se procedió a editar el código fuente del programa *Mission Planner* que se encuentra ubicado en el servicio de alojamiento *GitHub*, para lo cual se utilizaron los siguientes programas:

- Sistema Operativo: *Windows 2010*.
- Entorno de desarrollo integrado: *Visual Studio 2017*.
- Plataforma: *Microsoft .Net*
- Lenguaje de programación: *C#*.
- Componente de *.NET* para usar mapas: *Gmap*

Y para su compilación exitosa se siguieron los pasos sugeridos por sus mismos desarrolladores, como se puede ver en [32].

El procedimiento fue el siguiente:

- Añadir un ícono en la parte izquierda que active la selección del obstáculo en la clase *FlightPlanner.cs*:



Figura 3.5. Software Open Source Mission Planner con ícono para marcar obstáculos

- Graficar un polígono que permita encerrar el obstáculo y graficar la ruta del dron rodeando el obstáculo.



Figura 3.6. Aplicación de código libre *Mission Planner*: marcación del posible obstáculo



Figura 3.7. Software Open Source *Mission Planner* con recorrido esquivando el obstáculo

- Simulación, se la realizó con el simulador que tiene integrado en el planificador *Mission Planner* con el modelo *Multirotor* en x *Hexa*, puesto que es el modelo que tiene el dron agrícola real.



Figura 3.8. Software Open Source Mission Planner con simulación.

3.5 Ejecución de vuelos

Los primeros vuelos con el dron *small* fueron realizados en la cancha principal de fútbol de la ESPOL, y los vuelos finales tras el ajuste del control PID en terrenos abiertos del GEA-ESPOL; lugar en donde se realizaron las pruebas de vuelo con el dron agrícola multifuncional. Sin embargo, es preciso señalar que como punto de partida se realizó la simulación de rutas de vuelos planificadas sobre la misma cancha, para obtener resultados de tiempo de vuelo y área de cobertura total que podría abarcar un dron genérico en 1 hectárea. Estos datos son analizados en el siguiente capítulo.

En el caso del dron *small*, primero se realizaron vuelos controlados por RC en los modos de vuelo *Stabilize*, *PosHold* y *AltHold*, para determinar estabilidad de la nave antes de realizar vuelos autónomos a partir del GCS. Luego, se ejecutaron vuelos en modos *Guided* y *Auto* con el uso de la herramienta del planificador de vuelos de *Mission Planner* para crear polígonos que encierran un área requerida. Como ejemplo, ver Figura 3.9.

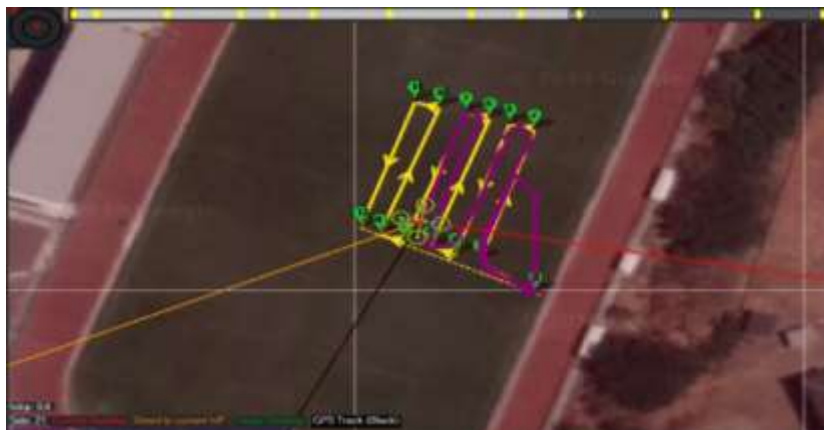


Figura 3.9. Visualizador del mapa de *Mission Planner* mientras el dron *small* realiza un vuelo automático con polígono definido por el usuario

Por su parte, en los vuelos autónomos ejecutados al inicio con altura relativa desde el planificador de *Mission Planner*, se observaron problemas para mantener la altura deseada, provocado por el desajuste del controlador PID genérico utilizado en estas pruebas de vuelo. Sin embargo, se buscaron soluciones como probar altura absoluta y con respecto al terreno. El análisis de esta implementación se detalla en el siguiente capítulo.

Finalmente, se ejecutaron vuelos con el dron agrícola multifuncional sin carga hasta que se obtuvieron problemas mecánicos con la estructura y demás inconvenientes que serán analizados en el siguiente capítulo.

3.5.1. Registros de vuelos almacenados

Se implementó una base de datos local con PostgreSQL con el modelo Entidad-Relación de la base de datos '*drone_flight*' que se muestra en la Figura 3.10. Sin embargo, no se automatizó el procedimiento de alojar los registros de vuelos en la base de datos, debido a problemas para establecer un punto de red *wifi*, el cual era necesario para establecer conexión con el servidor local que alberga la base de datos.

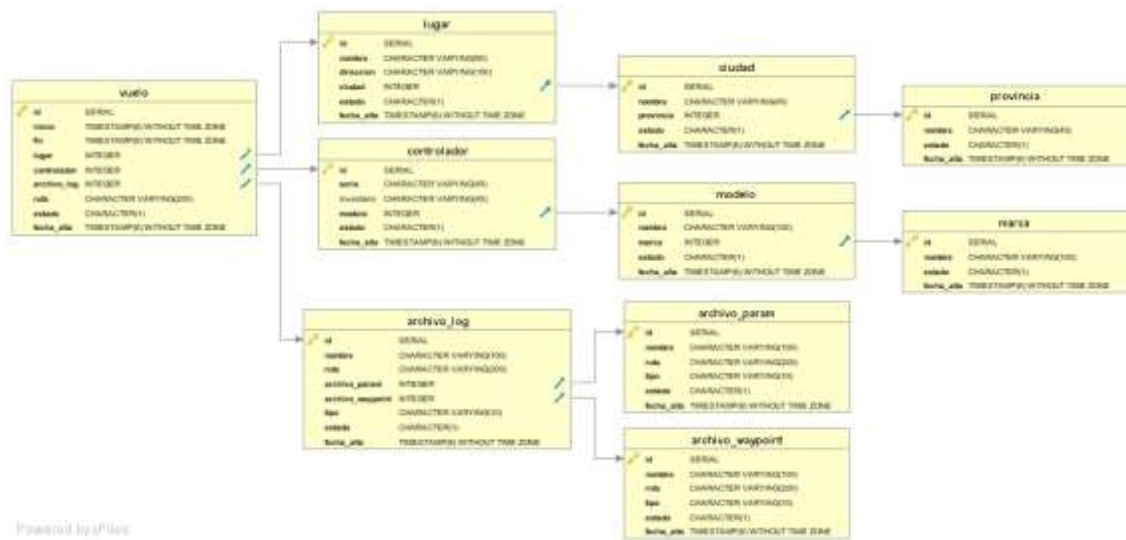


Figura 3.10. Diagrama Entidad-Relación de la base de datos drone_flight

CAPÍTULO IV

4. ANÁLISIS DE RESULTADOS

En esta sección se detallan los resultados encontrados tras varias pruebas de vuelo y funcionamiento del prototipo del dron agrícola multifuncional. Las cuales iniciaron con la simulación de vuelos a partir de los requisitos de la Tabla 2.2, para luego emprender las planificaciones de la Tabla 2.3, sobre el dron *small* y el agrícola respectivamente.

4.1 Evaluación de rutas de vuelo

Se realizaron simulaciones de vuelo a diferentes velocidades con recorrido total de 1.01 hectáreas, por medio del planificador *Mission Planner* y el *Software en bucle* (SITL), con el modelo de dron más aproximado al dron real que es: en forma x y 6 rotores, conocido como *hexa*, se simuló los vuelos tomando en cuenta los siguientes requerimientos del proyecto: 6 metros de distancia entre cada línea del recorrido, a 2 metros de altura y además se agregó un tiempo de espera de 1 segundo en cada punto del recorrido, cabe destacar que los segundos de espera no son aumentados por el planificador al tiempo estimado del recorrido del área total y a pesar de que el requerimiento de la velocidad es a 4 m/s, se realizaron pruebas a 3, 4, y 5 m/s, puesto que el simulador tiene un modelo predefinido del dron al cual no se le puede agregar las características físicas del dron como por ejemplo el peso, sus dimensiones, alcance de aspersores, etc., parámetros importantes que pueden hacer variar el tiempo del recorrido total. Además, se realizaron un total de 3 pruebas por cada velocidad y se tomó de preferencia el peor de los casos con respecto al mayor porcentaje gastado de batería.

4.1.1 Vuelos simulados

Las características globales de todos los vuelos se encuentran en la Tabla 4.1. Cada uno de los vuelos empieza con el porcentaje total de batería y con 24 voltios aproximadamente.

Datos Globales		
Área:	10102 m ²	1.01 ha
Altitud:	2 m	
Elevación del terreno:	100-107 m	
Distancia entre líneas:	6 m	
Número de líneas:	17 (Líneas largas del recorrido)	
Distancia total que recorre:	1.96 km	
Porcentaje de batería inicial:	100%	23.99 V
Tiempo de espera en cada punto:	1 seg	

Tabla 4.1. Parámetros globales de simulaciones de vuelo con 6 metros de separación entre las líneas del recorrido.

La prueba de vuelo a 3 m/s de velocidad, tuvo una duración de 13.35 minutos aproximadamente, el porcentaje de consumo de la batería al terminar el recorrido del área fue de 34% y al regresar al punto de partida desde el punto final del recorrido de las 1.01 hectáreas, el consumo de batería fue 37%. (ver Tabla 4.2)

3 metros/segundo	
Tiempo Aproximado:	13.35 min
Porcentaje de batería final sin RTL:	66% (Sin retornar al inicio)
Porcentaje de batería final con RTL:	63%

Tabla 4.2. Parámetros de las simulaciones de vuelo a 3 m/s de velocidad

La prueba de vuelo a 4 m/s de velocidad, duró 10.11 minutos aproximadamente, el porcentaje de consumo de la batería al terminar el recorrido del área descendió a 6% y al regresar al punto de partida desde el punto final del recorrido de las 1.01 hectáreas, ascendió también 6%. (ver Tabla 4.3)

4 metros/segundo	
Tiempo Aproximado:	10.11 min
Porcentaje de batería final sin RTL:	72% (Sin retornar al inicio)
Porcentaje de batería final con RTL:	69%

Tabla 4.3. Parámetros de las simulaciones de vuelo a 4 m/s de velocidad

La prueba de vuelo a 5 m/s de velocidad, duró 8.09 minutos aproximadamente, el porcentaje de consumo de la batería al contrario de la velocidad anterior

descendió, tanto al terminar el recorrido del área con una diferencia de 3%, como al regresar al punto de partida desde el punto final del recorrido de las 1.01 hectáreas, con una diferencia de 5%. (ver Tabla 4.4)

5 metros/segundo	
Tiempo Aproximado:	8.09 min
Porcentaje de batería final sin RTL:	75% (Sin retornar al inicio)
Porcentaje de batería final con RTL:	73%

Tabla 4.4. Parámetros de las simulaciones de vuelo a 5 m/s de velocidad

A medida que la velocidad aumenta el consumo de batería disminuye, es mejor tomar en cuenta el tiempo de la batería con el retorno hacia donde inició su vuelo (RTL), porque es una opción que se puede agregar fácilmente al recorrido y es necesario para evadir ir a ver el dron al último punto del recorrido y de esta manera evitar estar encima del sembrío a causa de recoger el dron.

El tiempo del planificador es distinto al tiempo real que se toma el dron en ejecutar la simulación de recorrer las 1.01 hectáreas, en este caso aumentó hasta 5 minutos, tomando en cuenta el tiempo del sistema de la laptop donde se realizaron las simulaciones, ver Tabla 4.5., pero este tiempo también depende de los segundos de espera que se planifiquen en cada punto del recorrido, para este caso fueron 32 puntos, por lo tanto 32 segundos totales de espera.

Propiedades de Portátil	
Marca:	Hp
Procesador:	Intel® Core™ i5-7200U CPU @ 2.50GHz 2.71 GHz
Tipo de sistema:	64 bits
Memoria RAM:	8 GB
Sistema operativo:	Windows 10 Pro

Tabla 4.5. Propiedades de portátil usada para las pruebas de vuelo simuladas y reales.

Por los tiempos estimados del planificador se tomó en cuenta los 13.35 minutos para el máximo y los 10.11 para el mínimo de tiempo, se apartó los 8.09 minutos de la prueba de 5 m/s por ser un valor poco probable a causa de las baterías, motores y peso del dron, por lo cual se determinó que el tiempo del dron en recorrer 1.01 hectárea con una velocidad de 4 m/s, toma un máximo de 18.35 minutos, al aumentar los 5 minutos de ejecución real de la simulación y un

mínimo de 10.43 minutos, al aumentar los 32 segundos de espera. Las baterías reales tienen una duración aproximada de 20 minutos por lo cual se obtuvieron los resultados de la Tabla 4.6.

Resultados de vuelos simulados estimados a 20 min	
Área mínima:	1.10 ha
Área máxima:	1.94 ha
Área promedio:	1.2 ha

Tabla 4.6. Resultados de vuelos simulados estimados al tiempo de duración de las baterías que es 20 min.

4.1.2 Vuelos del dron *small*

Para garantizar el correcto funcionamiento del controlador del FC *Pixhawk* fue preciso analizar los registros luego de cada vuelo, atendiendo sólo a lecturas de sensores, estabilidad en vuelo, altura correcta y potencia adecuada de los motores.

En la Tabla 4.7, se presentan los resultados de las 4 pruebas de vuelo manuales realizadas con el dron *small*, de donde se calcularon los errores relativos del desfase en los 3 ejes del controlador de actitud y de la altitud mientras el dron se mantenía estacionario a una altura determinada; y el balance promedio de motores registrado a manera de potencias de salida durante todo el vuelo. Estos fueron realizados en modo de vuelo *Stabilize*, *AltHold* y *PosHold*. Cabe recalcar que, tanto los indicadores escogidos como los valores máximos aceptables fueron puestos de forma empírica de acuerdo con la observación metódica realizada durante y posterior a cada vuelo, con el fin de entender el problema de pérdida de altura e inestabilidad del dron.

A partir de la Figura 4.1, se puede analizar el desfase constante en los ángulos de *pitch* y *roll*. Sin embargo, este hecho sólo afectaba la estabilidad de la nave en modo de vuelo *Stabilize* ya que, según la Figura 4.2, fue durante este modo de vuelo que perdía o ganaba demasiada altura obedeciendo al desfase de los ángulos. Mientras que, en modo *AltHold* mantuvo una altura estable, de igual manera se observó que el desfase de los ángulos produjo movimientos inesperados a la nave. Finalmente, fue el modo de vuelo *PosHold* el cual permitió que el dron mantenga la altura deseada y no reflejaba los errores en

el desfase de los ángulos del controlador de actitud. Por su parte, la medida del balance de los motores nunca fue directamente proporcional a la pérdida o ganancia de altura, ni al desfase de los ángulos.

Pruebas de vuelos con radio control manteniendo la altura										
Indicadores de vuelo			1° vuelo		2° vuelo		3° vuelo		4° vuelo	
Tipo	Descripción	Valor máximo aceptable	Stabilize	PosHold	AltHold	PosHold	AltHold	PosHold	Stabilize	PosHold
Actitud en estado estacionario	Desfase promedio entre ángulo de balanceo deseado y real en grados	0.5	2.08276	0.8559	0.7233	1.5547	0.7729	1.3381	2.7070	1.8500
	Desfase promedio entre ángulo de cabeceo deseado y real en grados	0.5	1.95517	1.3397	1.9619	0.3737	0.6648	0.9750	2.0050	0.2180
	Desfase promedio entre ángulo de guiñada deseado y real en grados de rumbo	0.5	0.05194	0.0077	0.0104	0.6651	0.0110	0.1884	0.0074	0.0089
Altitud durante todo el vuelo	Diferencia promedio entre altura deseada y altura calculada por EKF	0.2	1.68373	0.0538	0.0171	0.0124	0.0040	0.0163	1.2305	0.0086
Balance de motores durante todo el vuelo	Diferencia promedio entre salida máxima y mínima del motor	150	266		117		198		139	

Tabla 4.7. Resultados de registros de datos de vuelos controlados de forma manual acerca de la actitud, altitud y potencia de motores.

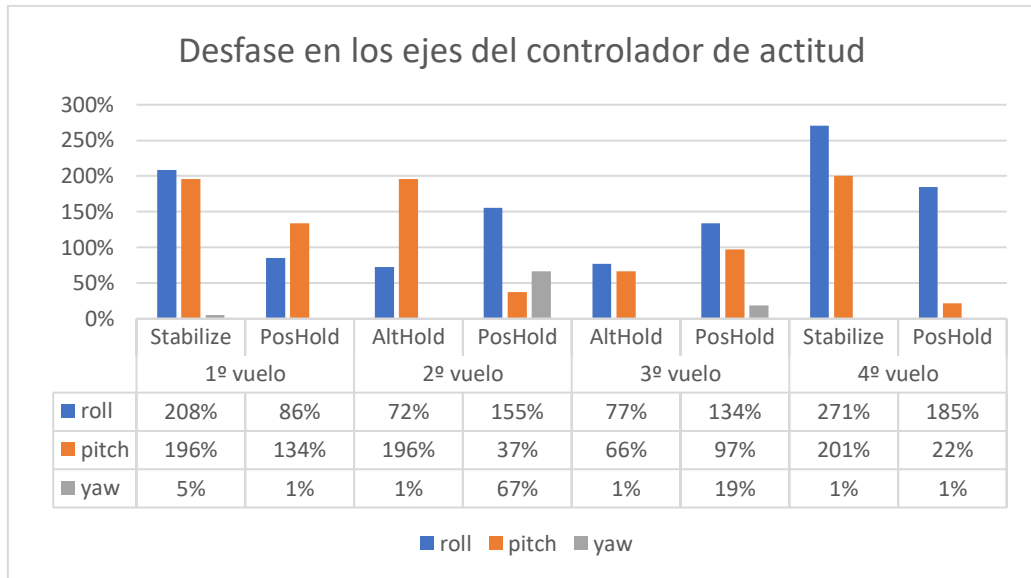


Figura 4.1. Porcentajes del error relativo del desfase de los ángulos de *roll*, *pitch* y *yaw*.

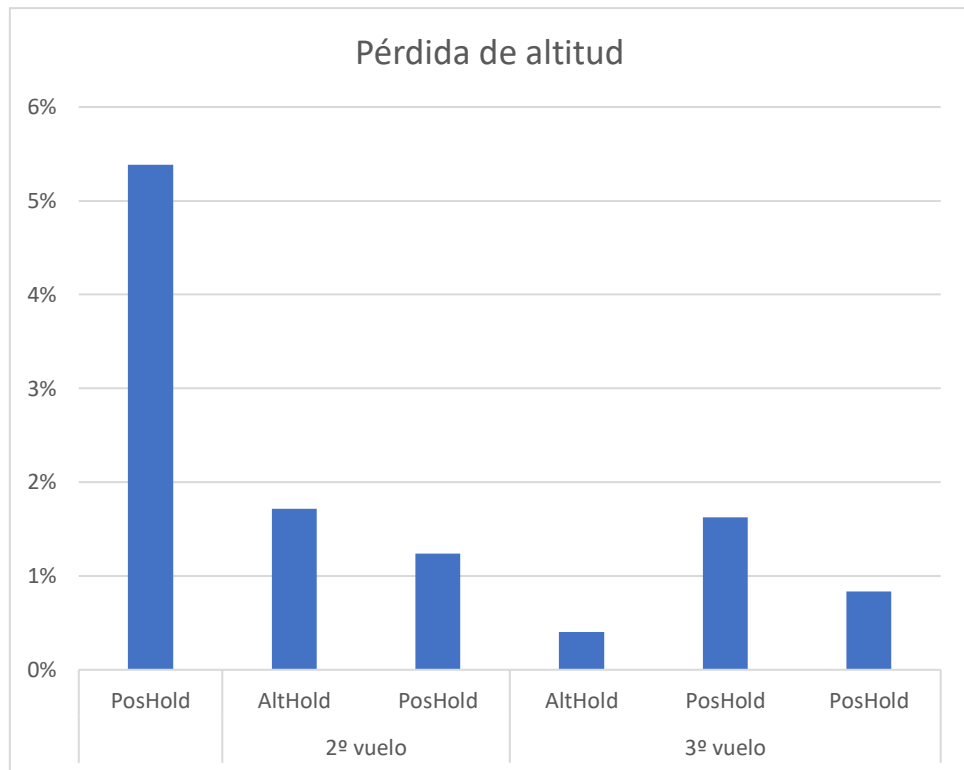


Figura 4.1. Porcentajes del error relativo entre altura real y altura desea

Debido a que, el problema de mantener la altura se agrava a medida que el dron debe avanzar hacia un punto de referencia, del inglés *waypoint*, se utilizaron diferentes tipos de alturas seleccionadas desde el planificador de

vuelos. De esta manera se logró determinar que, al utilizar altura con respecto al terreno se reduce el error relativo entre la altura deseada y la ejecutada por el dron, como lo refleja la Figura 4.3.

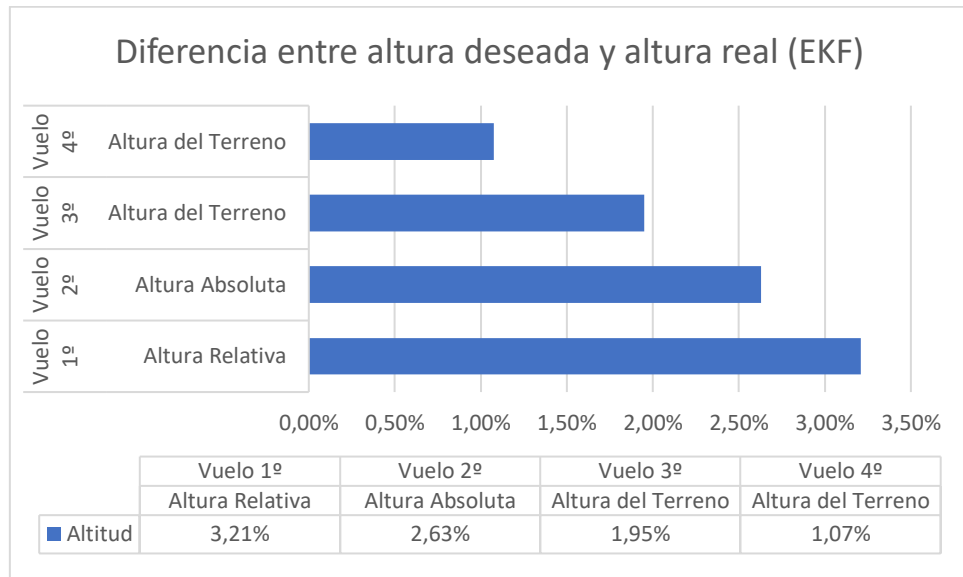


Figura 4.2. Porcentajes del error relativo de la diferencia entre la altura deseada y la real.

Lo expuesto anteriormente se muestra en las Figuras 4.4, 4.5 y 4.6 en las que se grafican la altura deseada y la altura real de cada vuelo, notando como hay mayor diferencia en los dos primeros gráficos donde se hace uso de la referencia relativa y absoluta para mantener la altura deseada. Sin embargo, la solución encontrada con el uso de referencia de altura con respecto al terreno hace que la altura varía respecto a la superficie del terreno según el mapa utilizado por defecto por *Mission Planner*. Esto dificulta la observación del vuelo que ante cualquier irregularidad del terreno perderá o ganará altura. Sin embargo, resulta ideal para las aplicaciones del presente proyecto, puesto que se requiere que la multifunción de fumigar y fertilizar se realice a la misma altura del suelo, acogiendo alguna irregularidad imprevista pero determinada eficazmente por el planificador de vuelos. De todas formas, es necesaria la implementación de sensores de flujo óptico o láser para garantizar un error a lo mucho en milímetros de la altura sobre la cual debe sobrevolar el cultivo.

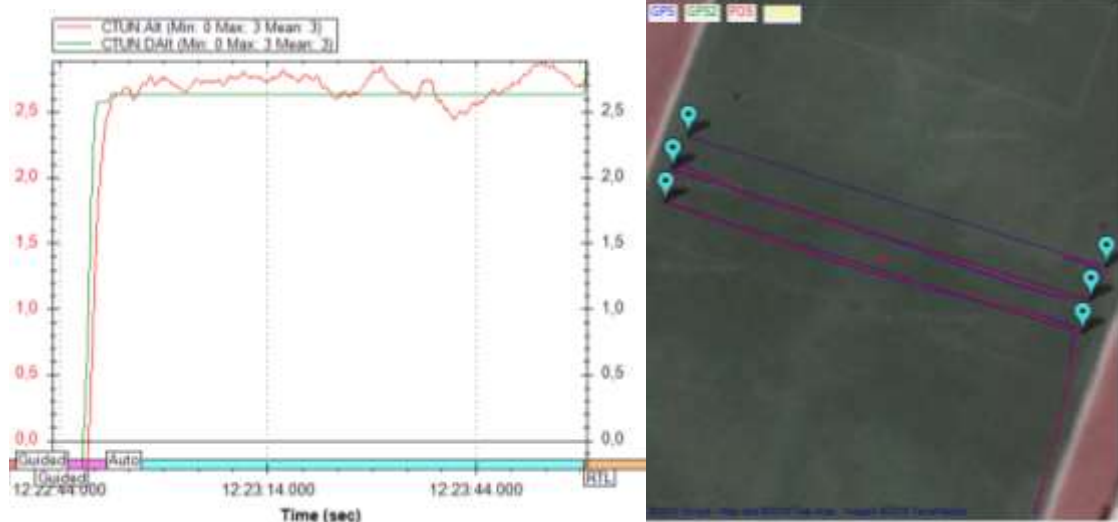


Figura 4.3. Gráfica de altura deseada y calculada durante el vuelo con referencia relativa. Mapa de ruta ejecutada

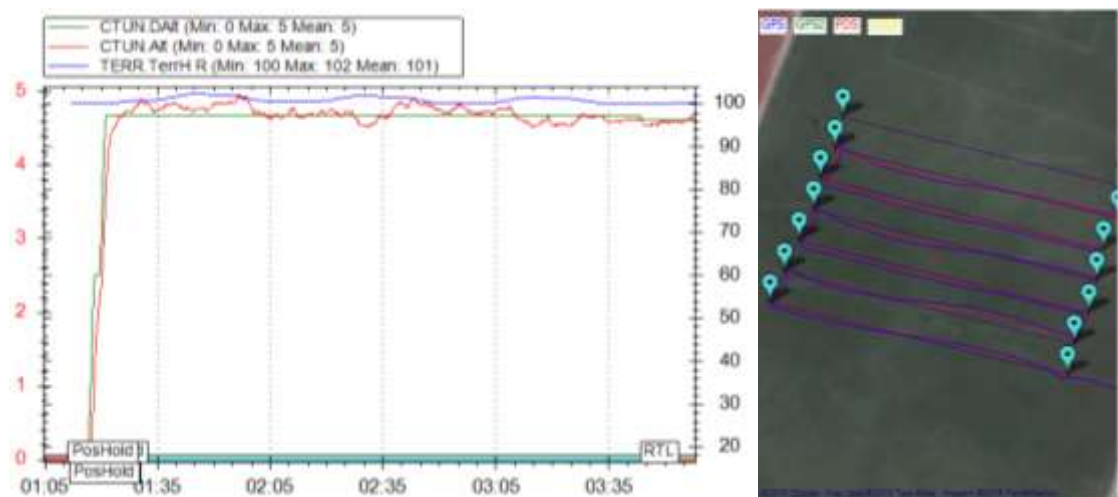


Figura 4.4. Gráfica de altura deseada y calculada durante el vuelo con referencia absoluta. Mapa de ruta ejecutada

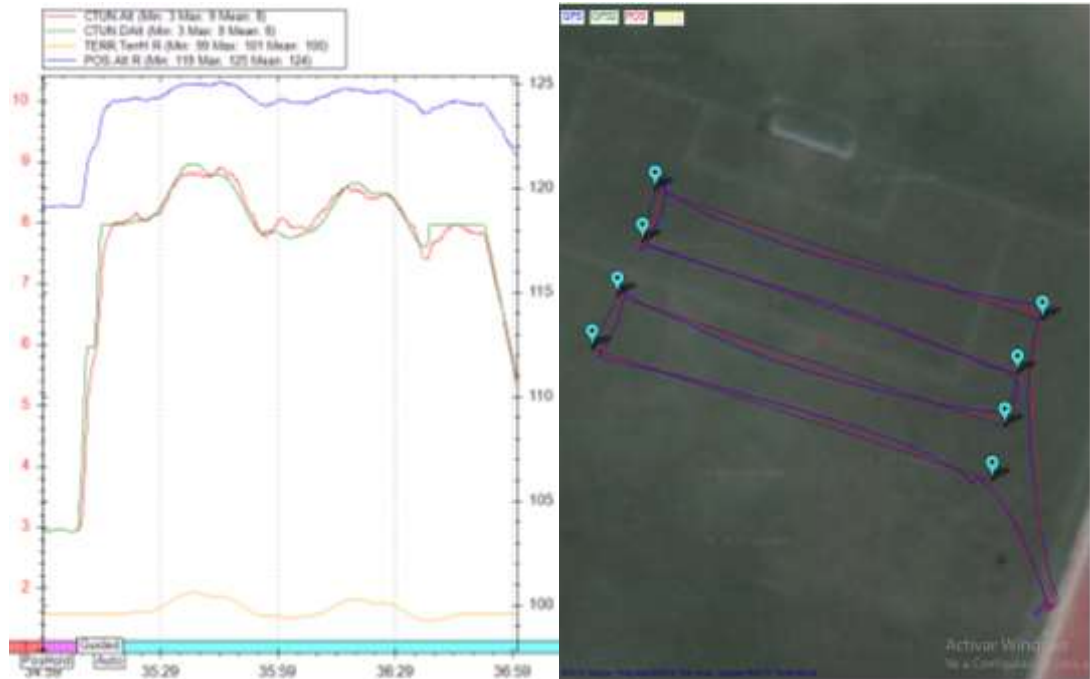


Figura 4.5. Gráfica de altura deseada y calculada durante el vuelo con referencia respecto al terreno. Mapa de ruta ejecutada

4.1.3 Vuelos del dron agrícola multifuncional

Los primeros vuelos de despegue y aterrizaje se efectuaron sin contratiempos, por lo que se prosiguió a planificar un vuelo con de avance de 10 metros, luego uno a 20 metros y finalmente otro a 30 metros de distancia desde el punto de partida. En todos los casos se incluyó un RTL para que el dron regrese al punto de partida, tal como debe cumplirlo en los cultivos de arroz. Sin embargo, al finalizar el tercer vuelo, a 30 metros de distancia, el dron experimentó grandes oscilaciones de cabeceo y balanceo que se aprecian en las Figuras 4.7 y 7.8 respectivamente. En ambos casos los valores del ángulo real sobrepasan los valores deseados hasta con 10 grados de diferencia en el caso del *pitch*, y llegan hasta los 40 grados en el caso de *roll*. Por lo que se precisó que, el dron debe pasar por el proceso de sintonización que recomienda *ArduPilot* como parte de los chequeos previos al primer vuelo.

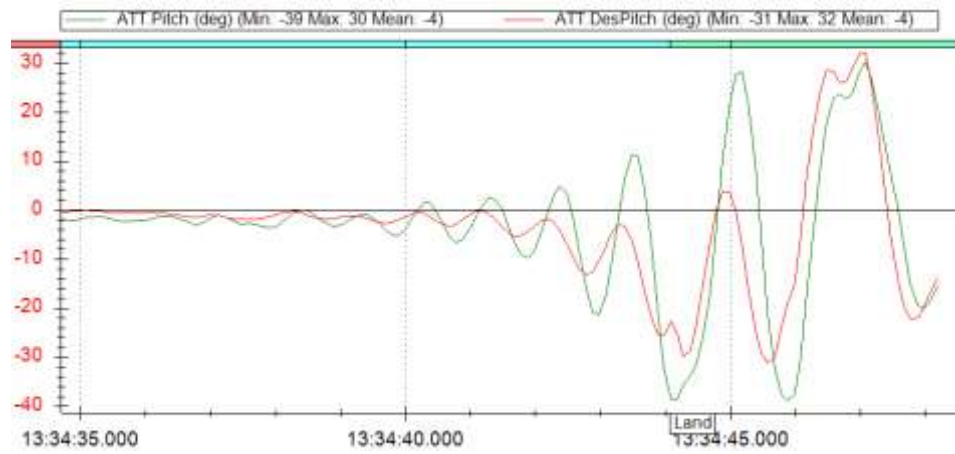


Figura 4.6. Gráfica del desfase entre ángulo *pitch* deseado y real.

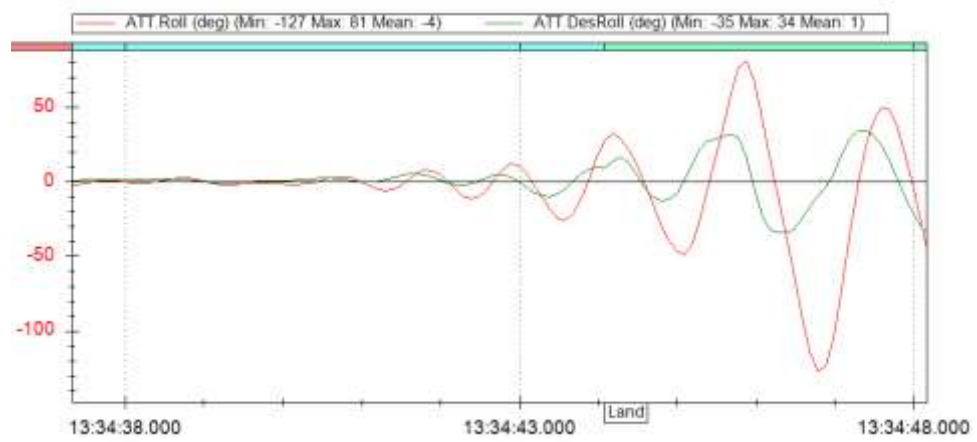


Figura 4.7. Gráfica del desfase entre el ángulo *roll* deseado y real.

CAPITULO V

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO A FUTURO

5.1 Conclusiones

El dron agrícola multifuncional con una sola carga de batería, es decir en 20 minutos, puede abarcar 1.10 hectáreas o un 55% de lo esperado en el peor de los casos, 1.94 hectáreas o un 97% de lo esperado en el mejor de los casos y en promedio, 1.52 hectáreas o un 76% de lo esperado, por lo tanto, en promedio el dron abarca 4.56 hectáreas por hora.

La tecnología *open source* puede ser utilizada para desarrollar aplicaciones agrícolas de fumigación y fertilización de cultivos sobre un dron industrial.

Rutas de vuelos planificadas con referencia de altura con respecto al terreno proporcionan mejor seguimiento de la trayectoria trazada, con menos del 2% de pérdida de altura.

Es necesaria la sintonización del controlador PID antes de ejecutar rutas de vuelos sobre un dron, sea de dimensiones pequeñas o grandes, para mantener estable la nave a lo largo de toda la trayectoria de vuelo.

5.2 Limitaciones

Cuando se establecía conexión entre el FC y el GCS por medio del protocolo *MAVlink*, el tiempo de la recepción de los parámetros sobrepasaba por mucho el tiempo de espera que una aplicación debe otorgar al usuario. Esto hizo que el trabajo de campo sea lento y costoso en medida de tiempo por las constantes veces en que se debía desconectar y volver a conectar a *Mission Planner*.

El uso de un GPS diferencial con error de precisión de hasta 5 metros generó incertidumbre en la exactitud de la posición del dron durante los vuelos automáticos.

5.3 Recomendaciones

Realizar todas las calibraciones de rutina antes de cualquier vuelo, sobre todo si las condiciones internas o externas del dron han cambiado. Por ejemplo, condiciones internas pueden ser integración de nuevos sensores, cambio en motores o algún periférico del controlador; condiciones externas son el lugar geográfico y hora del día donde el dron va a iniciar el vuelo, etc.

5.4 Trabajos a futuro

Completar el procedimiento de identificación del sistema para ajustar el PID sin necesidad de sintonizar dicho controlador, haciendo uso del trabajo inicial acerca del barrido de la frecuencia de *roll*, *pitch* y *yaw*, como datos de entrada para aproximar el modelo dinámico del dron agrícola.

Mapear las zonas a recorrer antes del vuelo y procesar los datos con *software* como *AgiSoft PhotoScan* el cual puede generar curvas de nivel, el modelo digital de elevaciones (DEM) y mapas de Índice de Vegetación de Diferencia Normalizada, conocido como *NDVI*, para alcanzar a desarrollar agricultura de precisión con las multifunciones del dron.

Agregar al simulador de *Mission Planner*, *SITL*, las opciones de:

- Uso estimado de batería.
- Peso del dron.
- Uso de bomba con el dron.

BIBLIOGRAFÍA

- [1] “Poisoning Prevention and Management,” *World Health Organization*, 2019. [Online]. Available: <https://www.who.int/ipcs/poisons/en/>. [Accessed: 18-May-2019].
- [2] “Información Agrícola y Pecuaria,” *Programa de Reparación Ambiental y Social del Ministerio del Ambiente (Ecuador)*, 2019. [Online]. Available: <http://pras.ambiente.gob.ec/web/siesap/informacion-ap>. [Accessed: 18-May-2019].
- [3] F. Javier and F. Briones, “ESTUDIO DE MERCADO PARA LA IMPORTACIÓN DE DRONES DE FUMIGACIÓN CON EL FIN DE OFRECER EL SERVICIO EN EL SECTOR AGRÍCOLA ESPECÍFICAMENTE EN CULTIVOS DE ARROZ, DE LOS CANTONES SANTA LUCÍA, DAULE Y NOBOL DE LA PROVINCIA DEL GUAYAS.,” UNIVERSIDAD DEL PACÍFICO, Guayaquil, 2017.
- [4] T. Rodríguez and M. Vásquez, “PLAN DE NEGOCIO PARA LA CREACIÓN DE UNA EMPRESA DE FUMIGACIÓN AÉREA POR HELICÓPTERO,” Universidad Laica Vicente Rocafuerte de Guayaquil, 2018.
- [5] G. Amarillo, M. Amarillo, F. Sarmiento, and A. Gonzalez, “Drones Aplicados a la Agricultura de Precisión Drones Applied to Precision Agriculture,” p. 11, 2015.
- [6] El Telegrafo, “El Gobierno emprende acciones para reducir los costos de producción en el agro,” *El Telegrafo*, Guayaquil, Ecuador, p. 1, 17-May-2019.
- [7] Novedades Agrícolas S.A, “Riego por Aspersión,” 2016. [Online]. Available: <https://www.novagric.com/es/riego/sistemas-de-riego/riego-por-aspersion>. [Accessed: 28-Jul-2019].
- [8] F. P. Bowden and D. T. Frank Philip Bowden, “Friction and Lubrication of Solids Part II,” 2001.
- [9] INEC and ESPAC, “Encuesta de Superficie y Producción Agropecuaria

Continua (ESPAC) 2018,” 2019.

- [10] SIPA, “Ficha del Cultivo de Arroz,” *Sistema de Información Pública Agropecuaria (SIPA)*, 2018. [Online]. Available: <http://sipa.agricultura.gob.ec/index.php/arroz>. [Accessed: 22-May-2019].
- [11] C. Hofmann, S. Lauber, B. Haefner, and G. Lanza, “Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework,” *Procedia Manuf.*, vol. 23, no. 2017, pp. 45–50, 2018.
- [12] M. O. Ahmad, J. Markkula, and M. Oivo, “Kanban in software development: A systematic literature review,” *Proc. - 39th Euromicro Conf. Ser. Softw. Eng. Adv. Appl. SEAA 2013*, no. December 2014, pp. 9–16, 2013.
- [13] R. B. Wakode, L. P. Raut, and P. Talmale, “Overview on Kanban Methodology and its Implementation Overview on Kanban Methodology and its Implementation,” *Ijsrd*, vol. 3, no. October, pp. 2518–2521, 2015.
- [14] C. Bernal, *Metodología de la investigación*, Tercera., vol. العدد الحـا. Bogotá: Pearson Education, 2001.
- [15] J. P. Zumba, “Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software,” *INNOVA Res. J.*, vol. 3, no. 10, pp. 20–33, 2019.
- [16] M. Lorenz, “What is Pixhawk,” 2019. [Online]. Available: <http://pixhawk.org/>. [Accessed: 20-Jun-2019].
- [17] AliExpress, “Pixhawk PX4 Autopilot PIX 2.4.8 32 Bit Flight Controller,” *IQHOBBY Store*, 2019. [Online]. Available: <https://www.aliexpress.com/item/32616029350.html>. [Accessed: 20-Jun-2019].
- [18] Radiolink Electronic Limited, “Radiolink R12DS 12-CH 2.4GHz DSSS & FHSS Receiver,” 2019. [Online]. Available: <http://radiolink.com.cn/doce/product-detail-126.html>. [Accessed: 21-Jun-2019].
- [19] RadioLink Electronic Limited, “Radiolink AT9S 10channel 2.4Ghz 10CH remote control system,” 2019. [Online]. Available: <http://radiolink.com.cn/doce/product-detail-119.html>. [Accessed: 22-Jun-2019].

- [20] HobbyKing, "FPV Radio Telemetry Kit." [Online]. Available: https://hobbyking.com/es_es/fpv-radio-telemetry-kit-915mhz-1.html. [Accessed: 22-Jun-2019].
- [21] ArduPilot Dev Team, "ArduPilot," 2019. [Online]. Available: <https://github.com/ArduPilot>. [Accessed: 20-Jun-2019].
- [22] ArduPilot Dev Team, "SITL Simulator (Software in the Loop)," 2019. [Online]. Available: <http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>. [Accessed: 30-Jul-2019].
- [23] Microsoft, "Características de Visual Studio," 2019. [Online]. Available: <https://docs.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview?view=vs-2019>. [Accessed: 01-Aug-2019].
- [24] Microsoft, "What is .NET?," 2019. [Online]. Available: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>. [Accessed: 01-Aug-2019].
- [25] ArduPilot Dev Team, "Tuning Process Instructions," 2019. [Online]. Available: <http://ardupilot.org/copter/docs/tuning-process-instructions.html>. [Accessed: 20-Jul-2019].
- [26] J. Liu, W. Pan, R. Qu, and M. Xu, "Research on the Application of PID Control with Neural Network and Parameter Adjustment Method of PID Controller," pp. 72–76, 2019.
- [27] M. M. Ferdous, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Fuzzy clustering based nonlinear system identification and controller development of Pixhawk based quadcopter," *9th Int. Conf. Adv. Comput. Intell. ICACI 2017*, pp. 223–230, 2017.
- [28] W. Saengphet, S. Tantrairatn, C. Thumtae, and J. Srisertpol, "Implementation of system identification and flight control system for UAV," *2017 3rd Int. Conf. Control. Autom. Robot. ICCAR 2017*, no. January 2018, pp. 678–683, 2017.
- [29] S. Sakulthong, S. Tantrairatn, and W. Saengphet, "Frequency Response System Identification and Flight Controller Tuning for Quadcopter UAV," *ESIT*

2018 - 3rd Int. Conf. Eng. Sci. Innov. Technol. Proc., pp. 1–6, 2019.

- [30] ArduPilot Dev Team, “Setting up SITL on Windows,” 2019. [Online]. Available: <http://ardupilot.org/dev/docs/sitl-native-on-windows.html>. [Accessed: 15-Jul-2019].
- [31] J. Ramos Gálvez, “Aplicación para la toma de datos de contexto para un UAV basado en Raspberry y ArduPilot,” 2017.
- [32] ArduPilot Dev Team, “Building Mission Planner with Visual Studio,” 2019. [Online]. Available: <http://ardupilot.org/dev/docs/buildin-mission-planner.html>. [Accessed: 20-Jul-2019].

ANEXOS

ANEXO 1: CHECKLIST



Centro de i + D + i de
Sistemas Computacionales

CHECKLIST		Código
Proyecto		
Fecha		
Lugar		
Hora		
Operarios		
Autorización		

	Antes	Campo
Equipos y Herramientas Fijas		
Radio Control con pilas cargadas (voltaje medido)		
Baterías de dron cargadas (voltaje medido)		
Batería de laptop.		
Herramientas manuales (destornillador, playo)		
Desarmadores		
Medidor de voltaje		
Telemetría de estación de tierra		
Repuestos / Auxiliares		
Tornillos para motores		
Tornillos para patas de dron		
Amarra de nylon		
Cinta aislante		
Destornilladores		
Revisión de Actividades		
Conexiones de elementos electrónicos		
Giro de los motores de forma correcta		
Potencia de los motores		
GPS (Por lo menos 8 satélites)		
Acelerómetro		