

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



**Implementación de un Sistema de Control de Vuelo Automático para un Vehículo
Aéreo no Tripulado (UAV)**

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

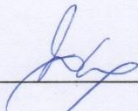
Leonardo Rodríguez León

Cartago, Noviembre de 2015

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA ELECTRÓNICA
PROYECTO DE GRADUACIÓN
ACTA DE APROBACIÓN

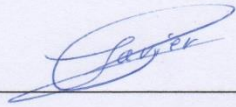
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Juan Carlos Jiménez Robles

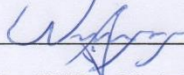
Profesor lector



Ing. Javier Pérez Rodríguez

Profesor lector

TEC | Tecnológico
de Costa Rica
Ingeniería Electrónica



Ing. William Marín Moreno
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 20 de Noviembre de 2015

Declaratoria de Autenticidad

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

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

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

Cartago, Lunes 23 de Noviembre 2015

Leonardo R.

Leonardo Rodríguez León

Céd: 1-1485-0631

Resumen

El problema de la erosión es muy común en los suelos de Costa Rica, los cuales pierden fertilidad afectando las actividades agrícolas, además este problema afecta el recurso hídrico debido a que lo contamina por medio de sedimentos. A través del tiempo el ser humano ha creado diferentes métodos para cuantificar el proceso de erosión con el fin de disminuir la presencia de este problema, uno de los más utilizados es el método de las parcelas de escorrentía que son porciones de terreno limitadas por paredes, que se mantienen bajo observación para cuantificar la erosión. Las parcelas de escorrentía suele ser un método costoso debido a su construcción y mantenimiento, y también resulta ser ineficaz ya que aporta información escasa o poco válida.

Es por ese problema que el Instituto Tecnológico de Costa Rica y el PRIAS del CeNAT se han dado a la tarea de cuantificar y modelar la erosión mediante el uso de Sistemas UAV (Vehículos Aéreos no Tripulados) dedicados a labores de fotogrametría. La idea es que estos sistemas logren adquirir variables físicas e imágenes georreferenciadas para reconstruir mapas de terrenos en gran resolución para cuantificar el proceso de la erosión en un tiempo dado y de forma automática.

En el presente proyecto se desarrolla la implementación de un sistema de control de vuelo de piloto automático para un Sistema UAV, el cual sea capaz de cumplir rutas de vuelo automatizadas programadas desde un ordenador mediante una plataforma de software especializada y obtener los datos del vuelo para el procesamiento posterior de georreferenciación de imágenes.

Palabras clave: fotogrametría, Mission Planner, Pixhawk, plataforma de piloto automático, sistema de control de vuelo automático, Vehículo Aéreo no Tripulado.

Abstract

The problem of erosion is very common in Costa Rica's soils. The soil loses fertility which affects agricultural activities, and this problem also affects water resources because it gets contaminated through sediments. Through time the human beings have created different methods to quantify the erosion process to reduce the presence of this problem, one of the most used methods is the runoff plots which are portions of land bounded by walls, which remain under observation to monitor the progress of erosion. The runoff plots are an expensive method because of its construction and maintenance, and it also turns out to be ineffective because they provide low valid information.

That problem has motivated a research initiative between the Costa Rica Institute of Technology and the PRIAS Laboratory of CeNAT with the purpose of quantifying and modeling erosion by using UAV (Unmanned Aerial Vehicles) systems dedicated to photogrammetry work. The idea is that these systems measure physical variables and georeferenced images to reconstruct maps of the ground in high resolution to quantify the erosion process automatically at a given time.

In this project, we develop the implementation of an autopilot flight control system for an UAV system, which is able to follow automated flight routes programmed from a computer using a specialized software platform and get the flight data for further processing of georeferencing of images.

Keywords: photogrammetry, Mission Planner, Pixhawk, autopilot platform, automatic flight control system, Unmanned Aerial Vehicle.

Dedicatoria

A mi madre Astrid León Alvarado y a mi padre Alexis Rodríguez Hernández por todos los años de formación hacia mi persona, el amor, apoyo, consejos y motivación que me han brindado en toda mi vida y cada logro que consiga en la vida siempre será gracias a ellos.

A mis hermanos Alexis, Katalina, Luis Carlos y mi sobrino Ignacio por la unión que tenemos como familia en todas las situaciones difíciles que hemos pasado, y por el apoyo constante con el cual puedo contar siempre.

Al resto de familiares con los que me he relacionado, especialmente los que me han brindado una mano durante mi formación profesional.

A mis amigos Elmer Soto y Martín Umaña con los que he pasado tanto momentos juntos desde el colegio, el apoyo incondicional y por ser los mejores amigos que he tenido en toda mi vida.

A todas las personas con las que me he relacionado, amistades, compañeros de trabajo con los que he crecido y he conocido durante este crecimiento como persona y profesional ya que gracias a todos ellos me encuentro realizando esta dedicatoria.

Agradecimientos

Un agradecimiento especial al Dr.-Ing Renato Rimolo Donadio, por brindarme esta oportunidad de formar parte de este proyecto tan especial, por su confianza hacia mi persona y disponibilidad de ayudar ante cualquier duda o problema para haber logrado con éxito el presente proyecto de graduación.

Agradezco al Ing. Sergio Arriola Valverde por su ayuda durante el desarrollo del proyecto, sus aportes y consejos para desarrollar el proyecto de la mejor manera y por facilitarme la documentación sobre su trabajo realizado.

A los profesores de la Escuela de Electrónica por sus enseñanzas en todos los cursos, cada uno con sus metodologías con las que he desarrollado varias destrezas y aumentar cada vez más mi interés en la carrera.

A mis amigos y compañeros del Instituto Tecnológico de Costa Rica, principalmente a Felipe Arce, Adrián Fallas y Rolando Esquivel por su amistad en toda la carrera, además la ayuda en los estudios y en los proyectos.

ÍNDICE GENERAL

| | |
|---|----------|
| Capítulo 1: Introducción | 1 |
| 1.1 Objetivos | 4 |
| 1.2 Estructura del documento | 4 |
| Capítulo 2: Marco teórico | 5 |
| 2.1 Fotogrametría..... | 5 |
| 2.2 Sistemas UAV | 7 |
| 2.2.1 Controlador de vuelo | 8 |
| 2.2.2 Sistema de posicionamiento global (GPS)..... | 9 |
| 2.2.3 Unidad de medición inercial (IMU)..... | 10 |
| 2.2.4 Giroscopio | 10 |
| 2.2.5 Acelerómetro | 11 |
| 2.2.6 Magnetómetro | 12 |
| 2.2.7 Barómetro | 12 |
| 2.2.8 Orientación del cuerpo en el espacio | 12 |
| 2.2.9 Sistema de alimentación | 13 |
| 2.2.10 Sistema de comunicación..... | 14 |
| 2.2.11 Plataformas de piloto automático | 14 |
| 2.2.12 Plataforma de software (Estación) | 18 |
| 2.3 Protocolos de comunicación | 18 |
| 2.3.1 UART (Universal Asynchronous Receiver-Transmitter | 18 |
| 2.3.2 SPI (Serial Peripheral Interface) | 19 |
| 2.3.3 I ² C (Inter-Integrated Circuit) | 20 |
| 2.3.4 USB (Universal Serial Bus) | 20 |
| 2.3.5 CAN (Controller Area Network) | 21 |
| 2.3.6 MAVLink..... | 22 |

| | |
|--|-----------|
| Capítulo 3: Selección e implementación del sistema de control de vuelo automático | 23 |
| 3.1 Caracterización del sistema UAV | 23 |
| 3.1.1 X8+..... | 23 |
| 3.1.2 Control FlySky FS-TH9X..... | 25 |
| 3.1.3 Batería LiPO 4S..... | 26 |
| 3.1.4 Radios de telemetría 3DR..... | 27 |
| 3.1.5 GPS con magnetómetro | 28 |
| 3.2 Planteamiento y selección de la solución final | 30 |
| 3.2.1 Solución 1: Plataformas comerciales de fotogrametría con UAV | 30 |
| 3.2.2 Solución 2: Plataformas comerciales basadas en “open source” y “open hardware” | 31 |
| 3.2.3 Solución 3: Diseño de plataforma de piloto automático | 31 |
| 3.2.4 Análisis y selección de la solución | 32 |
| 3.3 Descripción del proceso de diseño | 35 |
| 3.3.1 Plataforma de piloto automático..... | 36 |
| 3.3.2 Plataforma de software de piloto automático (Estación)..... | 39 |
| 3.3.3 Implementación de la plataforma de piloto automático | 41 |
| 3.3.4 Descripción de la programación de las misiones de vuelo autónomo..... | 43 |
| Capítulo 4: Resultados y análisis..... | 45 |
| 4.1 Resultados del sistema de control antes del primer vuelo..... | 45 |
| 4.2 Resultados del sistema de control en vuelo manual | 48 |
| 4.3 Resultados del sistema de control en vuelo automático..... | 50 |
| 4.3.1 Resultados de ruta simple..... | 50 |
| 4.3.2 Resultados de ruta con comando para fotogrametría | 53 |
| 4.4 Resultados del sistema de control después del vuelo | 55 |

| | |
|--|-----------|
| Capítulo 5: Conclusiones y recomendaciones | 57 |
| 5.1 Conclusiones | 57 |
| 5.2 Recomendaciones | 59 |
| Bibliografía..... | 60 |
| Apéndices..... | 64 |
| A.1 Selección de la plataforma de piloto automático y estación | 64 |
| A.2 Manual de calibración | 68 |
| A.3 Código de los scripts de Python | 79 |

ÍNDICE DE FIGURAS

| | | |
|--------------------|---|----|
| Figura 1.1 | Parcela experimental de escorrentía [2] | 2 |
| Figura 2.1 | Método general de la fotogrametría [6] | 6 |
| Figura 2.2 | Principio de trilateración para un GPS [18] | 9 |
| Figura 2.3 | Diagrama de la fuerza de Coriolis..... | 11 |
| Figura 2.4 | Principio físico del acelerómetro [20]..... | 11 |
| Figura 2.5 | Representación de los ángulos de Euler en un sistema UAV [22]..... | 13 |
| Figura 2.6 | Plataforma de piloto automático A2 (esquina superior izquierda), IMU (esquina superior derecha) y GPS [23] | 15 |
| Figura 2.7 | Plataforma piloto automático CC3D [25] | 16 |
| Figura 2.8 | Plataforma piloto automático AeroQuad32 [26]..... | 17 |
| Figura 2.9 | Formato de datos transmitidos en el protocolo UART [30]..... | 19 |
| Figura 2.10 | Diagrama de interconexión del protocolo SPI [4] | 19 |
| Figura 2.11 | Diagrama de interconexión para el protocolo I2C | 20 |
| Figura 2.12 | Diagrama de interconexión para el protocolo USB [30]..... | 21 |
| Figura 2.13 | Formato de datos transmitidos en el protocolo MAVLink [33]..... | 22 |
| Figura 3.1 | Sistema UAV X8+ de 3DR [34] | 24 |
| Figura 3.2 | Control FlySky FS-TH9X con pantalla del kit FPV [37] | 25 |
| Figura 3.3 | Batería LiPO 4S del sistema UAV | 26 |
| Figura 3.4 | Radios de telemetría 3DR [40]..... | 27 |
| Figura 3.5 | GPS con magnetómetro 3DR [42] | 28 |
| Figura 3.6 | Diagrama modular del control de vuelo automático para el sistema UAV..... | 35 |
| Figura 3.7 | Plataforma de piloto automático Pixhawk | 38 |
| Figura 3.8 | Interfaz de la plataforma Mission Planner | 40 |
| Figura 3.9 | Plataforma Pixhawk conectada y montada en el X8+..... | 42 |
| Figura 3.10 | Sistema UAV X8+ completo | 42 |
| Figura 3.11 | Interfaz de la programación de rutas con Mission Planner | 43 |
| Figura 4.1 | Conexión UAV-Estación | 46 |
| Figura 4.2 | Prueba de vuelo manual | 48 |
| Figura 4.3 | Lógica de la ruta simple de prueba a programar | 50 |

| | | |
|--------------------|--|----|
| Figura 4.4 | Ruta simple programada por script de Python | 51 |
| Figura 4.5 | Prueba de vuelo de piloto automático | 52 |
| Figura 4.6 | Lógica de la ruta de prueba a programar..... | 53 |
| Figura 4.7 | Ruta de comando de fotogrametría en Mission Planner | 54 |
| Figura 4.8 | Ruta de prueba completada en Mission Planner | 54 |
| Figura 4.9 | Gráfico de altitud relativa elaborado por Mission Planner | 55 |
| Figura A.1 | Conexión en el Pixhawk vía USB | 68 |
| Figura A.2 | Selección del puerto vía USB y la velocidad en Mission Planner | 68 |
| Figura A.3 | Procedimiento de actualización del firmware | 69 |
| Figura A.4 | Conexión en el Pixhawk por Radios de Telemetría..... | 70 |
| Figura A.5 | Selección de puerto y velocidad en Mission Planner..... | 71 |
| Figura A.6 | Pixhawk conectado a Mission Planner..... | 71 |
| Figura A.7 | Selección del frame en Mission Planner | 72 |
| Figura A.8 | Selección de calibración del acelerómetro | 73 |
| Figura A.9 | Posiciones calibración acelerómetro | 73 |
| Figura A.10 | Selección de calibración del magnetómetro..... | 74 |
| Figura A.11 | Ventana de calibración del magnetómetro | 75 |
| Figura A.12 | Pasos calibración magnetometro, paso 1 (izquierda) y paso 2 (derecha) | 75 |
| Figura A.13 | Representación gráfica del control..... | 76 |
| Figura A.14 | Proceso de calibración del control | 77 |

ÍNDICE DE TABLAS

| | | |
|------------------|---|----|
| Tabla 2.1 | Clasificación de los sistemas UAV | 7 |
| Tabla 3.1 | Requerimientos de la plataforma de piloto automático | 29 |
| Tabla 3.2 | Calificación de las soluciones planteadas..... | 34 |
| Tabla 4.1 | Pesos de los módulos del sistema de control de vuelo automático | 56 |
| Tabla A.1 | Tabla comparativa de las plataformas de piloto automático | 65 |
| Tabla A.2 | Tabla comparativa de las plataformas de software | 67 |

Capítulo 1: Introducción

Uno de los principales recursos que brinda la naturaleza a los seres humanos es el recurso del suelo ya que se pueden realizar muchísimas actividades de cualquier tipo en éste, en el caso de la agricultura se requiere que el suelo se encuentre en las mejores condiciones, ya que se necesita de un suelo fértil para el crecimiento y desarrollo de las plantas que sirven como alimento para el hombre y los animales.

Debido al aumento de la actividad agrícola en los suelos, también se ha dado un aumento del uso indiscriminado de productos químicos como fertilizantes, herbicidas y pesticidas que generan una degradación directa en los suelos.

El problema de la erosión es muy común alrededor del mundo y nuestro país no es una excepción donde principalmente se genera por malas prácticas agrícolas y por los procesos de deforestación.

A pesar de la implementación de programas de conservación de suelos por décadas, las pérdidas de los suelos debido a la erosión continúan amenazando la sostenibilidad de la agricultura comercial y de subsistencia en muchas partes de Costa Rica [1].

Los efectos aditivos de la erosión en la productividad del suelo se reflejan en la disminución de fertilidad del suelo y por ende de la producción de los cultivos, pero también las comunidades ubicadas en las partes más bajas del terreno salen afectadas debido a la deposición de sedimentos en riachuelos, ríos, lagos y represas hidroeléctricas.

La erosión de los suelos es el resultado de 3 factores: viento, agua y labranza. La labranza ha logrado su efecto negativo en los suelos tanto como los factores climáticos del viento y agua, según estudios recientes demuestran que la dirección del movimiento del suelo por tareas de labranza tiene su impacto [1].

Es aquí donde la cuantificación de la erosión toma un papel importante ya que a partir de los valores medidos de pérdida de suelo se puede hacer predicciones de riesgos de erosión y evaluar de una manera cuantitativa la eficacia de las prácticas de control [2].

A través del tiempo se han creado diferentes metodologías de cuantificación de la erosión, éstas pueden ser clasificadas como medidas directas o indirectas. Las medidas directas se realizan en la misma zona donde se presenta la erosión y las medidas indirectas se derivan del análisis de una cantidad considerable de resultados obtenidos de las medidas directas en varios años [2].

La metodología directa más empleada en Costa Rica es el caso de las parcelas de escorrentía que consiste en el aislamiento de una porción de terreno con datos definidos como el tamaño, tipo de suelo, longitud y grado de pendiente. El método en sí permite un mejor control del proceso de erosión.

Estas porciones de terreno contienen un sistema colector que se encarga de recibir los sedimentos y luego almacenar la cantidad de agua con los sedimentos producidos por la escorrentía superficial de la parcela para su debido análisis, la figura 1.1 muestra un ejemplo de una parcela de escorrentía.

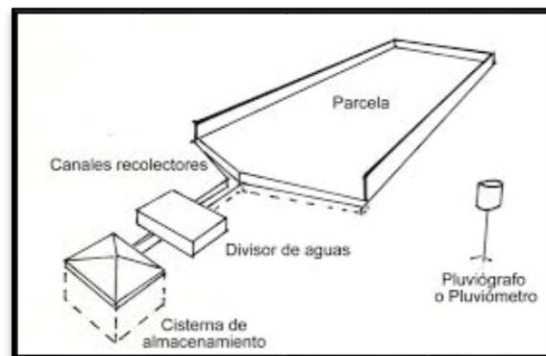


Figura 1.1 Parcela experimental de escorrentía [2].

Pese a ser el método más utilizado las parcelas para estudios de escorrentía resultan ser caras en construcción, mantenimiento y funcionamiento, además resultan ser ineficaces y emplean una gran cantidad de tiempo de personal a diferentes niveles por lo que ya no se consideran una opción viable para cuantificar la erosión [3].

Y es así que nace la iniciativa del PRIAS (Programa de Investigaciones Aerotransportadas) laboratorio adjunto al CeNAT (Centro Nacional de Alta Tecnología), y el ITCR (Instituto Tecnológico de Costa Rica), de buscar un método alternativo y moderno para cuantificar y modelar la erosión de forma más precisa.

Actualmente a nivel mundial se ha utilizado la técnica de capturas de fotografías satelitales o aéreas con aviones tripulados pero a gran escala, es decir la resolución espacial es baja y técnicamente no brindaría información detallada de los suelos para la medición que se necesita realizar.

Más recientemente se ha empezado a utilizar aeronaves con vuelos de menor altitud por lo que tienen que ser de menor tamaño y donde se llega a la solución de utilizar un sistema UAV (Unmanned Aerial Vehicle) para la captura de fotografías de alta resolución para cuantificar la erosión.

Dentro del ITCR ya se han realizado investigaciones relacionadas a la idea planteada, por lo que ya se ha trabajado con sistemas UAV y la creación de un sistema de adquisición de datos fotogramétricos para estos dispositivos, pero sin enfocarse directamente con el control de vuelo, el cual debe ser automatizado.

El dispositivo de adquisición de datos fue diseñado por el ingeniero Sergio Arriola como proyecto de Graduación para optar por el título de ingeniero en Electrónica Licenciatura, el sistema principalmente se encarga de las labores de telemetría y fotogrametría, ya que será capaz en dar registro de forma sistemática la información relevante que proviene de varios sensores, fotografías y enviar los datos directamente a una estación a tierra [4].

Y es por eso que se necesita implementar un dispositivo de control de vuelo de piloto automático que se complemente con el sistema de adquisición de datos para obtener la información de forma autónoma.

El desarrollo del presente proyecto consiste en el desarrollo de una solución para un sistema de control de vuelo automático para un sistema UAV con la meta final de poder realizar levantamientos fotogramétricos para cuantificar el proceso de erosión.

La solución planteada está claramente ligada a la utilización de un sistema embebido con gran capacidad de procesamiento ya que necesita la comunicación con varios sensores en tiempo real para lograr ubicar con gran precisión el sistema en un espacio físico. El dispositivo además tiene que ser de bajo peso y consumo de energía ya que se implementa directamente al sistema UAV.

Para lograr la solución previamente se tiene que realizar un análisis del sistema UAV a utilizar, y lograr una caracterización del sistema con las limitaciones generadas por los módulos conectados. Cabe resaltar que el ITCR y el PRIAS cuentan con el dron X8+ de 3D Robotics el cual se pretende explorar inicialmente e idealmente se le implementará el sistema de vuelo automático de ser posible.

1.1 Objetivos

El proyecto de graduación tiene como objetivo general:

- Desarrollar un sistema de piloto automático para el vehículo aéreo no tripulado multirrotor 3DR-X8+, que permita la programación de rutas de vuelo automatizadas a partir de un control central y sensores de navegación.

Objetivos específicos:

- Diseñar una solución con componentes de hardware y software para la implementación del sistema de piloto automático.
- Desarrollar las rutinas de programación necesarias para la comunicación y control del sistema de piloto automático así como la configuración del software que se requiera.
- Comprobar experimentalmente el funcionamiento del sistema de piloto automático incluyendo planeamiento de rutas, su descarga al sistema en la nave y ejecución de las rutas programadas.

1.2 Estructura del documento

El presente informe está compuesto de las siguientes partes:

- Capítulo 2: Marco teórico. Presenta los conceptos teóricos necesarios para el desarrollo del proyecto.
- Capítulo 3: Selección e implementación del sistema de control de vuelo automático. Aquí se detalla la comparación y utilización de criterios para llegar a la solución planteada y luego la implementación de ésta.
- Capítulo 4: Resultados y análisis. Se presentan los resultados obtenidos, así como el análisis y validación de estos.
- Capítulo 5: Conclusiones y recomendaciones.

Capítulo 2: Marco teórico

Este capítulo abarca principalmente temas ligados al desarrollo del proyecto, y pretende dar una idea al lector sobre conceptos relacionados con los sistemas UAV. Cada tema tratado se presenta de forma resumida con sus respectivas referencias bibliográficas, por lo que si desea profundizar en los temas se recomienda utilizar las referencias bibliográficas [4-33].

2.1 Fotogrametría

Es el fin principal por el cual se desea implementar un control de vuelo automático para el sistema UAV, en este proyecto no se abarca mucho este tema pero es importante definir el concepto.

La palabra fotogrametría proviene de la composición de tres términos en griego que son luz (photon), escritura (graphos) y medidas (metron) y se puede definir como el procedimiento de tomar fotografías con información georreferenciada y temporizada para luego obtener un plano del terreno [4].

El objetivo principal de la fotogrametría es la reconstrucción de un espacio tridimensional a partir de imágenes bidimensionales, es por eso que es necesaria la información proporcionada por cada imagen ya que se necesita para determinar la forma, dimensión y posición de un objeto. Esto se denomina método general de la fotogrametría [6]. En la figura 2.1 se ilustra el método mencionado anteriormente.

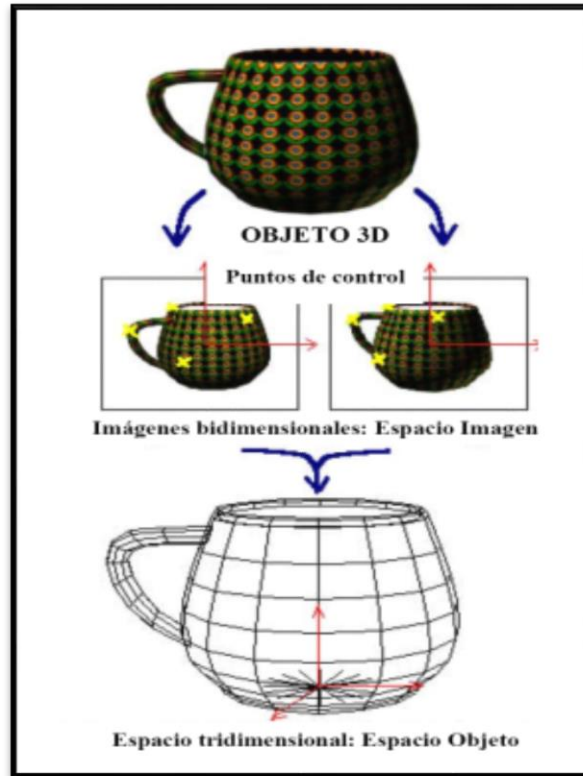


Figura 2.1 Método general de la fotogrametría [6].

La fotogrametría ha evolucionado con el paso del tiempo y ha dado origen a diferentes métodos de desarrollo como es el caso de la fotogrametría UAV que se describe como la fotogrametría aplicada por medio de una plataforma que opera de forma controlada o autónoma sin un piloto en el vehículo. Y el sistema UAV se encuentra equipado con cámaras fotográficas y de sensores necesarios para la orientación y posicionamiento en un sistema local o global de coordenadas [7].

En la actualidad existe una gran cantidad de herramientas de software para el proceso de reconstrucción de imágenes tridimensionales a base de fotografías referenciadas, se pueden mencionar plataformas como el Pix4Dmapper [8], Erdas Imagine [9] o el Agisoft [10].

2.2 Sistemas UAV

El término UAV proviene de las iniciales de su nombre en inglés Unmanned Aerial Vehicle que se traduce a Vehículo Aéreo No Tripulado y son aquellos vehículos o drones que son remotamente piloteados ya sea por control remoto o de forma autónoma [11].

En sus inicios los sistemas UAV se utilizaban principalmente para fines militares pero actualmente han aumentado en gran cantidad en el uso civil ya sea por investigadores, aficionados u otros en varios tipos de aplicaciones de las cuales resalta principalmente la observación y recolección de datos [11].

Debido a la gran cantidad de sistemas no hay una clasificación común aceptada debido a la amplia variedad de capacidades, tamaño y características de los sistemas. La mayoría de los UAV se describen en términos de peso máximo (carga incluida), rendimiento, altitud, radio de operación y su propósito por lo que en la tabla 2.1 se puede mostrar una clasificación de estos sistemas [12].

Tabla 2.1 Clasificación de los sistemas UAV.

| Categoría | Peso del UAV [kg] | Altitud Normal de Operación [pie] | Radio de Misión [km] | Resistencia | Uso Típico |
|----------------------|--------------------------|--|-----------------------------|--------------------|-------------------------------|
| MICRO | < 2 | Hasta 200 | 5 | Pocas horas | Reconocimiento, Inspección |
| MINI | 2-20 | Hasta 3000 | 25 | Hasta 2 días | Vigilancia |
| SMALL | 20-150 | Hasta 5000 | 50 | Hasta 2 días | Vigilancia |
| TACTICAL | 150-600 | Hasta 10000 | 200 | Hasta 2 días | Vigilancia |
| MALE | > 600 | Hasta 45000 | Ilimitado | Días/Semanas | Vigilancia, transporte |
| STRIKE/COMBAT | > 600 | Hasta 65000 | Ilimitado | Días/Semanas | Vigilancia, Recolección Datos |

Actualmente existen distintas marcas comerciales de sistemas UAV, todas con políticas y fines diferentes, entre las más conocidas a nivel mundial se encuentran:

- DJI: Esta compañía fue fundada en el 2006, se encuentra en China y se dedica a la manufactura de sistemas UAV para fotografía y videografía aérea. Además de los drones la empresa produce sus propios controladores y soportes para las cámaras, estos sistemas se consideran muy novedosos y potentes [13]. La compañía no brinda información detallada de los componentes de sus productos y de la programación.
- Parrot: Empresa que se fundó en 1994 en Francia y tiene como objetivo crear, desarrollar y comercializar productos inalámbricos de tecnología avanzada para consumidores y profesionales. La compañía desarrolla para 3 mercados primarios que son el automotriz, conexión de objetos y drones civiles, siendo estos últimos de código abierto por lo que poseen bastante información sobre su hardware y software [14].
- 3D Robotics: Fue fundada en el 2009 en Estados Unidos y es una de las compañías con tecnología más avanzada e inteligente para drones. Esta empresa comercializa las plataformas de piloto automático y gran cantidad de sensores poderosos además de los sistemas UAV [15]. Se basa en la política de código abierto por lo que también brinda información en cuanto a hardware y software de proyectos desarrollados.

2.2.1 Controlador de vuelo

Estas plataformas son las que permiten dotar de una mayor autonomía en vuelo a los sistemas UAV, gracias a estos sistemas el dron puede realizar los desplazamientos y maniobras necesarias mientras se mantienen en el aire. Estos movimientos son maniobrados desde tierra por una estación o por una misión de vuelo preestablecida [16].

Normalmente estos sistemas emplean la técnica de telemetría que consiste en la mediciones de magnitudes físicas y enviar la información a la estación, normalmente se realiza por medio de un enlace de radiofrecuencia [17].

La plataforma de piloto automático al ser la unidad central es la que interactúa con los sensores necesarios para el desarrollo de la tarea durante el vuelo del UAV, en los siguientes puntos se aborda una pequeña explicación de los sensores principales para la implementación de un control de piloto automático.

2.2.2 Sistema de posicionamiento global (GPS)

El GPS (Global Positioning System) es un sistema de navegación compuesto de una flotilla de satélites puestos en órbita por el Departamento de Defensa de los Estados Unidos y sus estaciones en tierra firme. Usando un GPS, uno puede determinar automáticamente su posición (latitud y longitud), altitud y tiempo en la tierra [18].

En el espacio exterior existen alrededor de 24 satélites disponibles de los cuales 3 de ellos utilizan el principio de trilateración tal como se muestra en la figura 2.2 para dar con 2 puntos de la ubicación donde uno es totalmente absurdo, actualmente se utiliza un cuarto satélite para determinar el posicionamiento global con más precisión (latitud, longitud y altitud) [18].

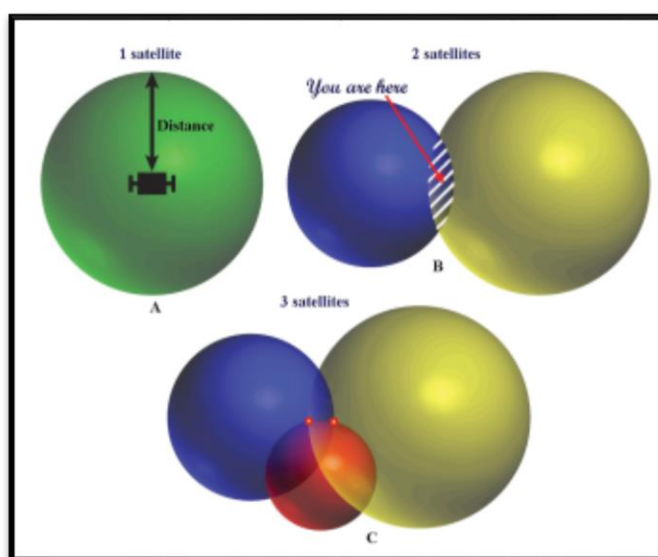


Figura 2.2 Principio de trilateración para un GPS [18].

Cabe destacar que la precisión de los datos de un sistema de posicionamiento global se ve afectada por condiciones como la interferencia atmosférica y reflejos de obstáculos en la tierra como árboles y edificios.

2.2.3 Unidad de medición inercial (IMU)

La IMU (Inertial Measurement Unit) es un dispositivo que se utiliza a estimar la actitud (orientación, velocidad angular y aceleración) de un cuerpo en las dimensiones (X, Y, Z) utilizando sensores inerciales [19]. En nuestro caso el cuerpo es un sistema UAV por lo que se necesita esta unidad para mantenerse estable durante el vuelo y lo realiza con ayuda con el control de vuelo.

Esta unidad de medición inercial normalmente está compuesta por 3 sensores (acelerómetro, giroscopio y magnetómetro), y como los sistemas UAV tienen carga y tamaño limitado necesitan una IMU pequeña y liviana, además de bajo costo es por eso que se utilizan sensores microelectromecánicos (MEMS), pero estos tienen deficiencias tales como el ruido y la deriva [19].

Un giroscopio puede dar las medidas perfectas del movimiento del sistema UAV y por ende su actitud, pero al ser MEMS tiene las deficiencias mencionadas anteriormente y es por eso que se necesitan los otros sensores para corregir estos errores mediante la fusión de los datos por un algoritmo para estimar la actitud del dron.

Para comprender el funcionamiento teórico de una IMU es necesario explicar la función de cada uno de los sensores, los cuales se explican brevemente en los puntos 2.2.4, 2.2.5 y 2.2.6.

2.2.4 Giroscopio

El giroscopio se utiliza para conocer la velocidad de giro de un móvil y por tanto su orientación en los 3 ejes, transformando la fuerza generada por un movimiento angular en una señal eléctrica proporcional a ella, las unidades en que se mide la variación de giro son rad/s [20].

Su funcionamiento se basa en el principio del giroscopio resonador, donde básicamente a través de una estructura resonante (microestructura) se mide la fuerza de Coriolis que se forma cuando se gira, esta variación es cuantificada por una capacitancia y procesada para generar una salida eléctrica. En la figura 2.3 se muestra la fuerza de Coriolis.

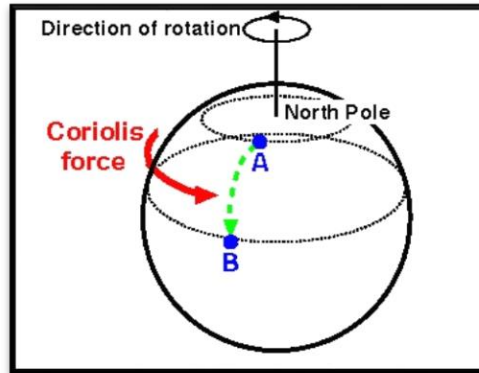


Figura 2.3 Diagrama de la fuerza de Coriolis.

2.2.5 Acelerómetro

La función de este sensor es medir la aceleración de un objeto en movimiento, la medida se da en las unidades de m/s^2 y normalmente trata de un dispositivo analógico que mediante un escalado de la tensión a su salida nos da información de la aceleración en cada eje (X, Y, Z).

La medida de la aceleración se da por medio de la variación de un capacitor en el chip que suspende partículas de silicio ancladas en un punto fijo, que se mueven libremente en el eje de medición. Cuando se presenta una aceleración la masa de partículas se desplaza respecto a su posición de origen y crea un desequilibrio en la capacitancia que se mide y nos da la información de la aceleración en el eje [20]. En la figura 2.4 se muestra el principio físico del acelerómetro.

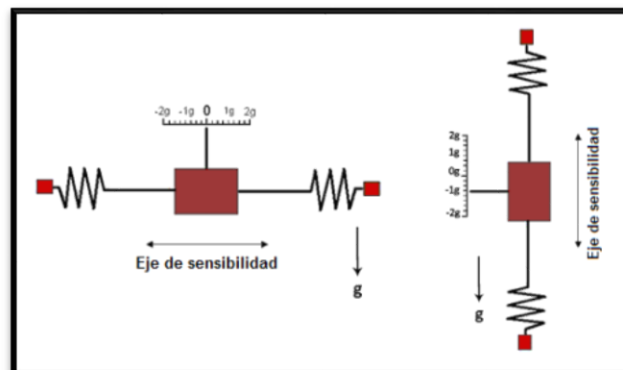


Figura 2.4 Principio físico del acelerómetro [20].

2.2.6 Magnetómetro

Es un sensor que mide la intensidad de campo magnético terrestre en los 3 ejes, gracias a estas 3 medidas se obtiene un vector de campo con el cual se determina la información del ángulo azimutal del sistema UAV [20].

El funcionamiento del magnetómetro se basa en la medición del campo por medio de unas magnetorresistencias que varían el valor de la resistencia al someterse a un campo magnético y normalmente este sensor da el valor en la unidad Tesla (T).

2.2.7 Barómetro

Esta clase de sensor se utiliza para medir la presión atmosférica, miden la variación de la presión estática del dron cuando aumenta o disminuye su altitud, por lo se obtiene una muy buena estimación de la altitud a la que se encuentra el sistema UAV [21].

2.2.8 Orientación del cuerpo en el espacio

La orientación de un cuerpo en el espacio tridimensional está definida por 3 grados de libertad, una forma sencilla para describir la orientación de un objeto respecto a un sistema de referencia es asignarle un nuevo sistema y luego analizar la relación espacial existente entre los 2 sistemas [22].

En la aeronáutica se utiliza el principio de los ángulos de Euler a los cuales se le denominan Roll (Φ), Pitch (θ) y Yaw (ψ), el teorema de rotación de Euler requiere de rotaciones sucesivas alrededor de los 3 ejes del sistema fijo sin realizar 2 rotaciones consecutivas sobre un mismo eje [22].

Los sistemas UAV utilizan este mismo teorema por las componentes brindadas por el IMU, la figura 2.5 muestra la representación de los ángulos de Euler para un dron (cuadricóptero).

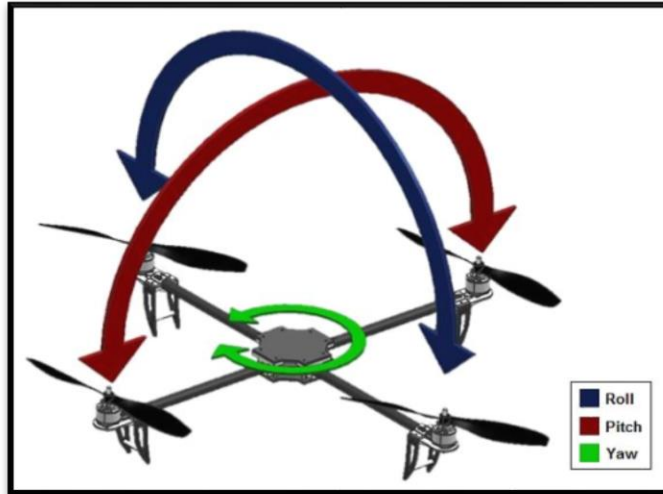


Figura 2.5 Representación de los ángulos de Euler en un sistema UAV [22].

2.2.9 Sistema de alimentación

La alimentación de los sistemas UAV es muy importante ya que de ésta depende la autonomía del dron, actualmente las baterías de polímero de litio (LiPO) son de las más utilizadas.

Las baterías LiPO nacen de una investigación de la conductividad iónica de los polímeros, donde se observa que dentro de sus características se posee una alta constante dieléctrica, además de la capacidad de disolver grandes cantidades de sales de litio y por ende una alta capacidad de corriente eléctrica [4].

Dentro de las características relevantes de las baterías LiPO se pueden mencionar las siguientes [4]:

- Tensión nominal bajo carga de 3 a 4.2 V.
- Auto descarga menor a un 10% por mes.
- Se conforman desde 1 celda hasta 12 celdas.
- Necesidad de un cargador balanceado.

2.2.10 Sistema de comunicación

En la mayoría de los casos el sistema de comunicación entre el dron y la estación base está compuesto por dos enlaces principales denominados enlace de control y enlace de telemetría.

El enlace de control se utiliza para controlar el sistema UAV en modo manual, o también para seleccionar el modo de vuelo. Hay varias marcas de sistemas de radiocontrol como lo es Futaba con su sistema de comunicación S.Bus basado en modulación FHSS (Frequency Hopping Spread Spectrum), además existen otras marcas como lo es Spektrum con el sistema DSM (Digital Spectrum Modulation) con la modulación DSSS (Direct Sequencing Spread Spectrum) y otros con radiocontroles que utilizan la modulación PPM (Pulse Position Modulation) [21].

El enlace de telemetría comunica el UAV con un computador en la estación, donde el ordenador cumple la función de visualizar y analizar los datos de telemetría, y gestionar las órdenes de navegación para el piloto automático, normalmente el protocolo de comunicación utilizado es el MAVlink el cual se detalla en el apartado 2.3.

Otro enlace que se puede resaltar es el llamado enlace de video el cual permite ver la imagen captada por la cámara integrada al sistema UAV desde una pantalla, generalmente utilizan sistemas de transmisión de video analógicos de señales con modulación FM, estos sistemas se conocen como FPV (First Person View) [21].

2.2.11 Plataformas de piloto automático

Actualmente existe una inmensa cantidad de plataformas de piloto automático de diferentes marcas y diferentes características, las plataformas que se mencionan a continuación fueron escogidas por este servidor por el conocimiento previo adquirido, no tienen relevancia alguna en cuanto a importancia ni cantidad de uso, pero sí poseen características importantes y cierta variación en los precios:

A2

Esta plataforma de piloto automático es la más reciente de la empresa DJI donde se mencionó anteriormente es de las marca más poderosas en sistemas UAV, y la describen como un sistema muy completo para aplicaciones de piloto automático comerciales e industriales, además de una localización muy precisa y un perfecto control de vuelo para entornos difíciles para los sistemas UAV [23].

Las plataformas DJI no son código abierto por lo que no traen descripción de su composición y tampoco de los módulos incluidos en el paquete de compra que son: un potente GPS con gran precisión, un IMU muy poderoso con una respuesta estable incluso en entornos de gran vibración y movimiento. También incluye una unidad de alimentación y un módulo de conexión bluetooth con sus respectivos cables de conexión, en la figura 2.6 se muestra la interfaz de dicha plataforma.

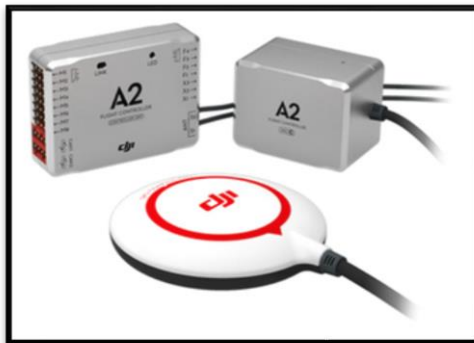


Figura 2.6 Plataforma de piloto automático A2 (esquina superior izquierda), IMU (esquina superior derecha) y GPS [23].

La unidad central cuenta con 2 puertos CAN-Bus, 12 Salidas PWM, Puerto paracaídas, puerto para Receptor S.BUS, puerto receptor DSM. Y al ser cerradas no se puede actualizar el firmware incluido y el software se adquiere en el portal de DJI [24], al ser una unidad muy poderosa y nueva tiene un costo elevado de \$1,299.

CC3D

Es una plataforma de piloto automático que corre el firmware de OpenPilot, es una placa pequeña de código abierto por lo que si trae descripción de sus componentes, el procesador es un STM32 de ARM de 32 bits el cual corre 90 MIPS con 128 KB de Flash y 20 KB de RAM [25].

La placa trae consigo un MPU6000 el cual incluye un giroscopio y acelerómetro, también soporta la conexión de varios receptores como Futaba S.Bus, PPM, DSM. Además cuenta con gran cantidad de entradas PWM configurables y salidas PWM para la conexión a los motores, solo soporta tricópteros, cuadricópteros y hexacópteros [25].

La plataforma en si es muy útil pero no incluye otros módulos por lo que es necesario conseguir los adecuados, y entre los puertos presentes en la placa se encuentran: Un puerto serial principal configurable ya sea para las antenas, GPS u otros, un puerto flexible configurable para diferentes protocolos de comunicación y un puerto USB.

La unidad CC3D cuenta con una EEPROM de 4 Mbits para guardar la configuración del usuario, el software llamado GCS se adquiere en el sitio web de OpenPilot y se encuentra disponible para cualquier versión de Windows, Linux y Mac [25]. Al ser una placa muy pequeña y un poco limitada ya que no trae otros módulos incluidos su precio es de \$24.99, en la figura 2.7 se muestra una imagen del producto.

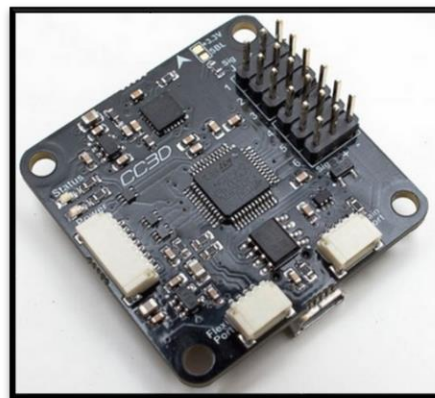


Figura 2.7 Plataforma piloto automático CC3D [25].

AeroQuad 32

Se considera un controlador de vuelo muy completo que contiene un procesador ARM de 32-bits de 168 MHz con 1MB de Flash, 192KB de RAM y FPU. La plataforma es fabricada y distribuida por AeroQuad la cual trabaja también con la idea de código abierto pero como su nombre lo indica es para cuadricópteros. La plataforma es muy estable y trae incluido el MPU 6000 como la mayoría de plataformas, además de un Barómetro y Magnetómetro [26]. En la figura 2.8 se visualiza la placa.

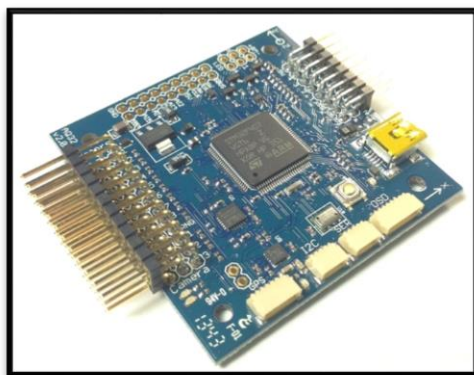


Figura 2.8 Plataforma piloto automático AeroQuad32 [26].

Entre los puertos posee bastantes canales PWM de entrada y salida, además de 3 USART, 1 SPI, 1 I2C, microUSB, microSD, 6 ADC y otros.

La plataforma es compatible con Arduino y trae gran cantidad de documentación en el portal de Aeroquad, así como el software y código disponible para todos los usuarios, se considera una plataforma confiable y estable por lo que tiene un costo de \$149.95.

Como se mencionó previamente existen muchas plataformas de piloto automático, la marca 3DR distribuye algunas que no se plantearon previamente pero se explicarán en el capítulo 3 como parte del diseño.

2.2.12 Plataforma de software (Estación)

Esta unidad cumple la función de analizar y visualizar los datos recibidos por parte del sistema UAV, y también gestionar las órdenes de navegación para el piloto automático como se comentó anteriormente.

Al igual que las plataformas de hardware, existen muchas plataformas de software ya sean de código abierto que mencionaremos como “open source”, generalmente en las “closed source” (contrario a “open source”) el firmware y el software vienen en el paquete con el hardware, en cambio las “open source” son compatibles con varias plataformas de software, donde la misma plataforma provee el firmware adecuado para el sistema UAV (Copter, Plane, Rover).

Algunos ejemplos de estas plataformas son: DJI Ground Control Station [27], OpenPilot GCS [28], Multiwii [29] y otros. En el capítulo 3 se abarca el tema de diferentes plataformas de software consideradas para la solución del problema.

2.3 Protocolos de Comunicación

Las plataformas de piloto automático al estar conectadas con diferentes sensores, éstos poseen diferentes protocolos de comunicación por lo que las plataformas deben contener varios puertos diferentes para ser más flexibles y completas. Entre los protocolos más relevantes en las plataformas de piloto automático se encuentran el UART, SPI, I²C, USB, CAN y MAVLink.

2.3.1 UART (Universal Asynchronous Receiver-Transmitter)

El protocolo de comunicación UART o SCI (Serial Communications Interface) es una interfaz asincrónica que permite el envío de datos entre dispositivos en serie sobre una línea de transmisión y una línea de recepción [30].

Su característica principal es que la línea se encuentra en alto y el protocolo no cuenta con señal de reloj, entonces cuando se necesita enviar información se debe enviar un bit de inicio, seguido de los bits de datos y por ultimo un bit de parada, la figura 2.9 muestra el formato de transmisión de datos [30].



Figura 2.9. Formato de datos transmitidos en el protocolo UART [30].

Este protocolo se caracteriza por su bajo costo y una implementación sencilla en dispositivos ya que solo necesita 2 conexiones (Tx/Rx) [4].

2.3.2 SPI (Serial Peripheral Interface)

Esta interfaz envía los datos en serie, pero a diferencia de la comunicación UART requiere de la sincronización para el envío de datos, por lo que posee una línea de Clock para notificar cuando se enviará la información sincronizada con este mismo [30].

Además este protocolo tiene la característica que permite comunicar un dispositivo Master con varios Slaves, donde el Master decide con que dispositivo intercambiar información por medio de la línea SS (Slave Select), en la figura 2.10 se puede observar el diagrama de conexión del protocolo SPI [4].

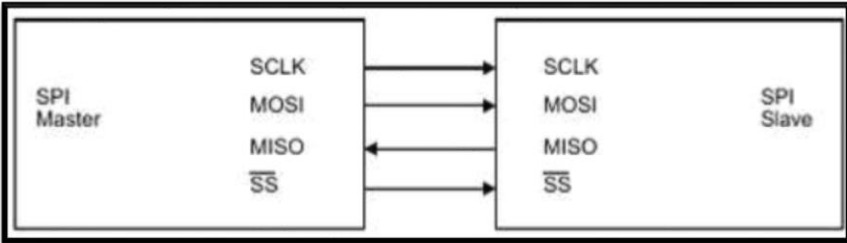


Figura 2.10 Diagrama de interconexión del protocolo SPI [4].

De la figura 2.10 se puede observar las conexiones MISO y MOSI que significan Master Input Slave Output y Master Output Slave Input respectivamente.

2.3.3 I²C (Inter-Integrated Circuit)

Se considera un protocolo más simple en cuanto a la cantidad de líneas implementadas pero resulta que su configuración es más compleja, ya que cuentas solo con 2 líneas SCL (Serial Clock) y SDA (Serial Data) [30].

Entre las características principales de este protocolo se encuentra que se permite la conexión de múltiples Masters y Slaves, donde cada Slave tiene una dirección específica a la cual el Master envía la información. Los paquetes a transmitir están formados por un byte de dirección y luego un byte de datos [30]. En la figura 2.11 se observa la interconexión en el protocolo I²C.

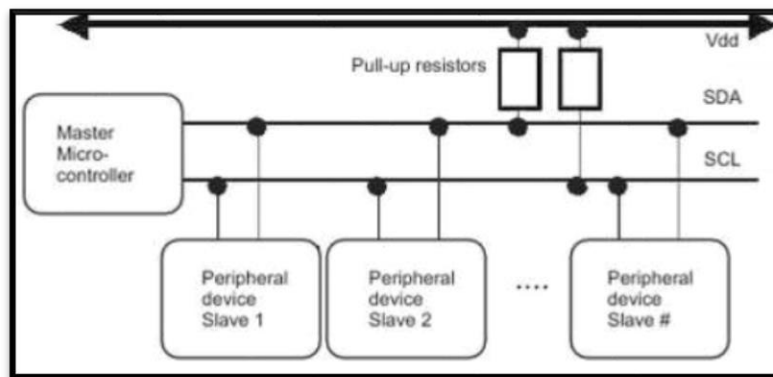


Figura 2.11 Diagrama de interconexión para el protocolo I²C.

2.3.4 USB (Universal Serial Bus)

Es la interfaz por excelencia de las computadoras y dispositivos actuales, aunque es más compleja es muy eficiente para el envío de datos. Se necesita un dispositivo Host que maneja la comunicación y de uno o varios Slaves para la transferencia de datos [30].

Al ser un protocolo más complejo tiene gran cantidad de características de las cuales resaltan [30]:

- Protocolo muy versátil para una gran cantidad de dispositivos (Almacenamiento, Mouse, Teclado y otros).
- Configuración automática, es decir cuando se conecta a un computador se detecta automáticamente, indicando drivers necesarios.
- Cables estándar.
- No requiere suministro de potencia (Provee pin de 5 V).

El protocolo consta de 4 conexiones donde transfiere la señal y alimentación tal y como se muestra en la figura 2.12.

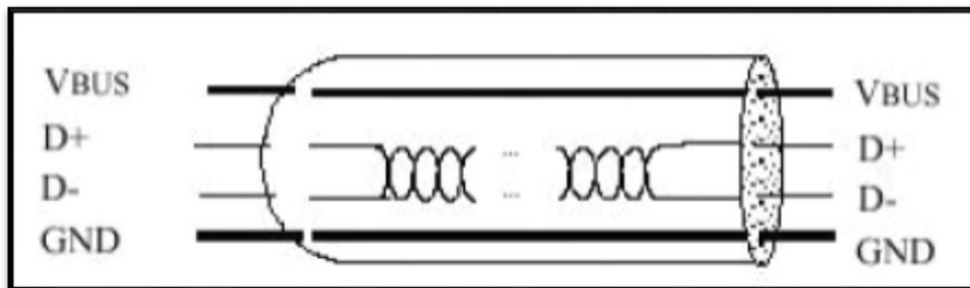


Figura 2.12 Diagrama de interconexión para el protocolo USB [30].

2.3.5 CAN (Controller Area Network)

Trata de un protocolo de comunicación donde todos los dispositivos están conectados con todos por medio de un bus de datos, donde los paquetes enviados se llaman mensajes [31].

Dentro de las características se puede destacar que el protocolo funciona por medio de priorización de mensaje, y además es multiMaster es decir si el bus está libre cualquier nodo puede transmitir a cualquier dirección (todos los nodos reciben la información), además el mensaje contiene información de dirección y priorización [31].

2.3.6 MAVLink

Es el protocolo utilizado para realizar comunicaciones con el sistema UAV, su nombre completo es Micro Air Vehicle Communication Protocol y consiste en la utilización de una biblioteca de cabeceras diseñadas para los sistemas UAV [32].

La información se envía por medio de paquetes llamados mensajes que son básicamente un paquete de bytes codificados por la plataforma de software (estación) para enviarlos a la unidad de piloto automático y viceversa. Cada paquete consta de 17 bytes donde 6 son de cabeceras, 9 de información y 2 de error tal como se puede visualizar en la figura 2.13 [33].

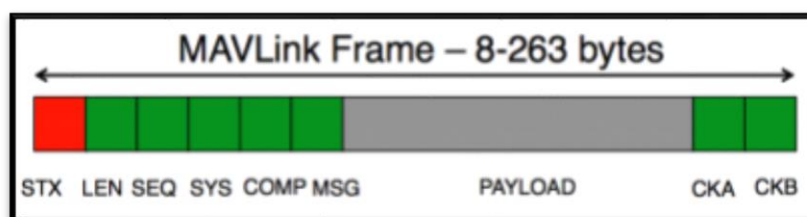


Figura 2.13 Formato de datos transmitidos en el protocolo MAVLink [33].

Capítulo 3: Selección e implementación del sistema de control de vuelo automático

En esta sección se detalla el procedimiento empleado para llegar a la implementación de un sistema de control de vuelo automático en el vehículo aéreo no tripulado multirrotor 3DR-X8+. El procedimiento se basa en la caracterización inicial del mismo sistema UAV con los dispositivos incluidos en el paquete, para así luego determinar una serie de requerimientos de donde se plantean varias soluciones posibles.

Una vez planteadas las posibles soluciones se realiza un análisis comparativo de éstas, con el fin de seleccionar la solución que más se ajuste a las necesidades del proyecto con un mejor rendimiento. Seleccionada la solución se procede a realizar un análisis y selección de los componentes necesarios para el desarrollo de ésta, para finalmente implementarla en el sistema UAV.

3.1 Caracterización del sistema UAV

El sistema seleccionado para implementar el control de piloto automático es el 3DR-X8+ el cual se adquiere por medio de la tienda en el portal de la compañía 3D Robotics [34]. El sistema trae consigo una serie de módulos incluidos en el paquete comprado por lo que es importante citar las características de los dispositivos y del mismo dron.

3.1.1 X8+

El UAV X8+ es una plataforma creada por 3D Robotics, a la cual la describen como una herramienta inteligente, portable y resistente para realizar varias tareas con cargas personalizadas. Su diseño está optimizado para el uso de una cámara GoPro con el fin adquirir fotos aéreas y videos, pero es capaz de soportar diferentes equipos de cámaras, sensores y mucho más [35].

El dron está hecho de aluminio lo que lo hace bastante resistente y suficientemente ligero para maximizar la movilidad, además se clasifica como un octocóptero ya que cuenta

con 8 hélices distribuidas en 4 extremidades (2 en cada extremidad). En la figura 3.1 se muestra el sistema UAV modelo X8+.



Figura 3.1 Sistema UAV X8+ de 3DR [34].

Como se mencionó en el marco teórico la compañía 3DR se rige en la política de “open source” por lo que se puede adquirir gran información acerca de sus productos. En el caso del modelo X8+ cuenta con las siguientes especificaciones [34]:

- Dimensiones: 35 cm x 51 cm x 20 cm y peso: 2.56 kg.
- Capacidad de carga de 800 g (Es posible más de 1 kg pero reduce tiempo de vuelo).
- Tiempo de vuelo 15 máximo sin carga extra.
- Frecuencia de operación: 433 MHz.
- Batería LiPo 4S 14.8V 10 Ah 10C y pesa 803 g (Cargador Incluido).
- Motores: eight SunnySky V2216-12 KV800 II.
- El paquete incluye GPS con magnetómetro, antenas para telemetría, control FlySky FS-TH9X.
- Frame Type: X (Importante para la configuración de piloto automático).
- Incluye Manual de Operación [36].

El manual de operación contiene información relevante sobre el uso del sistema UAV X8+, como el ensamblaje de algunas partes, la conexión de la batería, encender y apagar los motores (arm/disarm), utilización del control y algunos consejos de seguridad [36].

De las especificaciones anteriores es importante detallar algunos módulos incluidos en el paquete ya que cumplen una función importante en el desarrollo de la solución:

3.1.2 Control FlySky FS-TH9X

Esta unidad se encarga principalmente del manejo manual del sistema UAV, se alimenta de 8 baterías AA que necesitan ser cambiadas a los 9.1 V (el control lo notifica), también muestra información de vuelo del estado del dron [36]. La calibración del control se realiza por medio de la plataforma de software, donde se puede configurar modos de vuelo predeterminados entre ellos el vuelo automático y el manejo de la suspensión de Cardán o llamado en inglés como “Gimbal” (un eje).

El control trae consigo un kit de FPV (First Person View) para la cámara GoPro que incluye una pantalla para visualizar el video, un transmisor de video que trabaja con la frecuencia de 5.8 GHz, y sus respectivas antenas. En la figura 3.2 se muestra el control con el respectivo kit acoplado [37].



Figura 3.2 Control FlySky FS-TH9X con pantalla del kit FPV [37].

Cabe mencionar que la suspensión de Cardán que se menciona de ahora en adelante como “Gimbal” incluido en el sistema es el Tarot Gimbal que está diseñado especialmente para las cámaras GoPro Hero y su movimiento se da en 2 ejes, es un dispositivo muy estable y preciso en su movimiento debido a la acción del giroscopio y acelerómetro procesados por la unidad central que contiene la cual es un procesador ARM de 32 bits [38].

Como se mencionó anteriormente el sistema UAV utilizado soporta otros tipos de “Gimbal” y cámaras con mejor desempeño, por lo que la utilización del Tarot Gimbal y la cámara GoPro se utilizarán para la validación del uso de estos dispositivos en una misión de vuelo pero no para la meta del proyecto.

3.1.3 Batería LiPO 4S

Es el sistema de alimentación del UAV y de los módulos adicionales del sistema, la batería tiene una tensión nominal de 14.8 V y está compuesta por 4 celdas, una carga 10000 mAh y pesa 803 g (peso ya está contemplado en el modelo X8+), incluye un cargador balanceado donde se llega a cargar máximo a 16.8 V (4.2 V cada celda). La figura 3.3 ilustra la batería a utilizar.



Figura 3.3 Batería LiPO 4S del sistema UAV.

Si la batería se encuentra totalmente cargada dura aproximadamente 15 minutos de vuelo, dependiendo de la carga, condiciones del ambiente y tipo de vuelo, con gran cantidad de equipo baja su rendimiento a 5 minutos, la batería se puede utilizar hasta los 14.1 V y de ahí es necesario cargarla de nuevo.

El control le indica la información de carga en el vuelo y el sistema UAV tiene un procedimiento de seguridad de aterrizar en caso de batería baja. El estado de la batería normalmente es visible en las plataformas de software (Estación).

El proceso de carga de estas baterías se realiza por medio de un cargador especializado el cual puede realizar distintos tipos de cargas para baterías de diferentes celdas. Entre los tipos de carga se pueden destacar el proceso de carga balanceado que se utiliza cuando se alimenta el dron en una misión de vuelo (16.8 V, 4.2 V cada celda) y también se encuentra el proceso de carga de almacenamiento cuando no se va a utilizar la batería por un gran periodo de tiempo.

3.1.4 Radios de telemetría 3DR

La función de estos dispositivos es crear el enlace de datos en tiempo real entre la estación y la plataforma de piloto automático, la frecuencia de operación es de 433 MHz. Entre las características de este sistema de comunicación se puede mencionar que son ajustables en varios parámetros como la potencia máxima de salida, ciclo de trabajo y otros [39].

La conexión de estos se da por medio de micro-USB (3.7 – 6 VDC) en el caso de la estación, y en el caso de la plataforma de piloto automático tiene un puerto de 6 pines Serie (UART, 3.3 V) y como se mencionó en el marco teórico el protocolo de comunicación entre ellos es el MAVLink [40]. La figura 3.4 muestra los radios a utilizar.



Figura 3.4 Radios de telemetría 3DR [40].

El rango en cuanto a la conexión de estos ronda entre los 300 a 500 metros dependiendo de las interferencias de otras fuentes, asimismo se puede mejorar el rango con el uso de otras antenas, o disminuyendo la tasa de transferencia de datos [39].

3.1.5 GPS con magnetómetro

Este sensor también es distribuido por la empresa 3DR y lo catalogan como un GPS de gran precisión y alto rendimiento que con brinda muchas características en la placa, entre ellas la inclusión de un magnetómetro digital alejado de las interferencias magnéticas del vehículo para un mejor desempeño. El dispositivo se actualiza a la frecuencia de 5 Hz y es configurable por medio de la estación, por lo que contiene una EEPROM para guardar configuración, además el módulo GPS (u-blox NEO-7) posee una precisión de ± 2 m y el magnetómetro (HMC5883L) con una precisión de $\pm 1^\circ$ [41].

En cuanto a la conexión de este sensor requiere 2 protocolos, para conectar el GPS necesita un puerto de 6 pines Serie (UART), y en cuanto al magnetómetro se comunica I²C (4 pines). Además cuenta con varios filtros y un regulador de 3.3 V, todo se encuentra en un estuche protector tal como se observa en la figura 3.5 [42].



Figura 3.5 GPS con magnetómetro 3DR [42].

Por medio de la estación se puede realizar la calibración del magnetómetro para una mejor orientación en cuanto al campo magnético y también se puede configurar el GPS del cual se recomienda utilizar la configuración por defecto.

Los módulos mencionados previamente al venir incluidos en el paquete forman parte del desarrollo de la solución sin comparación alguna, quedando solamente por cubrir la sección de estabilización del sistema a partir de una IMU, y principalmente la plataforma de piloto automático la cual se puede considerar como la unidad principal ya que interactúa con los módulos citados, y por ultimo seleccionar la plataforma de software para programar la unidad de piloto automático, realizar las tareas de configuración y calibración de varios módulos.

Con los módulos detallados anteriormente es posible determinar una serie de requerimientos que debe incluir la plataforma de piloto automático, en la tabla 3.1 se muestran ciertos requerimientos que se deben contemplar en el momento de implementar la plataforma de piloto automático.

Tabla 3.1 Requerimientos de la plataforma de piloto automático.

| Requerimientos | Justificación |
|--|---|
| Gran Capacidad de Procesamiento | Al ser un sistema de control de vuelo debe tener poseer un sistema operativo de tiempo real (RTOS), ya que debe de dar respuesta a diferentes situaciones con la gran cantidad de sensores conectados y realizarlas en el mejor tiempo posible. |
| Bajo Peso | El diseño debe cumplir con un peso menor a los 500 g para un mejor rendimiento del dron. |
| Puerto de Alimentación Batería 4S | La plataforma al estar montada en el propio sistema UAV, necesita alimentarse de la misma batería del sistema. |
| Conexión con Control, Telemetría, GPS, Magnetómetro y otros | Debe poseer varios puertos de comunicación entre ellos (3 Serie, I ² C, etc) y puertos adicionales. |
| Mínimo 10 Salidas PWM | Estas salidas son necesarias para el manejo de los motores del sistema (8) y del “Gimbal”. |
| Programable por medio de la estación | Es importante que la plataforma sea programable por medio de un software especializado y además posea una interfaz donde muestre los datos durante el desarrollo de la misión. |

3.2 Planteamiento y selección de la solución final

Actualmente existe una gran cantidad de plataformas de piloto automático comerciales con características diferentes, como por ejemplo algunas plataformas incluyen la unidad de medición inercial en el sistema, por lo que contienen por defecto el debido procesamiento de los datos del sensor facilitando la configuración de la plataforma, por eso la selección de este sensor que juega un papel fundamental en el sistema.

Con respecto a la lista de requerimientos y análisis del problema planteado se formularon 3 posibles soluciones que sean capaces de implementar un control de vuelo automático para el sistema UAV 3DR-X8+.

3.2.1 Solución 1: Plataformas comerciales de fotogrametría para UAV.

Se plantea esta solución como la más simple ya que consiste en la selección de una plataforma comercial diseñada específicamente para labores de fotogrametría, normalmente estas plataformas son “closed source” y “closed hardware” por lo que son poco flexibles, además son limitadas en cuanto al uso de hardware ya que normalmente incluyen en el paquete de compra los sensores únicamente compatibles con la unidad central por lo que se descartarían automáticamente los módulos detallados en la sección anterior y casi considerando también la opción de descartar el sistema UAV previamente definido.

Estas plataformas traen por defecto el uso de únicamente una plataforma de programación por lo que se puede decir que es un sistema que se utiliza básicamente conectándolo y usándolo (término conocido como “plug and play”), es decir nada más se sincroniza con la estación, se programa y se utiliza, sin poder realizar ninguna modificación de hardware, y acceso limitado al software.

El costo ronda por los \$500 hasta los \$1500, debido a que incluyen los sensores a utilizar, y ya que poseen por defecto una gran capacidad de procesamiento para tareas de fotogrametría.

3.2.2 Solución 2: Plataformas comerciales basadas en “open source” y “open hardware”.

Este tipo de plataformas se pueden considerar más flexibles con respecto a la solución 1 ya que con sus políticas de “open source” y “open hardware” se cuenta con una inmensa cantidad de información sobre éstas, y generalmente con gran cantidad de software libre disponible en repositorios.

Se pueden clasificar con plataformas de propósito general ya que se pueden configurar de diferentes maneras dependiendo de la necesidad del usuario, estas plataformas son compatibles con gran cantidad de módulos y no necesariamente de la misma marca o distribuidora.

En cuanto a la estación de programación, existen plataformas compatibles con varias estaciones por lo que se puede seleccionar la que más se ajuste a las necesidades del usuario, esto se logra a que soportan varios firmwares lo que aumenta su flexibilidad.

El precio de estas unidades ronda entre los \$40 hasta los \$500 dólares, dependiendo de las características y de los modelos incluidos, por lo que hay que considerar los requerimientos mencionados.

3.2.3 Solución 3: Diseño de plataforma de piloto automático.

Esta solución se basa en el diseño y desarrollo de la unidad de piloto automático, el proceso se define principalmente como el diseño de un sistema embebido partiendo de los requerimientos a diferencia de las soluciones anteriores que se selecciona una plataforma que cumpla los requerimientos.

Entre las ventajas se encuentra el menor consumo de potencia y peso ya que sería la solución más óptima, pero trae como gran desventaja el tiempo que la solución necesitaría ya que son tantos aspectos a considerar además del ensamble, interconexión y pruebas.

Al ser una plataforma totalmente personalizada surge la necesidad de realizar un programa de estación totalmente personalizada, desarrollando una interfaz con interacción

cómoda con el usuario, y la programación de esta misma y todos los dispositivos de hardware utilizados por lo que incluye mucho más consumo de tiempo.

La opción de la unidad de medición inercial incluida en la selección del sistema queda totalmente descartada en esta solución por lo que es un sensor más de carácter externo.

3.2.4 Análisis y selección de la solución

De las soluciones planteadas es necesario realizar una comparación en varios aspectos críticos en el desarrollo del proyecto, para llegar a la selección de la propuesta más viable y que tenga el mejor rendimiento. Entre los aspectos a considerar se encuentran la funcionalidad, costo, flexibilidad y tiempo de implementación.

Funcionalidad: Se puede considerar esta característica como la menos importante ya que las 3 soluciones poseen un gran nivel de funcionalidad y se pueden utilizar en tareas de fotogrametría, dependiendo de la solución el desarrollo de dicha característica está ligada al tiempo de implementación pero el fin sigue siendo el mismo.

Costo: El aspecto económico es importante de considerar y más cuando hay gran diferencia de precios entre las soluciones, en cuanto a la solución 1 tiene un costo bastante elevado por el hecho de comprar una plataforma que realiza específicamente la tarea de fotogrametría con los sensores necesarios incluidos en la compra, además el costo aumenta debido a la facilidad de uso de estas unidades comerciales ya que prácticamente como se citó anteriormente son “plug and play”, la única tarea que realiza el usuario es programar la ruta y acciones durante la misión. La solución 2 disminuye su costo considerablemente ya que es una plataforma más general que puede cumplir varias tareas dependiendo del usuario, no es tan intuitiva como la solución 1 ya que normalmente éstas aceptan la conexión de una gran cantidad de dispositivos de diferentes características, y al ser “open source” y “open hardware” disminuye su costo con el fin de ayudar a los investigadores en el área de UAV. El costo de la propuesta 3 no está definido de manera clara ya que existe muchos factores que hay que tomar en cuenta como la compra de componentes ya sea de la unidad central y

periféricos, circuitos integrados, procesos de soldadura, interconexiones y otros insumos extras que no se habían contemplado antes de la propuesta de diseño por lo que su costo puede considerarse más caro que la solución 2 y en teoría más barato que la solución 1.

Flexibilidad: A flexibilidad se refiere a que con ayuda de la plataforma se puedan realizar modificaciones en cuanto al hardware del equipo, software y propio firmware de la unidad central para así mejorarla a futuro. La flexibilidad en la solución 1 es nula, es decir al ser un sistema totalmente específico la mayoría de veces resultan ser cerrados por lo que resulta casi imposible realizar cambios en el software y hardware. A diferencia de la solución anterior la solución 2 brinda mayor flexibilidad con el simple hecho de ser plataformas “open source” y “open hardware” por lo que son muy flexibles en cuanto a cambios ya sea por correcciones o mejoras. La solución 3 al ser una plataforma diseñada y fabricada por uno mismo para un fin específico, difícilmente va a ser flexible en futuro a largo plazo, pero sí se pueden considerar mejoras en un futuro cercano por lo que sí se puede considerar un algo flexible.

Tiempo de implementación: Este aspecto es muy importante en el desarrollo del proyecto ya que solo se cuentan con 16 semanas de elaboración, por lo que es importante definir un tiempo estimado de cada solución. La solución 1 se considera la propuesta con el menor tiempo posible ya que solo consiste en la compra de la plataforma con sus respectivos sensores, conexión al sistema y programación con el único software disponible, es lo único posible de realizar ya que al ser en su mayoría sistemas cerrados no se cuenta con mucha información de los dispositivos. El tiempo en la solución 2 es mayor ya que requiere más elaboración en cuanto al análisis y selección de los componentes restantes, se necesita considerar además el estudio de los alcances de la misma plataforma seleccionada ya que al ser abierta cuenta con gran cantidad de información al respecto, la misma selección de las plataformas de software compatibles con su análisis previo, además las tareas como la implementación del sistema completo, programación de rutinas y por último la documentación de manuales de uso; todo esto es posible realizarse dentro de las 16 semanas sin la presencia de algún inconveniente de fuerza mayor que retrase el desarrollo del proyecto. Y para finalizar la comparación se encuentra la solución 3 donde tiene bastante

desventaja en el tiempo de implementación ya que se considera casi imposible su desarrollo en el lapso establecido ya que existe una gran cantidad de detalles a considerar, además del mismo desarrollo de la estación por lo que aumenta el tiempo en esta solución por lo que se puede determinar que queda prácticamente descartada por este aspecto.

La tabla 3.2 califica los aspectos detallados anteriormente de forma resumida para cada propuesta de solución con el fin de seleccionar la mejor solución, el criterio de la calificación se desarrolló por este servidor basado en los conocimientos previos y análisis propio. Cabe destacar que la calificación se da del 1 al 5 donde 5 es la mejor puntuación.

Tabla 3.2 Calificación de las soluciones planteadas.

| Aspecto | Solución 1 | Solución 2 | Solución 3 |
|---------------------------------|-------------------|-------------------|-------------------|
| Funcionalidad | Buena (5) | Buena (5) | Buena (5) |
| Costo | Muy elevado (1) | Normal (5) | Elevado (3) |
| Flexibilidad | Nula (1) | Buena (4) | Limitada (2) |
| Tiempo de implementación | Poco (2) | Normal (4) | Mucho (1) |
| Ponderación | 9 | 18 | 11 |

Basado en la comparación explicada en los aspectos definidos y la calificación dada en la tabla 3.2 se considera la solución 2 como la mejor opción para el desarrollo del proyecto ya que tiene gran ventaja sobre las otras soluciones planteadas en todos los factores considerados. Una vez seleccionada la solución es posible realizar la descripción del propio diseño donde se parte de la misma selección de la unidad central y de ser necesario un módulo adicional, además de la estación para la debida programación de rutas.

3.3 Descripción del proceso de diseño

Al haber seleccionado la solución 2 en donde se utilizará una plataforma comercial basada en “open source” y “open hardware” que sea compatible con los módulos previamente mencionados y que cumpla los requerimientos de la tabla 3.1 es posible realizar un diagrama modular de la implementación de la plataforma con los componentes para mostrar una idea del funcionamiento del sistema de control de vuelo automático para un sistema UAV. En la figura 3.6 se muestra dicho diagrama con la conexión a los módulos necesarios para el desarrollo del control.

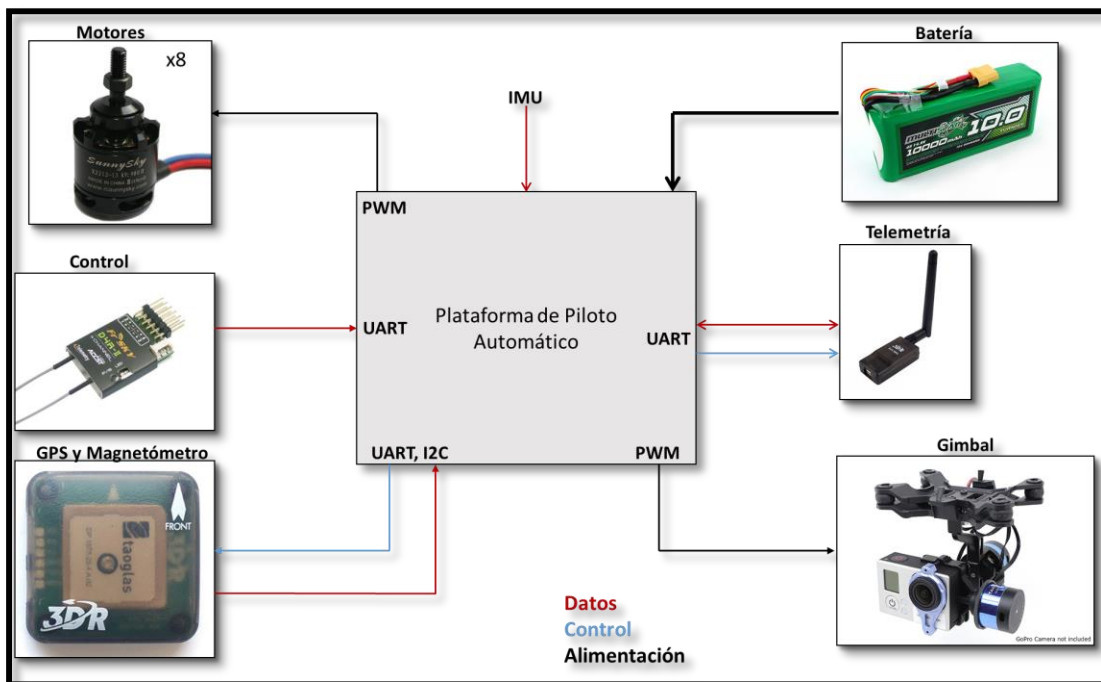


Figura 3.6 Diagrama modular del control de vuelo automático para el sistema X8+.

Contemplado todo lo explicado anteriormente es necesario seleccionar la plataforma de piloto automático, considerando principalmente la tabla 3.1 con los requerimientos de dicha plataforma.

3.3.1 Plataforma de piloto automático

La misma empresa 3DR brinda 3 plataformas de piloto automático que se pueden denominar como las 3 plataformas principales ya que al ser de la misma empresa están diseñadas para los modelos desarrollados en ella; existen otras plataformas de otras compañías las cuales puede funcionar o imitaciones con pequeñas variaciones las cuales no se contemplan en el análisis y es mejor descartarlas para evitar problemas de compatibilidad ya sea con el sistema UAV o los sensores.

Las 3 plataformas de piloto automático que distribuye la empresa 3DR son el APM 2.6, PX4 y el Pixhawk, siendo ésta última la más novedosa y más completa para el desarrollo del proyecto por lo cual fue seleccionada. La comparación y análisis entre las 3 plataformas se encuentra en el Apéndice A.1, por lo que aquí solo detallaremos las características del Pixhawk para resaltar que cumple con los requerimientos de la tabla 3.1.

El Pixhawk es una plataforma relativamente nueva diseñada por el proyecto PX4 y manufacturada por 3DR desde el año 2014, es una plataforma de tiempo real y multihilo con un entorno de programación Unix/Linux dando un rendimiento increíble, flexibilidad y confiabilidad para manejar cualquier tipo de vehículo autónomo [43]. A continuación se muestra una lista de las características más importantes:

- Procesador: ARM Cortex-M4 con FPU (32-bits, 168 MHz, 256 KB RAM y 2 MB Flash).
- Procesador de respaldo ARM Cortex-M3 (32 bits, 24 MHz).
- Incluye MPU 6000 (IMU) y Barómetro (Configurables y Calibración).
- Puertos: 5 Serie (UART), 2 CAN, I²C, 14 PWM, SPI, SD, MicroUSB, 3.3 y 6.6 V ADC y Pines entrada y salida (Digitales y analógicas) y otros.
- Sistema Operativo NuttX (Tiempo real, flexible y confiable).
- Incluye un Parlante, Botón de seguridad, Tarjeta SD y LEDs indicadores.
- Compatible con 2 firmwares: PX4 y APM.

Dentro de las características resalta la inclusión de la IMU en la misma plataforma, el MPU 6000 es un sensor que contiene un giroscopio y un acelerómetro en los 3 ejes. El giroscopio incluido tiene una salida de datos digital de 16 bits en un rango de ± 250 hasta ± 2000 dps y el acelerómetro en un rango de ± 2 a ± 16 g con una salida 16 bits también [44].

Además del MPU 6000 la unidad cuenta con el barómetro MEAS MS5611 el cual tiene una salida digital de 24 bits con un rango de 10 a 1200 hPa que en altura respecto al nivel del mar se transforman en 26000 a -1500 m [45]. Y el Pixhawk cuenta con un acelerómetro de 14 bits y un giroscopio de 16 bits de respaldo por lo que se encuentra bien equipado en cuanto a los sensores restantes.

Otra característica importante son los puertos incluidos ya que tiene una enorme cantidad por lo que cubre el requisito de puertos mínimos, y la ventaja de estos puertos es que detectan automáticamente los módulos y los configura por lo que es totalmente compatible con los módulos detallados anteriormente y además en la misma plataforma y con ayuda del software de la estación se puede realizar la calibración de los sensores externos e internos de la plataforma.

La plataforma posee 14 salidas PWM de las cuales 8 son principales con las que se controlan los motores o llamadas ESC (Controladores de velocidad electrónicos) los cuales se pueden calibrar, y las otras 6 salidas se denominan auxiliares que se pueden utilizar para manejar los servomotores de un “Gimbal” o el disparador de una cámara.

Existen otros puertos disponibles en el Pixhawk como la entrada del receptor de satélite que principalmente se utiliza para el uso de transmisores de la marca Spektrum, también cuenta con el puerto Futaba para los controles de esta marca, lo mismo sucede con la entrada PPM.

El Pixhawk al ser concebido para sistemas UAV incluye en el paquete de compra con un regulador de voltaje para las baterías 3S y 4S con una salida de 5V ya que el puerto de alimentación que funciona la plataforma es 4 V a 6 V de entrada aproximadamente, por lo que cumple otro de los requisitos predefinidos.

Las dimensiones del Pixhawk son bastante pequeñas para una unidad lógica tan poderosa con gran cantidad de puertos, ya que su ancho es de 50 mm con 15.5 mm de altura y un largo de 81.5 mm y lo que la hace muy ligera con un peso de 38 g por lo que el resto de componentes debe pesar menos que 462 g. La plataforma posee una interfaz muy intuitiva en la cual los puertos se encuentran debidamente asignados tal como se observa en la figura 3.7.

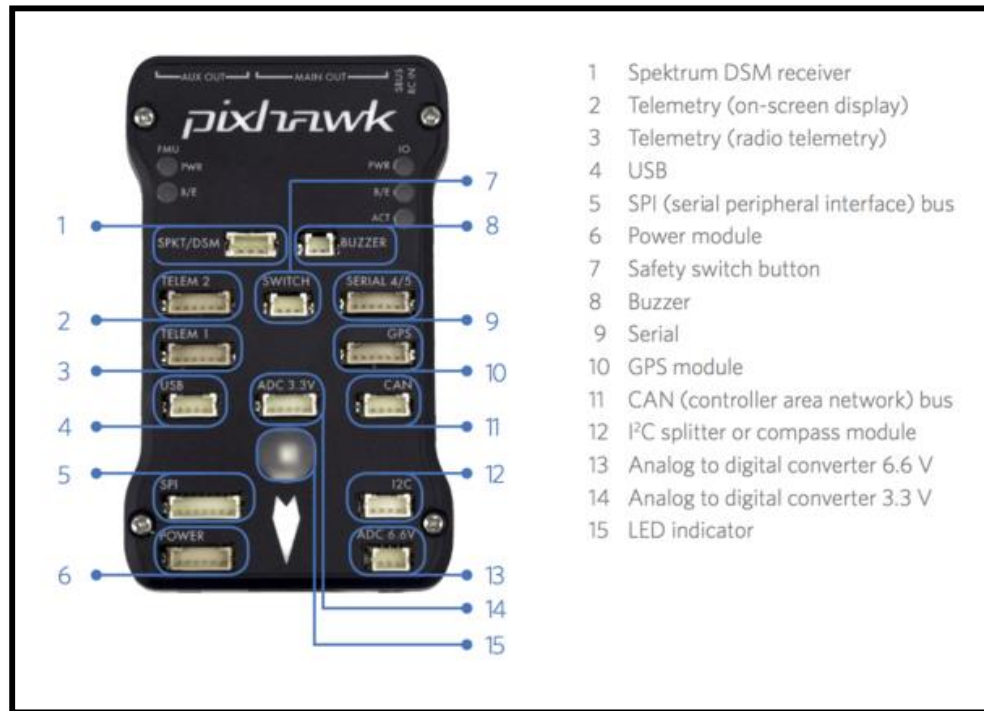


Figura 3.7 Plataforma de piloto automático Pixhawk.

En cuanto al parlante cumple la función de notificar cualquier acción en la plataforma con sonidos predeterminados, al igual que los LEDs que son indicadores visuales, también el botón de seguridad funciona principalmente para evitar el armado de los motores no planeados y finalmente la tarjeta SD cumple varias funciones de almacenamiento de misiones, datos de misiones y otros usos configurables.

La característica de que el Pixhawk es compatible con 2 firmwares toma gran relevancia en el proceso de diseño del proyecto ya que la elección del firmware está directamente ligada con la elección de la estación que es el siguiente paso el desarrollo de la solución.

3.3.2 Plataforma de software de piloto automático (Estación)

El Pixhawk soporta dos firmwares diferentes que son el PX4 [46] y APM [47], donde el PX4 es un software de control creado por el proyecto PX4 y el APM es un software creado por el proyecto ArduPilot/APM que trabaja en conjunto con 3DR en sus productos. Para aclarar la función del firmware es el código base que se encuentra en el hardware, el cual configura la plataforma al tipo de vehículo a utilizar, es decir contiene las rutinas para el control del sistema. Además de que la funcionalidad se basa en la interpretación de los comandos recibidos, y la misma configuración de rutas, y es por eso que cada actualización del firmware lo que se pretende mejorar es la funcionalidad ya que expande la cantidad de comandos soportados para realizar diferentes actividades.

Cada software tiene su debida plataforma de programación, es decir el PX4 trabaja con la plataforma QGroundControl [48] y APM trabaja con Mission Planner o APM Planner dependiendo del sistema operativo [49]. En el desarrollo del proyecto se seleccionó la plataforma Mission Planner con el debido firmware de APM específicamente llamado APMCopter cuando se trabaja con vehículos multirrotores. (El debido análisis y selección de la plataforma se encuentra en el Apéndice A.1).

Mission Planner es un programa de estación terrestre con gran cantidad de características especiales bajo el software de ArduPilot. Esta plataforma puede controlar distintos vehículos como aviones, multirrotores o vehículos terrestres.

La interfaz de Mission Planner es bastante intuitiva y además es perfecta para los nuevos usuarios de sistemas autónomos, en nuestro caso es un sistema UAV y la plataforma es únicamente compatible con Windows. La figura 3.8 muestra dicha interfaz en su pantalla inicial cuando se utiliza el programa.



Figura 3.8 Interfaz de la plataforma Mission Planner.

Esta plataforma se puede utilizar como suplemento del control dinámico de cualquier vehículo o también únicamente como una plataforma de configuración y programación del sistema [49]. Entre las funciones que puede realizar Mission Planner se encuentran:

- Carga del debido firmware en la plataforma de piloto automático.
- Configuración y calibración del sistema para un rendimiento óptimo.
- Creación de misiones de vuelo por ubicación de puntos mediante Google Maps (por defecto).
- Análisis de los datos de la misión de vuelo creados por el mismo sistema.
- Control manual del vehículo.
- Monitoreo del estado del sistema UAV durante la misión de vuelo.
- Interacción de la interfaz con un simulador de vuelo SITL.
- Georreferenciación de imágenes.
- Programación mediante scripts de Python.

De las funciones mencionadas anteriormente es importante decir que en las misiones de vuelo además de programar las rutas se pueden controlar otros factores como el mismo “Gimbal” lo cual es un punto importante en el control de la plataforma de piloto automático.

Otra función de gran importancia es la georreferenciación de imágenes que se puede realizar por medio de Mission Planner, donde después de una misión de vuelo el usuario selecciona la carpeta de fotografías obtenidas y luego la información de vuelo obtenida por los radios de telemetría.

En cuanto el desarrollo de scripts de Python es un punto importante de explorar ya que se puede automatizar de cierta manera la creación de misiones de vuelo o el propio manejo del sistema UAV por medio de Mission Planner.

Ya que se cuenta con la plataforma de piloto automático se puede proceder a implementarla en el sistema UAV y realizar la conexión con los demás módulos para luego realizar la conexión a la estación y cargar la debida programación a la misma plataforma.

3.3.3 Implementación de la plataforma de piloto automático

El proceso de implementación de la plataforma de piloto automático y los sensores al sistema UAV resulta un procedimiento relativamente fácil debido a la interfaz de la propia plataforma de piloto automático para realizar las conexiones sin problema alguno y los demás módulos al venir en el paquete de compra con el X8+ vienen ubicados en espacios específicos para el comportamiento óptimo de ellos y ende del sistema UAV.

Las conexiones que se realizaron según los números de la figura 3.7 son:

- 2. Receptor del Control FlySky FS-TH9X
- 3. Radio de Telemetría 3DR
- 6. Batería 4S
- 7. Botón de Seguridad
- 8. Parlante
- 10. GPS
- 12. Divisor de I²C
- Motores a salidas PWM Main Out
- “Gimbal” a salidas PWM Aux Out
- RC In al receptor del Control FlySky FS-TH9X

Cabe destacar que dicha plataforma tiene la capacidad de conectar en el puerto I²C un divisor para aumentar la cantidad de puertos I²C, aunque por el momento solo requiera el único puerto para el magnetómetro.

Una vez conectados todos los dispositivos a la plataforma cumpliendo el diseño de la figura 3.6, ésta se adapta en el espacio diseñado en el X8+ el cual queda completo en cuanto a la implementación del hardware tal como se observa en la figura 3.9.



Figura 3.9 Plataforma Pixhawk conectada y montada en el X8+.

Por razones de seguridad en los vuelos y de protección a la plataforma de piloto automático se le ensambla una placa de protección encima de ella, la cual venía incluida con el sistema UAV, en la figura 3.10 se muestra el sistema UAV completo en donde se puede visualizar tal protección.

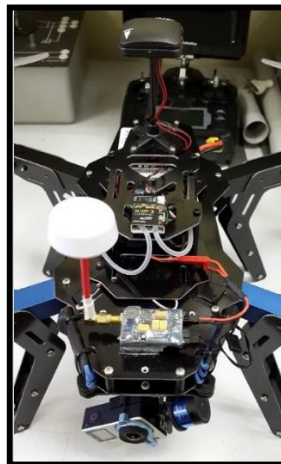


Figura 3.10 Sistema UAV X8+ completo.

Ya que se posee el sistema UAV con el sistema de control de piloto automático implementado y suponiendo el debido comportamiento de cada uno de los módulos en el sistema, se puede proceder a la programación de la plataforma por medio de Mission Planner y Python.

3.3.4 Descripción de la programación de las misiones de vuelo autónomo

La programación de las rutas de vuelo por medio de Mission Planner toma en cuenta muchos factores, ya que se puede realizar por la ubicación de puntos en Google Maps, cada punto considera varios aspectos medidas con los sensores. La figura 3.11 muestra los parámetros de cada punto de programación y la indicación de los puntos en el mapa.

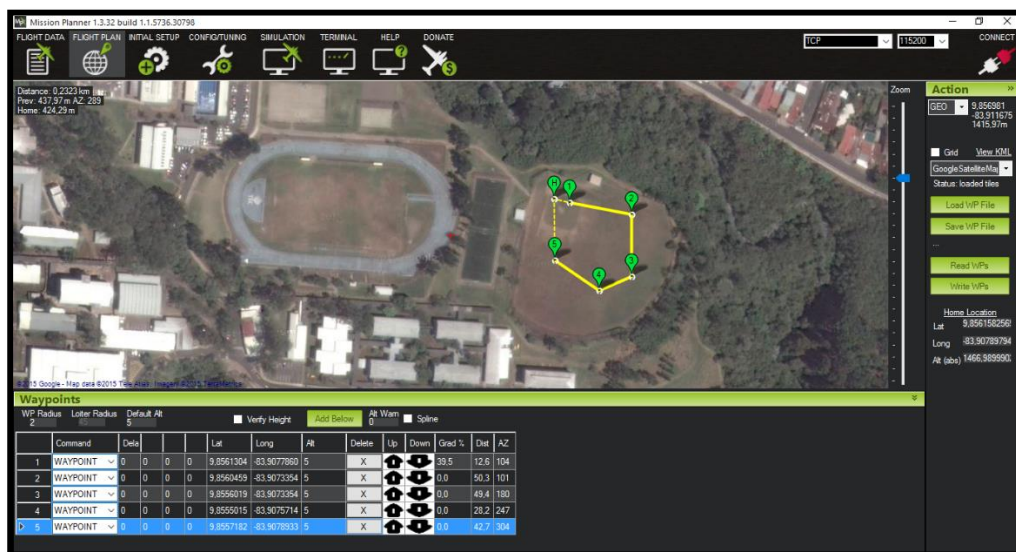


Figura 3.11 Interfaz de la programación de rutas con Mission Planner.

Entre los parámetros más importantes a considerar siempre son la latitud, longitud y altitud tal como se observa en la parte inferior figura 3.11 ya que determinan la ubicación del punto a ir y la altura deseada, en cuanto al delay se refiere a la cantidad de tiempo que debe de permanecer en cierto punto para luego ir a otro.

Básicamente la tarea del sistema UAV en una misión se resume en elevación a cierta altura, cumplimiento de ruta programada y aterrizaje. En tareas de fotogrametría debe cumplir los mismos puntos, pero adicionalmente se necesita el control el “Gimbal” o

solamente la activación de captura de imágenes de la cámara ya sea con el “Gimbal” fijo durante toda la misión o en movimiento.

Para fines del proyecto el cumplimiento de rutas y cierta demostración del manejo del “Gimbal” es suficiente ya que el sistema UAV tiene equipado una cámara GoPro que no tiene un disparador adicional y solamente posee el botón de captura.

Mediante Mission Planner se puede realizar todo lo descrito anteriormente, y tiene la ventaja de que automáticamente en cada misión de vuelo genera un archivo en la computadora donde se está utilizando el programa el cual contiene la información relevante del vuelo y la cual se puede utilizar para la misma georreferenciación de imágenes.

Con todo lo dicho previamente el usuario solo se encarga de la programación de la ruta comprendiendo la función de los comandos que se pueden utilizar mediante Mission Planner. Una vez programada la ruta el usuario la carga en el Pixhawk y queda listo para cumplirla en cualquier momento que se active la función de piloto automático.

El proceso de la programación de la ruta se puede automatizar de cierta manera mediante scripts desarrollados en Python, y es aquí donde estos scripts toman cierta importancia ya que se intenta automatizar cualquier tarea que se realice en Mission Planner solamente con la ejecución de un script.

En el desarrollo del proyecto únicamente se utilizan estos scripts para automatizar el proceso de programación de rutas, con Mission Planner se realiza la programación de una ruta en un lugar en específico, en cambio con un script puede realizar la programación de la ruta ya sea en el lugar específico o alrededor del sistema UAV con la interacción del GPS con una lógica matemática básica de la suma y resta de grados de latitud y longitud de la posición de referencia.

Ya que el proceso de programación de rutas se realiza por medio de scripts de Python, se desarrollarán varios scripts para la validación del cumplimiento de misiones de manera autónoma donde se probará el sistema UAV en diferentes escenarios para comprobar su buen funcionamiento.

Capítulo 4: Resultados y análisis

En este capítulo se probaron los diferentes módulos que forman parte del sistema de control de vuelo automático, primero se comprobó el correcto funcionamiento de estos en el sistema antes de volar y luego todo el sistema en conjunto en varias pruebas de vuelo que fueron programadas por medio de scripts de Python.

Antes de avanzar con la validación del sistema, es necesario comprobar la versión del firmware y actualizarla de ser necesario, el proceso de actualización del firmware se encuentra en el Apéndice A.2 en conjuntos con varios procesos de calibración.

4.1 Resultados del sistema de control antes del primer vuelo

Se necesita comprobar la correcta conexión entre la estación y el sistema UAV, y de los diferentes módulos conectados. Al alimentar la plataforma Pixhawk responde con normalidad, reconociendo cada uno de los sensores conectados e indicando por medio del parlante reproduciendo sonidos de su inicialización al igual que la secuencia de colores mostrada por el LED incluido.

El primer paso que se realiza es conectar el radio restante en el puerto USB de la computadora corriendo Mission Planner, se logró realizar la conexión entre los dispositivos, donde la estación muestra el estado del sistema UAV en tiempo real tal como se muestra en la figura 4.1 donde en la esquina superior derecha se comprueba la correcta conexión entre sistemas y el estado del sistema UAV en la parte izquierda.



Figura 4.1 Conexión UAV-Estación.

La primera impresión que se da al iniciar la conexión es la ubicación del sistema UAV en Google Maps por medio de la interfaz de Mission Planner, la cual es bastante precisa ya que es muy cercana al punto dado en la aplicación real de Google Maps y gracias a que se recibe una señal estable del GPS. Los datos exactos dados por Mission Planner se observan en la parte inferior de la figura 4.1, así también la altitud no es 0 m al iniciar pero define una altitud relativa de 0 m al armar los motores.

Con el enlace de comunicación completo, la estación indica la presencia de problemas en el sistema si es que los hay, es posible que el sistema necesite calibración del magnetómetro, acelerómetro, control y motores por lo que es necesario realizar este proceso antes de continuar con la comprobación. El manual de calibración se encuentra en el Apéndice A.2.

Entre los problemas comunes que presenta es el GPS HDOP que se refiere a la precisión del GPS, el sistema no permite el armado de motores si es mayor que 2, por lo que siempre procure tener una buena señal, entre más cercano a la unidad es mejor. Otros problemas que se pueden presentar son en el barómetro, magnetómetro y otros del GPS los que no se han presentado.

Una vez que la plataforma se encuentra calibrada el sistema se encuentra perfectamente orientado por el magnetómetro, indicando esta misma orientación en cuanto a los puntos cardinales en la interfaz de Mission Planner, en la parte izquierda de la figura 4.1.

Cabe indicar que siempre que se alimenta el sistema es preferible realizarlo en una zona estable y nivelada ya que la plataforma por defecto define ese valor como el valor de referencia y por lo que siempre se recomienda darle tiempo por la propia calibración del giroscopio.

En cuanto al comportamiento del “Gimbal” con el control o la misma estación se encuentra respondiendo de la mejor manera, ya que el movimiento del “Gimbal” responde a las indicaciones que se dan, el movimiento realizado es solo en un eje de arriba hacia abajo (tilt). La misma cámara con el kit de FPV envía el video sin interferencia ya que se encuentra a una distancia bastante corta por lo que es importante velar por este factor a distancias más grandes cuando se encuentre en pleno vuelo, al igual que los radios de telemetría que en teoría tienen más alcance.

Después de la calibración de los motores, estos responden a la misma intensidad cuando se produce el armado de motores y cambian su intensidad de diferentes maneras al mover los joysticks del control y se detienen en el procedimiento de desarmado por lo que se vuelve válido el propio comportamiento del control y los motores. El armado de motores es posible cuando se haya presionado el botón de seguridad y además el sistema se encuentre estable (buena señal de GPS y perfectamente calibrado) que se demuestra por medio del LED del Pixhawk que se muestra en color verde.

La primera prueba del equipo respondiendo en tierra se logró sin problema alguno, se comprobó el debido funcionamiento de los radios de telemetría, GPS, IMU, motores, “Gimbal”, estación y la misma plataforma de piloto automático por ende todo el sistema funciona al inicializar y se encuentra listo para volar.

4.2 Resultados del sistema de control en vuelo manual

Estas pruebas se realizaron con el fin de comprender el comportamiento del sistema UAV, y ver como éste se desempeña en sus primeros vuelos. El control tiene por defecto 3 tipos de vuelo que se pueden utilizar, el primero es el “Standard” el cual se controla con pura acción del control y no necesita acción del GPS ya que la ubicación se da por el usuario, este método es bastante complicado y normalmente es para expertos o personas con gran experiencia con el manejo de estos tipos de sistemas.

El otro modo es el “Loiter” el cual se maneja el sistema UAV de manera manual pero al no haber ninguna acción del control el dron permanece en la misma ubicación y en la misma altura, es el modo indicado para realizar las pruebas de acción del control de piloto automático en cuanto al desempeño de los módulos en un vuelo manual. El último modo de operación es el “Auto” que se utiliza cuando se programan las misiones de vuelo autónomo.

El primer paso es el propio armado de motores, luego se eleva el vehículo a cierta altura en este caso lo haremos a 6 m que se pueden visualizar por medio de la interfaz de la estación. La figura 4.2 muestra el sistema a dicha altura y el estado del mismo por medio de Mission Planner.



Figura 4.2 Prueba de vuelo manual.

En la altura fija no se realiza ninguna acción del control remoto para probar la acción de los sensores sin perturbación alguna, el sistema se mantiene estable en cuanto a la inclinación del propio y la altura no varía de gran manera. La ubicación cambia poco a poco en lo más mínimo debido a la acción del viento pero también se debe a la resolución del GPS la cual es bastante alta pero no es perfecta, el cambio no se nota en cuanto a la visualización en Mission Planner al ser tan mínimo. El resto de comandos del control se ejecutan con normalidad por lo que el sistema en si se encuentra perfectamente calibrado y muy estable.

En cuanto a la visión mediante el FPV kit se mostró el video sin interferencia aproximadamente a los 75 m de distancia por lo que es importante en los vuelos manuales y automáticos a cortas distancias.

La validación de la cobertura de los radios de telemetría se dio para un área de aproximadamente de 1000 m² (campo de béisbol) y la conexión se mantuvo estable entre estación y sistema UAV por lo que tiene gran alcance en cuanto a las pruebas que se realizaron para el sistema.

Por último se realiza el aterrizaje manual del sistema UAV el cual hay que disminuir la altitud lentamente hasta alcanzar el suelo para luego realizar el desarmado de motores, este paso es bastante lento en cuanto a que el sistema dura aproximadamente 3 s en reconocer que se está realizando el desarmado de los motores lo que podría ser un problema en el manejo manual del sistema.

El desarrollo de las 2 pruebas realizadas anteriormente comprobó el buen comportamiento de los módulos del sistema de control de vuelo automático en cuanto a la precisión de ubicación y la estabilidad del sistema en el aire y de la conexión a la estación, y la señal recibida por el control manual del sistema, por lo que en teoría deben de funcionar en las diferentes misiones de vuelo autónomas que son las pruebas que se realizaron y se muestran los resultados a continuación.

4.3 Resultados del sistema de control en vuelo automático

Un punto importante de este tipo de pruebas es la programación, ya que se comprobó que el sistema UAV se mantiene en el vuelo debería funcionar de la misma manera con la única diferencia en que la plataforma debe realizar más procesamiento en cuanto el cumplimiento de las rutas de prueba, en esta parte si se realizaron varias pruebas para comprobar el uso de diferentes comandos de piloto automático que resultan indispensables para cumplir con la meta del proyecto y su fines de fotogrametría.

4.3.1 Resultados de ruta simple

Se empieza con una ruta simple ya que las pruebas se realizan para validar el funcionamiento de los scripts de Python y de los comandos básicos en las misiones de piloto automático. La figura 4.3 muestra la lógica de la ruta a programar, donde básicamente es un cuadrado donde los comandos a utilizar son los principales: Takeoff, Waypoint y Land.

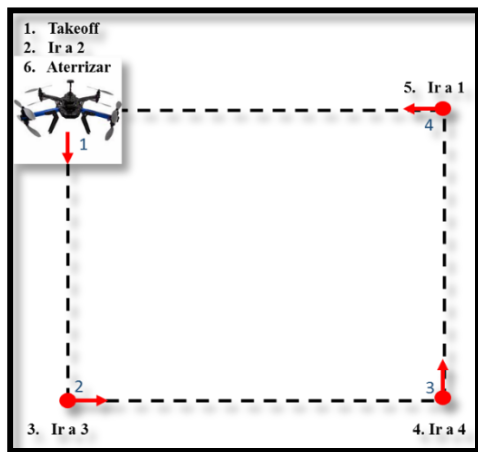


Figura 4.3 Lógica de la ruta simple de prueba a programar.

Para ejecutar la misión en cualquier espacio se necesita la ubicación del sistema UAV por medio del GPS antes de programar la misión por lo que se necesita que el sistema se encuentre alimentado, el GPS tenga buena señal y exista conexión entre UAV-estación.

En el Apéndice A.3 se encuentra el código de programación en Python del cual es necesario importar ciertas bibliotecas como la de Mission Planner y la de MAVLink, con el script listo se ejecuta sin problema alguno, la figura 4.4 muestra la ruta programada en la interfaz de Mission Planner y la ventana que se genera al correr el script indicando que se completó la programación.

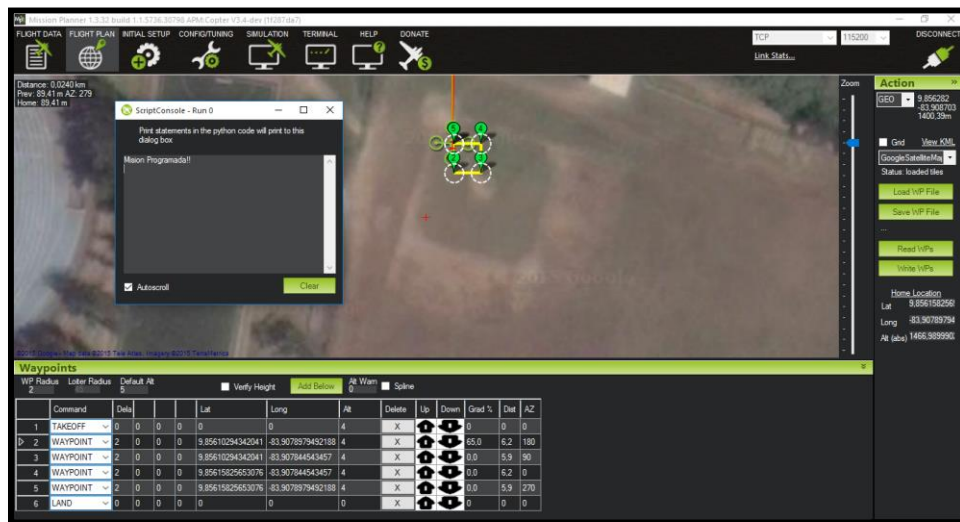


Figura 4.4 Ruta simple programada por script de Python.

Con la ruta cargada en la plataforma Pixhawk se hizo el vuelo de prueba, con las características de que se hizo a baja altura (4 m) y la distancia entre cada punto era aproximadamente de 5 m, además en cada punto se hizo un tiempo de espera de 2 s ya que cada punto tiene un radio de 2 m por defecto es decir que normalmente el sistema UAV al tocar con el radio se dirige al otro punto por lo que con los 2 s es suficiente para que llegue al punto en sí.

Otro aspecto importante es la velocidad que era de 6 m/s el cual es otro parámetro por defecto el cual se puede cambiar antes de iniciar la misión, o en la misma misión se puede cambiar con el comando Do Change Speed en cual no se utilizó ya que no tiene trascendencia alguna.

Entre los resultados obtenidos se puede decir que el sistema UAV no tuvo problemas con el cumplimiento de la ruta, la figura 4.5 muestra la ruta completada por el sistema en el modo de piloto automático y su estado después del aterrizaje.



Figura 4.5 Prueba de vuelo piloto automático.

La parte del descenso al ser automática se realiza a la velocidad que el usuario requiera, y por defecto se encuentra que a alturas menores de 10 m el sistema UAV aterriza a 0.5 m/s, además desarma los motores si el joystick izquierdo se encuentra en el mínimo por lo que mejoró en este aspecto en cuanto a la prueba anterior.

Como se puede apreciar en la figura 4.5 la ruta cumplida no es precisa en cuanto a la programación pero básicamente el sistema cumple la ruta la cuál mejora con una mejor señal de GPS, también factores como mayores distancias y el radio de los puntos mejora su desempeño.

Posteriormente se cumplieron más rutas aumentando ciertos parámetros como la distancia entre puntos, cantidad de puntos, altura y velocidad donde el sistema cumplió las misiones sin problema alguno.

Al cumplirse los vuelos se comprueba el perfecto funcionamiento del sistema de control de vuelo automático en cada uno de sus módulos, ya que se mantuvo estable y bien ubicado durante el desarrollo de las rutas, quedando únicamente pendiente el conocer si puede realizar tareas de fotogrametría.

4.3.2 Resultados de ruta con comando para fotogrametría

Mission Planner tiene varios comandos útiles para realizar tareas de fotografía aérea, los cuales se utilizan para manejar el “Gimbal” y tomar fotografías, en nuestro caso la cámara GoPro no tiene la opción de tomar una fotografía que no sea por medio del botón por lo que nos enfocaremos en el manejo del “Gimbal” solamente.

El comando utilizado se denomina Do Set ROI el cual el frente del sistema UAV y el “Gimbal” apuntan hacia cierto punto con altitud incluida (enfoco del “Gimbal”), la ruta programada en este tipo en teoría es mucho más sencilla donde el sistema UAV se dirige a cierto punto en línea recta apuntando hacia cierto punto y luego se devuelve apuntando al lado opuesto, la figura 4.6 muestra la lógica de la ruta a programar en este caso.

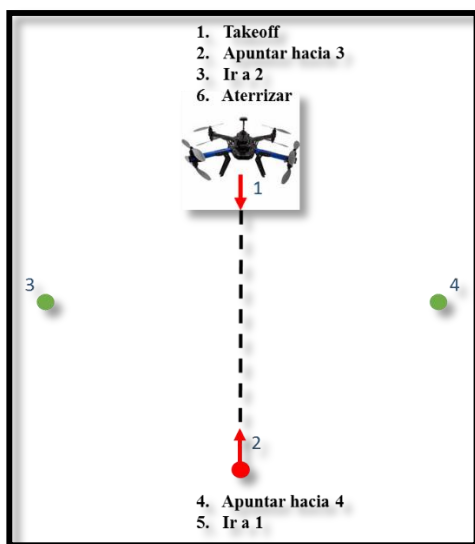


Figura 4.6 Lógica de la ruta de prueba a programar.

Una vez planteada la ruta se programó por medio de Python tomando como base la programación anterior, ya que se utilizan las mismas bibliotecas, y la misma cantidad de comandos pero con diferentes funciones. El código de programación también se encuentra en el Apéndice A.3, donde Mission Planner programó la ruta sin problema alguno reconociendo el nuevo comando a utilizar tal como se muestra en la figura 4.7.



Figura 4.7 Ruta de comando de fotogrametría en Mission Planner.

Se puede observar en la figura 4.7 que a diferencia de la prueba anterior los puntos rojos (2 y 4) son hacia donde apunta el frente del sistema UAV en vez de apuntar al punto donde se dirige el sistema.

La misma ruta fue cargada en la plataforma Pixhawk y se puso a prueba en el mismo campo donde se realizaron las pruebas anteriores, la figura 4.8 muestra la ruta completada exitosamente, en donde no hubo ningún problema en el desarrollo de ella ni en los parámetros dados por medio de la estación.



Figura 4.8 Ruta de prueba completada en Mission Planner.

Entre los factores a observar durante el desarrollo de la prueba fue el comprobar el cambio en cuanto al frente del sistema UAV donde también realizó movimientos del “Gimbal” para apuntar a la altitud requerida, en la misma ruta se realizaron cambios de altitud, distancia y otros factores para validar el perfecto comportamiento del sistema.

4.4 Resultados del sistema de control después del vuelo

Con las rutas de vuelo cumplidas, la plataforma de Mission Planner autogenera unos archivos llamados logs que contienen información referente al comportamiento del sistema durante su funcionamiento, es decir desde que se alimentó el sistema hasta que se cortó la alimentación. Estos archivos se utilizan para analizar la información obtenida por medio de gráficos que realiza el mismo programa, la figura 4.9 muestra un gráfico con información básica de la altitud de un vuelo realizado.

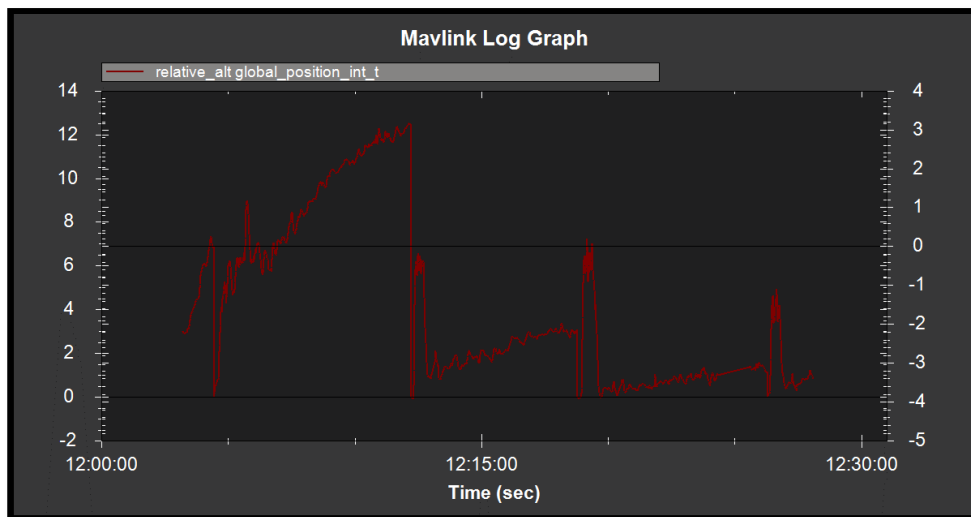


Figura 4.9 Gráfico de altitud relativa elaborado por Mission Planner.

El gráfico anterior muestra el cambio de altitud relativa del sistema en el tiempo, como se mencionó anteriormente el valor de 0 m se obtiene cada vez que se arman los motores o se realizó un aterrizaje. Existen una gran cantidad de parámetros que se pueden graficar y analizar por este tipo de archivos como el comportamiento de la IMU, el GPS, alimentación y otros los cuales dieron resultados como los observados durante el desarrollo de las rutas de prueba.

El fin principal de los logs es el de georreferenciar imágenes en tareas de fotogrametría por lo que en teoría se pueden cumplir este tipo de tareas con esta misma plataforma con el equipo indicado lo que no se puede realizar con una cámara GoPro.

Para finalizar el proyecto queda el cumplimiento de un requerimiento planteado, que el peso de la unidad sea menor a 500 g, aunque la validación del sistema quedó totalmente completa se le puede realizar cambios para disminuir el peso de ser necesario, en la tabla 4.1 se muestra el peso aproximado de cada módulo que constituye el sistema de control.

Tabla 4.1 Pesos de los módulos del sistema de control de vuelo automático.

| Módulo | Peso (g) |
|---------------------|-----------------|
| Pixhawk | 38 |
| GPS | 34 |
| Radio Telemetría | 6 |
| “Gimbal” con Cámara | 289 |
| Extras | 30 |
| Total | 397 |

Con lo mostrado en la tabla 4.1 el sistema en su totalidad tiene un peso aproximado de 397 g, cabe mencionar que la batería se considera en el peso del sistema UAV por lo que no se toma en cuenta en esta unidad, en cuanto a los extras se refiere a todos aquellos cables y módulos livianos adicionales incluidos en el sistema.

Dicho lo anterior se cumplen todos los objetivos planteados en el proyecto, por lo que se comprueba el buen funcionamiento del sistema en su totalidad, por lo que la implementación del sistema de control de vuelo automático planteada cumple la etapa de validación ya que obtuvieron los resultados esperados.

5.1 Conclusiones

En este trabajo se logró desarrollar y verificar exitosamente un sistema de piloto automático para la aplicación de fotogrametría con UAV basado en la plataforma Pixhawk.

La implementación de una solución basada en “open hardware” y “open source” es mucho más económica a cualquier otra solución pensada, también es mucho más dinámica en cuanto a que tiene gran cantidad de información ya sea del hardware y software que facilita la conexión entre bastantes dispositivos y constante actualización del software gracias a la gran cantidad de desarrolladores de código.

La utilización de una plataforma de piloto automático de propósito general como el Pixhawk tiene la ventaja de personalización para lograr realizar una tarea específica, lo que la hace una plataforma ideal para cualquier implementación con características únicas para los sistemas UAV.

Se logró implementar un sistema de control de vuelo automático estable para el sistema UAV X8+ de la empresa 3DR compuesto de dispositivos distribuidos por la misma compañía lo que facilitó la compatibilidad entre módulos y su fácil adaptación entre sí sin necesidad de drivers adicionales.

De la implementación del sistema de control de vuelo automático se determinaron las siguientes características:

- Posee una unidad central con gran cantidad de puertos disponibles y salidas PWM dejando abierta la ampliación del sistema mediante módulos adicionales para mejorar su rendimiento o personalizar su funcionalidad.
- Gran estabilidad debido a la buena calibración y acción del giroscopio y acelerómetro en los 3 ejes incluidos en la plataforma Pixhawk para determinar los ángulos “pitch”, “roll” y “yaw” a gran precisión.

- Su rendimiento depende principalmente en la precisión de seguimiento de rutas que aumenta conforme la ubicación horizontal de la señal de GPS se acerca al valor de la unidad.
- La capacidad de realizar tareas de fotogrametría con ayuda de la plataforma Mission Planner con gran cantidad comandos para estos fines y la herramienta de georreferenciar imágenes.
- Comunicación con la estación terrestre por medio de una enlace de radio frecuencia (433 MHz) de gran alcance dentro de un área de aproximadamente de 1000 m² sin problema alguno.
- Alta configuración en varios aspectos como velocidad del sistema UAV, tipos de vuelo, procedimientos de seguridad para evitar problemas ante varias situaciones, por medio de la estación
- Peso menor al de 400 g lo que lo hace un sistema bastante liviano en cuanto a las limitaciones del sistema UAV X8+, por lo que no limita el rendimiento máximo de tiempo de vuelo.

El firmware ArduCopter es el software ideal para sistemas UAV ya que además de ser libre y tener infinidad de información al respecto en diferentes aplicaciones también muy flexible con el usuario porque se presta a la personalización del mismo para diferentes fines en diferentes sistemas.

La plataforma de software Mission Planner es una plataforma muy poderosa que se complementa de forma perfecta con el sistema de control de vuelo automático para mostrar el estado del UAV en vuelo al usuario, además de características únicas de configuración, planeamiento de misiones, georreferenciación de imágenes y análisis post vuelo para determinar fallos cuando se presenten problemas.

El uso de scripts desarrollados en Python ayuda en la automatización del uso de la plataforma de software en programación de misiones, además de la ventaja de programar dichas misiones de forma relativa y mejorando la exactitud en grados de latitud y longitud y la misma altitud.

5.2 Recomendaciones

Luego del trabajo desarrollado se citan a continuación varias recomendaciones que son importantes considerar para el desarrollo del trabajo a futuro con la plataforma seleccionada.

Por motivos de seguridad ante cualquier proceso de calibración o prueba estacionaria con el sistema UAV en un área cerrada se recomienda realizarla sin las hélices para evitar cualquier accidente.

Se puede mejorar el alcance de los radios de telemetría mediante el uso de una antena adicional o el cambio por transmisores más potentes o reduciendo la misma tasa de datos enviados.

El uso de un segundo módulo de GPS mejora el funcionamiento del sistema de control de vuelo automático en cuanto a redundancia de la señal de GPS al ser un sistema tan dependiente de este mismo módulo, siendo una opción que se puede explorar a futuro.

Se recomienda el uso de una cámara fotográfica con disparador infrarrojo con el debido “Gimbal” para labores de fotogrametría, y mejoras en el desfase de georreferenciación de imágenes.

Para el manejo manual del sistema UAV se recomienda el modo “Loiter” para los inexperimentados y realizarlo con la presencia de una persona familiarizada con el sistema ante cualquier emergencia.

Ante cualquier misión de vuelo en piloto automático mantener el control manual del sistema al alcance para manejar el sistema UAV en caso de que se presente alguna situación inesperada.

Estar al tanto del portal Ardupilot.com el cual se mantiene en constante actualización sobre las nuevas ventajas del nuevo firmware, comandos y aplicaciones realizadas en la plataforma Pixhawk para los sistemas UAV.

Bibliografía

- [1] G.Mehuys, K.Tiessen, & M.Villatoro, “Erosión por Labranza con Arado de Disco en Suelos Volcánicos de Ladera en Costa Rica”. Costa Rica, 2009.
- [2] C.F.Morales, “Manual Evaluaciones en la Parcela de Escorrentía y Erosion”. Lima:1996.
- [3] N.W.Hudson, “Medición sobre el terreno de la erosión del suelo y la escorrentía”, Roma: FAO, 1997. [Online]. Available [http://www.fao.org/docrep/t0848s/t0848s05.htm#capítulo 3 parcelas de campo](http://www.fao.org/docrep/t0848s/t0848s05.htm#capítulo%203%20parcelas%20de%20campo).
- [4] S.Arriola, “Diseño de un Sistema de Adquisición de Datos para un Vehículo Aéreo no Tripulado (UAV)”, Lic dissertation, Instituto Tecnológico de Costa Rica, Cartago, 2015.
- [5] L.Coelho, J.Nunes, “Fotogramtería digital”. Rio de Janeiro: EdUERJ, 2007.
- [6] D.H.López, “Introducción a la fotogrametría digital”, Universidad de Castilla, La Mancha, 2006.
- [7] H.EISENBEIß, “UAV Photogrammetry”, Ph.D dissertation, Zurich, 2009.
- [8] Pix4Dmapper, Agosto 2015 [Online]. Available: <https://pix4d.com/products/>.
- [9] Erdas imagine 2015 product features and comparisons, Agosto 2015 [Online]. Available: <https://p.widencdn.net/0citnh>.
- [10] About Agisoft, Agosto 2015 [Online]. Available: <http://www.agisoft.com/about/>.
- [11] D.Gilman, “Unmanned Aerial Vehicles in Humanitarian Response”, OCHA, 2014.
- [12] S.G.Gupta, M.M.Ghonge, P.M.Jawandhiya, “Review of Unmanned Aircraft System (UAS)”, in International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol 2, pp. 1646-1658, Apr 2013.
- [13] About DJI, Agosto 2015 [Online]. Available: <http://www.dji.com/company>.

- [14] About Parrot, Agosto 2015 [Online]. Available: <http://www.parrot.com/usa/aboutparrot/corporate-overview/>.
- [15] About 3DR, Agosto 2015 [Online]. Available: <http://3drobotics.com/about/>.
- [16] V.M.Pérez, “Sistema de control que permite el vuelo autónomo de drones”, Universidad de Alicante.
- [17] M.Riquelme, “Diseño y construcción de un avión no tripulado basado en sistemas y dispositivos COTS”, Lic dissertation, Universidad Politécnica de Cartagena, Cartagena, 2013.
- [18] J.Rey, “El Sistema de Posicionamiento Global – GPS”, Universidad de Florida, 2006.
- [19] S.Fux, “Development of a planar low cost Inertial Measurement Unit for UAVs and MAVs”, Master dissertation, Instituto Tecnológico de Zurich, 2008.
- [20] G.Ferrer, “Integración Kalman de sensores inerciales INS con GPS en un UAV”, RSLab TSC UPC, 2009.
- [21] G.Crespo, “Sistema de enlace robusto para la teleoperación de un UAV (vehículo aéreo no tripulado) en la plataforma robótica ARGOS”, Lic dissertation, Universidad Autonoma de Madrid, 2014.
- [22] F.Jaramillo, A.Gómez, “SISTEMA DE CONTROL PARA LA ESTABILIDAD Y ORIENTACIÓN DE UN HELICÓPTERO QUADROTOR”, Lic dissertation, Escuela de Ingeniería de Antioquia, 2013.
- [23] DJI A2, Agosto 2015 [Online]. Available: <http://www.dji.com/product/a2>.
- [24] DJI A2 User Manual, Agosto 2015 [Online]. Available: http://download.dji-innovations.com/downloads/a2/en/A2_Quick_Start_Guide_v1.22_en.pdf
- [25] OpenPilot CopterControl Platform, Agosto 2015 [Online]. Available: <https://www.openpilot.org/products/openpilot-coptercontrol-platform/>
- [26] AeroQuad 32 Flight Control Board, Agosto 2015 [Online]. Available: http://www.aeroquadstore.com/AeroQuad_32_Flight_Control_Board_Version_2_p/raq32-001.htm

- [27] DJI PC Ground Station, Agosto 2015 [Online]. Available: <http://www.dji.com/product/pc-ground-station/download>
- [28] OpenPilot GCS, Agosto 2015 [Online]. Available: <https://www.openpilot.org/download/>
- [29] MultiWii Software, Agosto 2015 [Online]. Available: <http://www.multiwii.com/software>
- [30] J.Osio, L.Antonini, “Descripción General de un Microcontrolador (Módulos de Comunicación)”, Universidad Nacional de la Plata, 2011.
- [31] “Introducción al bus CAN”, Departamento de Sistemas Embebidos, Universidad de Murcia.
- [32] C.Ortega, “IMPLEMENTACIÓN DE UN TUNNEL IN THE SKY PARA MICRO UAV”, Lic dissertation, Universidad Politecnica de Catalunya, 2013.
- [33] S. Balasubramanian, “MavLink Tutorial for Absolute Dummies”, Holanda.
- [34] X8+, Septiembre 2015 [Online]. Available: <https://3drobotics.com/x8/>
- [35] Store X8+, Septiembre 2015 [Online]. Available: <https://store.3drobotics.com/products/x8-plus>
- [36] X8+ Operation Manual, Septiembre 2015 [Online]. Available: <https://3drobotics.com/wp-content/uploads/2015/04/X8-Operation-Manual-vC.pdf>
- [37] LiveView Kit for GoPro User Guide, Septiembre 2015 [Online]. Available: https://3drobotics.com/wp-content/uploads/2015/04/LiveViewKit_vE.pdf
- [38] Gimbal User Guide, Septiembre 2015 [Online]. Available: <https://3drobotics.com/wp-content/uploads/2014/09/Gimbal-User-Guide-vA.pdf>
- [39] 3DR Radio v2, Septiembre 2015 [Online]. Available: <http://copter.ardupilot.com/wiki/common-3dr-radio-version-2/>
- [40] 3DR Radio v2 Quick Start Guide, Septiembre 2015 [Online]. Available: <https://3drobotics.com/wp-content/uploads/2013/10/3DR-Radio-V2-doc1.pdf>

- [41] 3DR uBlox GPS with Compass, Septiembre 2015 [Online]. Available: <https://3drobotics.com/wp-content/uploads/2013/08/3DR-uBlox-GPS-web-version.pdf>
- [42] 3DR uBlox GPS + Compass Module, Septiembre 2015 [Online]. Available: <http://copter.ardupilot.com/wiki/common-installing-3dr-ublox-gps-compass-module/>
- [43] Pixhawk Overview, Septiembre 2015 [Online]. Available: <http://copter.ardupilot.com/wiki/common-pixhawk-overview/>
- [44] MPU 6000 Datasheet, Septiembre 2015 [Online]. Available: http://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf
- [45] MS5611 Datasheet, Septiembre 2015 [Online]. Available: <http://www.meas-spec.com/downloads/MS5611-01BA03.pdf>
- [46] PX4 Firmware, Septiembre 2015 [Online]. Available: <https://pixhawk.org/firmware/downloads>
- [47] APM Firmware, Septiembre 2015 [Online]. Available: <http://firmware.diydrones.com/>
- [48] QGROUNDCONTROL, Septiembre 2015 [Online]. Available: <http://qgroundcontrol.org/start>
- [49] Mission Planner, Septiembre 2015 [Online]. Available: <http://planner.ardupilot.com/>

Apéndices

A.1 Selección de la plataforma de piloto automático y estación

El proceso de selección es el más importante en el desarrollo de cualquier proyecto ya que depende mucho de los requerimientos y restricciones. En esta sección se plantea una tabla comparativa de las características más importantes de las plataformas de piloto automático distribuidas por la empresa 3D Robotics para luego dar una explicación del porqué se seleccionó dicha plataforma y después se realiza el mismo procedimiento para las plataformas de software disponibles.

Selección de la Plataforma de Piloto Automático


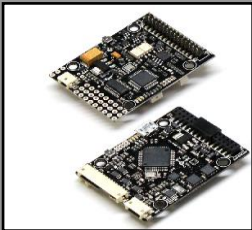

Es la unidad principal del sistema que se configura para manejar el sistema UAV de forma autónoma y también interactúa con los sensores del sistema para procesar los datos medidos para mantener el sistema estable y en buen funcionamiento.

Las plataformas a comparar son las siguientes:

- APM 2.6
- PX4
- Pixhawk

Entre las características importantes a considerar es su tiempo de respuesta, capacidad, puertos, limitaciones y peso, las cuales están claramente ligadas al precio del producto por lo que también es un aspecto importante. La tabla A.1 muestra las características de dichas plataformas y cabe indicar que las 3 plataformas pueden utilizar las mismas plataformas de software para programarlas por lo que no es un factor a considerar.

Tabla A.1 Tabla comparativa de las plataformas de piloto automático.

| Plataforma | APM 2.6 | PX4 | Pixhawk |
|--------------------|---|--|---|
| |  |  |  |
| Procesador | ATmega 2560 8 bits | Cortex M4F 32 bits | Cortex M4F 32 bits |
| Frecuencia | 16 MHz | 168 MHz | 168 MHz |
| RAM | 8 KB | 192 KB | 256 KB |
| Flash | 256 KB | 1024 KB | 2 MB |
| Sensores | MPU 6000 Barómetro | MPU 6000 Barómetro Magnetómetro Giroscopio | MPU 6000 Barómetro Magnetómetro Giroscopio Acelerómetro |
| Interfaces | 2 UART, I ² C, microUSB | 4 UART, 2 I ² C, SPI, CAN, microUSB, microSD | 5 UART, I ² C, SPI, 2 CAN, microUSB, SD |
| Salidas PWM | 8 | 8 | 14 |
| Peso | 28 g | 8.10 g | 38 g |
| Precio | \$ 159.99 | \$ 149.99 | \$ 199.99 |

A pesar que las 3 plataformas tienen diferente rendimiento y diferentes sensores, la experiencia con el usuario es similar al igual que ciertas características.

La plataforma elegida es el Pixhawk ya que tiene un procesador más potente que el APM 2.6 y está basado en el PX4 pero con más características adicionales, el Pixhawk es la última y más avanzada de las 3 plataformas mostradas con más capacidad en su RAM y Flash y con las características que contiene giroscopio y acelerómetro de respaldo por lo que la hace más confiable.

Además el Pixhawk tiene más puertos en caso de ser necesarios, y el número de salidas PWM es un factor importante ya que las otras plataformas estarían limitadas al manejo de los motores si tener salidas disponibles para el manejo de otros servomotores como el “Gimbal” y disparadores de ciertas cámaras.

A pesar de que es más pesado y más costoso la diferencia es mínima en comparación a las otras características donde si es mucho mejor, y es por eso que el Pixhawk se vuelve la plataforma idónea para el desarrollo del proyecto.

Selección de la Plataforma de Software (Estación)



Para la selección de la plataforma de software hay que considerar varios aspectos y entre ellos está el firmware que utilizan, es decir el Pixhawk es compatible con 2 firmwares que son el APM y PX4 donde ambos corren en el mismo middleware de PX4 y se comunican vía MAVLink.

Los 2 firmwares son muy similares en cuanto su funcionamiento, entre las diferencias se puede mencionar que el PX4 es muy general en cuanto al tipo de sistema ya que utiliza una base de código común con un enfoque flexible y estructurado que permite ejecutar el control ya sea de un avión, multicoptero o terrestre con las misma misión de vuelo y lo realiza por medio de la estación QGroundControl.

El APM o ArduPilot se basa en el uso de varios firmwares dependiendo del tipo de vehiculo autónomo, el cual se carga por medio de la estación que es este caso puede ser Mission Planner o APM Planner dependiendo del sistema operativo.

Los firmwares PX4 y APM son bastantes confiables y contienen una gran cantidad de información al respecto por lo que la elección del firmware se dará por la comparación de las plataformas de software en sí, ya que es más importante analizar sus características y su interacción con el usuario. La tabla A.2 muestra dicha comparación.

Tabla A.2 Tabla comparativa de las plataformas de software.

| Estación |  |  |  |
|---------------------------------|---|--|---|
| Firmware | PX4 | ArduPilot | ArduPilot |
| Sistema Operativo | Windows, Mac, Linux | Windows, Mac | Windows, Mac, Linux |
| Waypoints | Sí | Sí | Sí |
| Scripts | Sí (Python) | No | No |
| Georreferenciar Imágenes | Sí | No | No |

Las 3 GCS (Estaciones de Control Terrestre) tienen licencia GPL de código abierto y libre lo que le permite a los usuarios la libertad de usar, estudiar, compartir (copiar) y modificar el software sin problema alguno por lo que tiene gran cantidad de información al respecto.

La plataforma elegida es Mission Planner ya que es la estación con más características y más utilizada por entusiastas y desarrolladores, por lo que en términos de operatividad, complejidad, consumo de recursos la hacen la mejor opción. Además tiene la ventaja que se pueden programar misiones por medio de scripts hechos en Python y también la opción de georreferenciar imágenes que es una tarea importante en la fotogrametría.

Pesa a que APM Planner es de los mismos desarrolladores de Mission Planner es básicamente una extensión de esta última por lo que tiene menos características y lo hace menos completa. Y la interfaz es bastante intuitiva por lo que también ayuda a los nuevos usuarios.

A.2 Manual de calibración

Actualización de Firmware

Es un proceso simple y rápido que actualizará la plataforma Pixhawk con el firmware más reciente que proporciona Mission Planner. El proceso solo se puede realizar vía USB por lo que se conecta el Pixhawk por el puerto micro-USB tal como se observa en la figura A.1.



Figura A.1 Conexión en el Pixhawk vía USB.

En Mission Planner en la esquina superior derecha seleccione el puerto en auto o el número de puerto asignado por Windows y fije la tasa de baudios en 115200, NO presione **Connect** la figura A.2 ilustra este procedimiento.

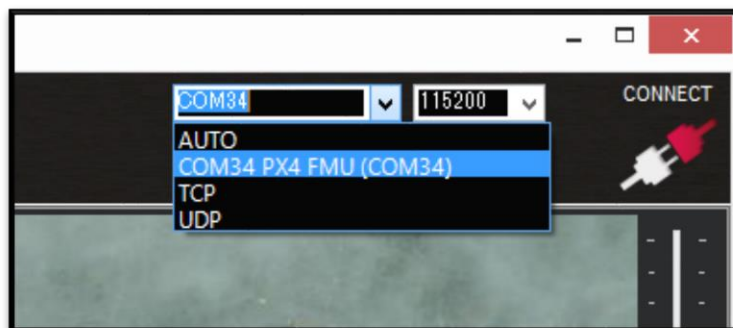


Figura A.2 Selección del puerto vía USB y la velocidad en Mission Planner.

Luego en Mission Planner seleccione la opción de *Initial Setup* y luego *Install Firmware*, seleccione la opción llamada *ArduCopter V3.2.1 Octa-quad*. Una vez seleccionada la opción Mission Planner detecta cual plataforma de piloto se está utilizando por lo que le solicita al usuario que desconecte la placa, presione OK y luego conecte de nuevo la plataforma, este procedimiento se observa en la figura A.3.

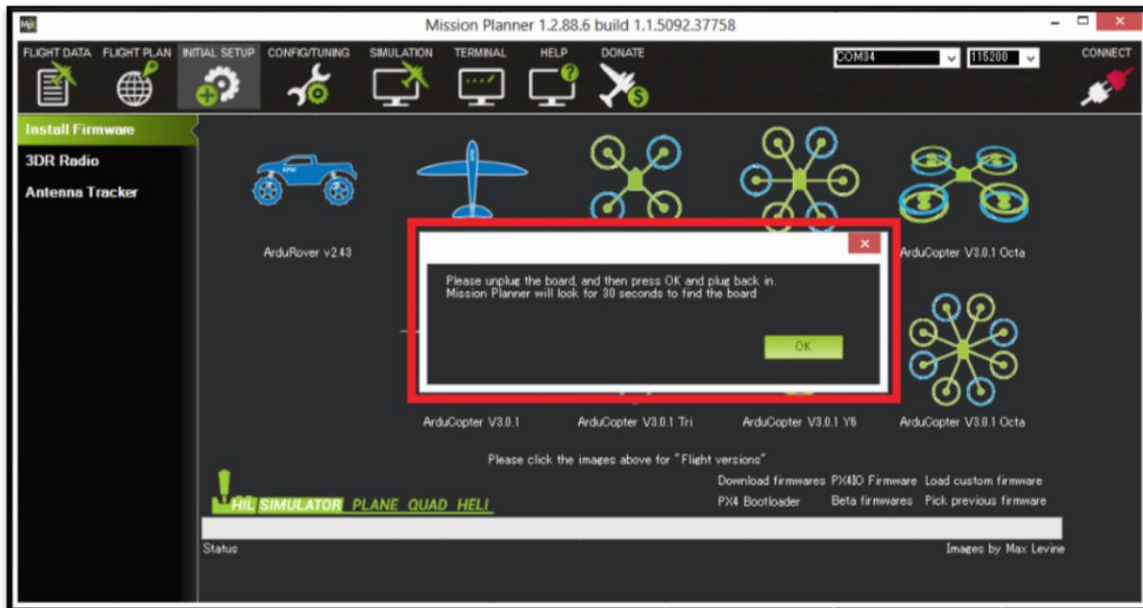


Figura A.3 Procedimiento de actualización del firmware.

Cumplido lo anterior debajo en la barra en status aparecerán palabras como, “erase...”, “program...”, “verify...” and “Upload Done”. Y la actualización habrá terminado.

Normalmente cuando se conecta la plataforma de piloto automático a Mission Planner le notifica si existe una versión de firmware reciente y si quiere actualizarla.

Calibración

Es necesario calibrar el equipo antes del primer vuelo para evitar cualquier tipo de problemas durante el vuelo, aun así el UAV X8+ tiene predeterminado procedimientos de seguridad como es el caso de no armar los motores si algún componente no se encuentra calibrado o en buen estado (Pre-Arm Safety Check).

La calibración se lleva a cabo por medio de la estación, en este caso se utiliza Mission Planner, y la plataforma de piloto automático utilizada es el Pixhawk para el modelo 3DR-X8+. Antes de iniciar con la guía de calibración es importante mencionar que la información fue tomada del portal ardupilot.com donde lo realizan de forma general, y aquí se realiza de forma específica con los componentes que se mencionaron anteriormente.

Durante el proceso de calibración el sistema UAV necesita estar conectado a la estación, en este caso la conexión se puede realizar por medio del cable USB como se mostró en la figura A.1 o los radios de telemetría.

Si la conexión se realiza por medio de los radios de telemetría uno se conecta en el puerto Telem1 (default) y el otro en la PC por medio del puerto micro-USB del radio. Las conexiones se muestran en la figura A.4



Figura A.4 Conexión en el Pixhawk por Radios de Telemetría.

Luego la instalación de los drivers se realiza de manera automática, y se le asigna un puerto como COM “número” (Verificar el número del puerto en el Device Manager).

Cuando se abre Mission Planner en la esquina superior derecha de la pantalla aparece una pequeña pestaña sobre el puerto de conexión y otra sobre la tasa de baudios tal como se aprecia en la figura A.5.

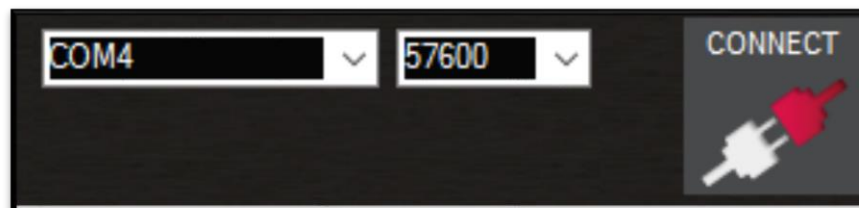


Figura A.5 Selección de puerto y velocidad en Mission Planner.

En las pestañas se elige el puerto asignado automáticamente, y en la velocidad se elige 115200 (USB) o 57600(Radios) y dar click en el botón que dice **Connect**. Si la conexión se logra debería mostrarse algo similar a la figura A.6.

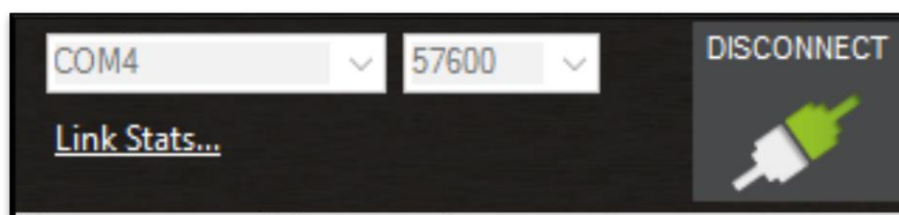


Figura A.6 Pixhawk conectado a Mission Planner.

Una vez conectado el sistema UAV a la estación es importante seleccionar el tipo de frame, y esto se lleva a cabo en la estación en la ventana llamada **Initial Setup**, donde luego se selecciona la opción llamada **Mandatory Hardware** y luego **Frame Type** y se escoge la opción “X” que de todas formas viene predeterminada. Este proceso se visualiza en la figura A.7.

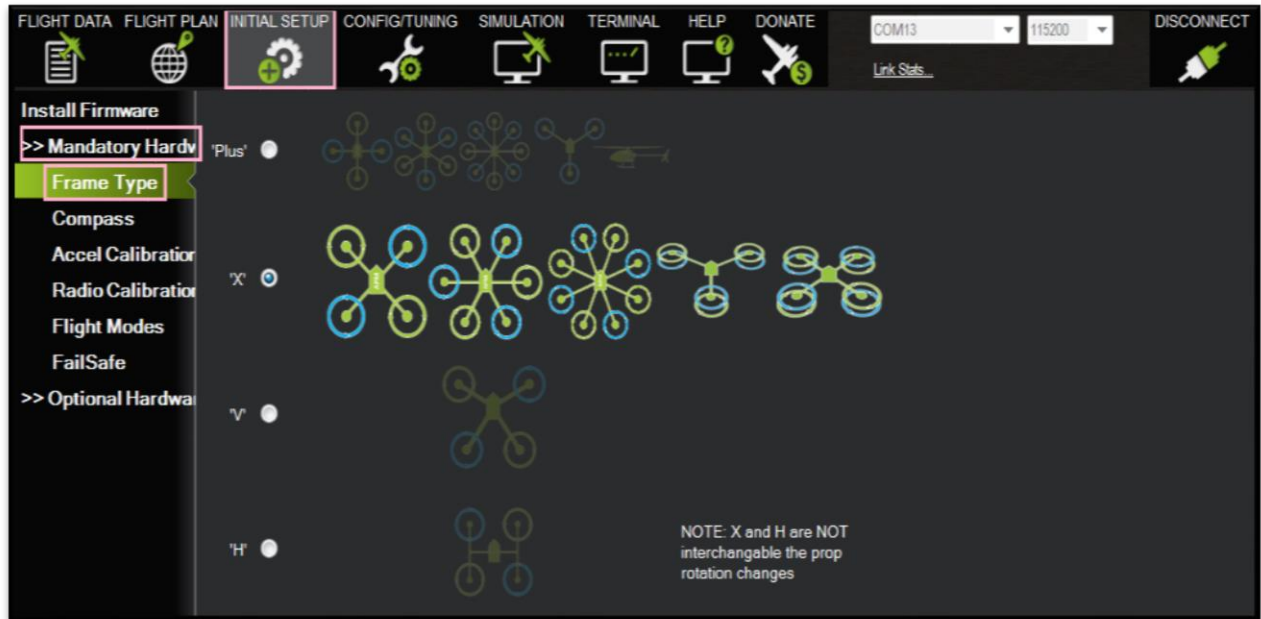


Figura A.7 Selección del frame en Mission Planner.

Ya que se seleccionó el frame es posible proceder a la calibración del equipo siempre y cuando se mantenga la conexión entre el UAV y la estación. Entre el equipo a calibrar se encuentran el magnetómetro, acelerómetro, control y ESA. A continuación se explica paso a paso el proceso de calibración de cada componente:

Calibración Acelerómetro

En Mission Planner se ingresa a *Initial Setup, Mandatory Hardware* y luego *Accel Calibration* tal como se observa en la figura A.8.

Normalmente para los procesos de calibración del acelerómetro y el magnetómetro se recomienda el uso de la comunicación por medio de los radios de telemetría para evitar complicaciones y limitaciones con el cable USB, además se recomienda realizarlo en una superficie totalmente plana ya que va a ser la referencia de calibración.

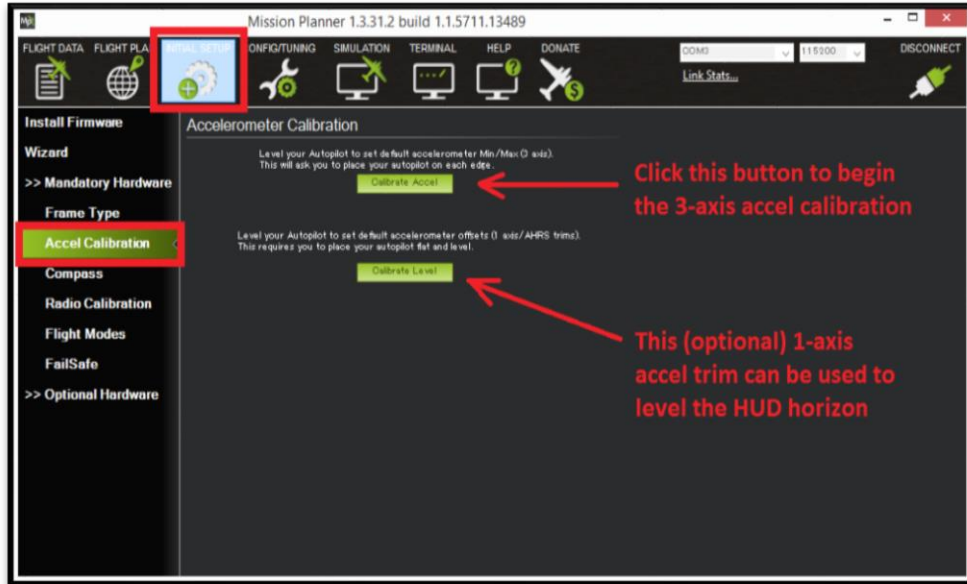


Figura A.8 Selección de calibración del acelerómetro.

Cuando se encuentre listo seleccione *Calibrate Accel*, una vez seleccionada Mission Planner le pedirá que coloque el sistema UAV en ciertas posiciones y presione una tecla cuando se encuentre en las posiciones requeridas. La figura A.9 muestra la secuencia a seguir con el modelo X8+. Una vez completado el proceso de calibración Misión Planner mostrará en la pantalla “**Calibration Successful**” y el acelerómetro estará listo para el vuelo.

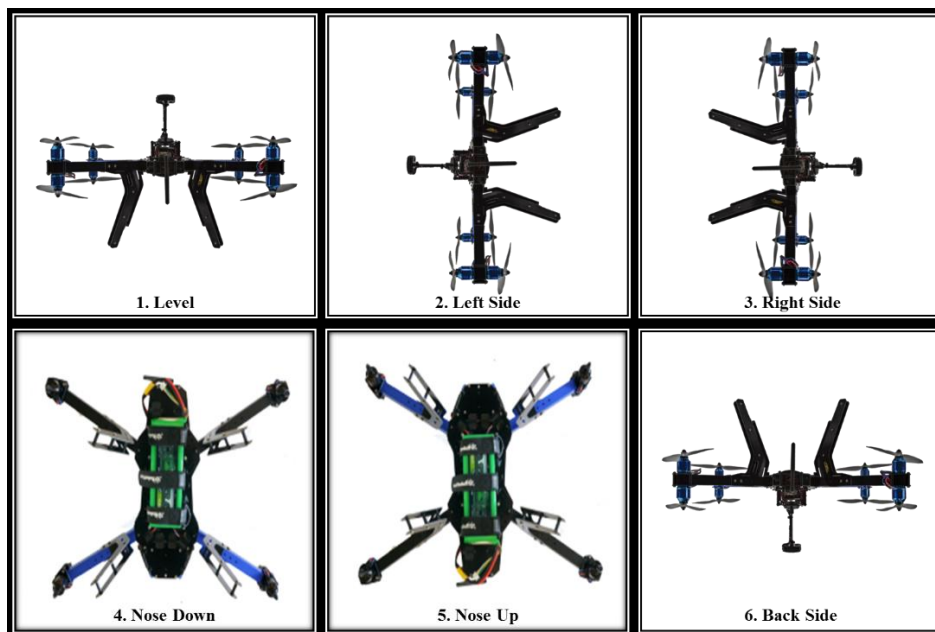


Figura A.9 Posiciones calibración acelerómetro.

Calibración Magnetómetro

Este proceso de calibración tiene gran grado de complejidad, para una mejor explicación se recomienda ver este [video](#) [1] que utilizan otra clase de dron, incluye también el los otros procesos de calibración y la carga de un nuevo firmware en el Pixhawk.

Para comenzar se ingresa a *Initial Setup, Mandatory Hardware* y luego *Compass* dentro de Mission Planner tal como visualiza en la figura A.10. Asegúrese que las opciones de *Enable* y *AutoDec* se encuentren seleccionadas, y seleccione *Pixhawk/PX4*.

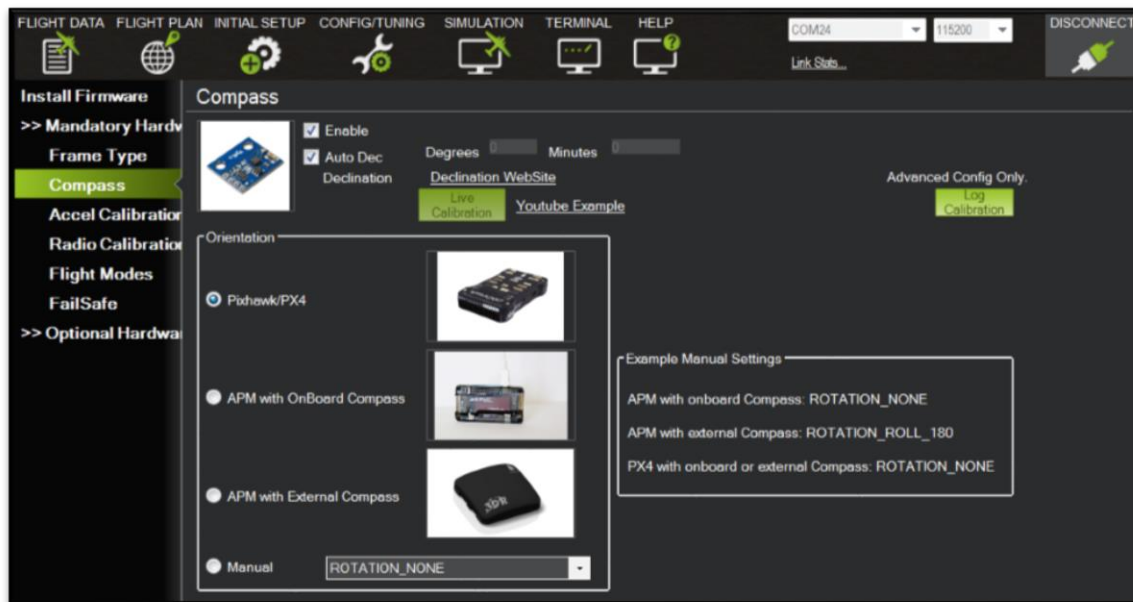


Figura A.10 Selección de calibración del magnetómetro.

Cuando selecciona *Pixhawk/PX4* aparece una ventana preguntando si la versión del firmware es mejor que APM:copter 3.01 por lo que le damos click en *Yes* ya que se posee la versión 3.2.1. Ya seleccionada la opción *Pixhawk/PX4* se procede a dar click en el botón llamado *Live Calibration*, donde aparecerá una ventana de que tiene 60 segundos para rotar el Pixhawk y presione *OK*. Luego aparece una ventana del proceso de calibración como la que se muestra en la figura A.11, donde el objetivo trata de pasar el punto rojo por todos los puntos blanco para la calibración.

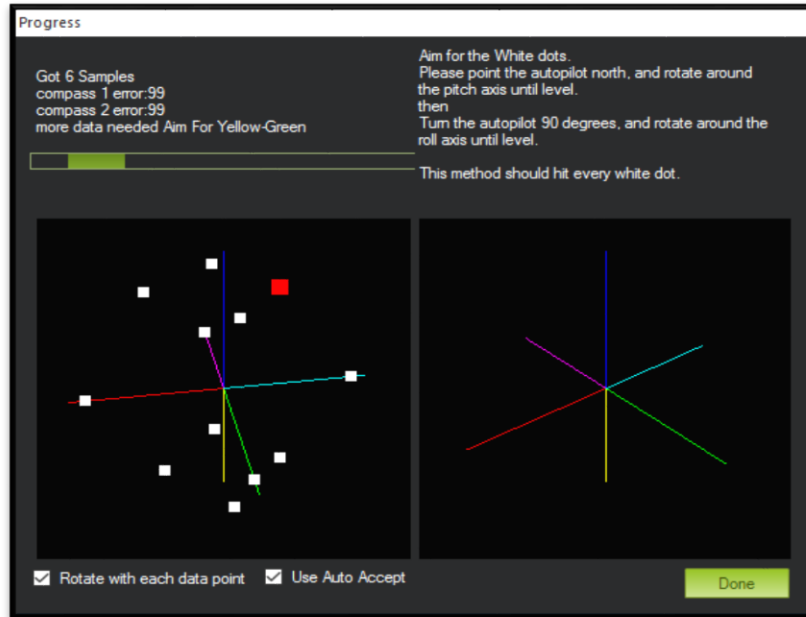


Figura A.11 Ventana de calibración del magnetómetro.

Durante el proceso de calibración siga los siguientes pasos:

Paso 1: Apunte el frente del X8+ hacia el norte y rote el dron lentamente sobre el eje “pitch” dando 2 vueltas.

Paso 2: Gire su cuerpo 90° a la izquierda y rote el UAV sobre el eje “roll” despacio dando 2 vueltas.

Ambos pasos se pueden representar en la figura A.12 que se muestra a continuación para una mejor comprensión (recuerde apuntar con las extremidades azules hacia el norte en el paso 1):

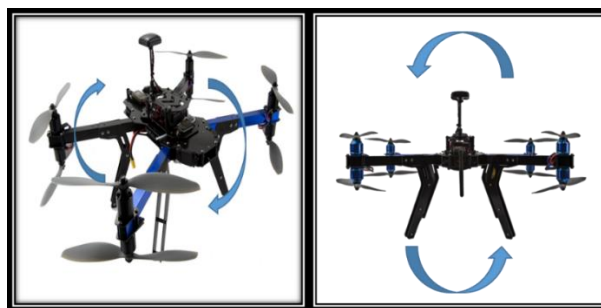


Figura A.12 Pasos calibración magnetómetro, paso 1 (izquierda) y paso 2 (derecha).

Finalizada la calibración aparecerá una ventana mostrando los nuevos valores de los offsets los cuales deberían estar entre 150 y -150. Es posible que necesite repetir el proceso varias veces.

Calibración Control

La calibración del control y ESC es relativamente mucho más sencilla, en el caso del control verifique que se encuentre en modo standard (STD) y enciéndalo, recuerde que el joystick izquierdo controla throttle y yaw, y el joystick derecho controla pitch y roll tal como se observa en la figura A.13.

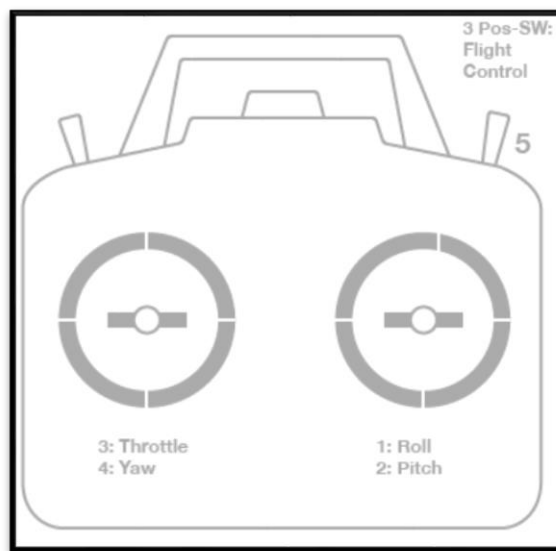


Figura A.13 Representación gráfica del control.

En Mission Planner hay que ingresar a **Initial Setup, Mandatory Hardware** y luego **Radio Calibration**, luego hacer click en el botón que dice **Calibrate Radio** y aparecerá una ventana para asegurar que el control se encuentra encendido, la batería **NO** está conectada al UAV y las hélices **NO** se encuentran montadas, si es así presione **OK** para iniciar la calibración.

Mueva el joystick hacia los límites izquierdo y derecho para calibrar el roll (canal 1) y luego hacia arriba y abajo para calibrar el pitch (canal 2). Realice el mismo paso para el joystick izquierdo para el throttle (canal 3) y el yaw (canal 4). Luego pase el switch de modos por los 3 modos (canal 5) los cambios en la calibración deberían observarse similares como en la figura A.14.



Figura A.14 Proceso de calibración del control.

En nuestro caso en manejo del “Gimbal” rotamos la perilla llamada tilt al máximo y al mínimo y los cambios se visualizarán en el canal 6 y los canales 7 y 8 son opcionales (consultar en el sitio ardupilot.com). Seleccionar *Click when Done* cuando termine el proceso y le aparecerá una ventana con los resultados de la calibración.

Calibración ESC

Los controladores de velocidad electrónicos (ESC) son los responsables de girar los motores a la velocidad solicitada por el usuario. La mayoría de ESCs necesitan ser calibrados para saber cuál es el mínimo y máximo de valores que el control puede entregar. Hay varios métodos de calibración pero en nuestro caso utilizaremos el método llamado *All at Once* que generalmente funciona para cualquier UAV de 3DR.

En este proceso de calibración asegúrese de que la calibración del control ya fue realizada, además que las hélices NO estén montadas, el Pixhawk NO se encuentra conectado vía USB, la batería TAMPOCO está conectada y siga los siguientes pasos:

1. Encienda el control y suba el joystick izquierdo al máximo.
2. Conecte la batería al sistema UAV y el Pixhawk hará una secuencia de colores de rojo, azul y amarillo en el LED indicando que la próxima vez se realizará la calibración de los ESCs.
3. Desconecte la batería y conéctela de nuevo.
4. Mantenga presionado el botón de seguridad hasta que se mantenga en rojo sólido.
5. Espere a una serie de sonidos normalmente son 4 de las celdas de la batería y 2 más de que se encuentra en modo de calibración.
6. Baje el joystick izquierdo del control al mínimo.
7. Espere a un largo tono indicando que la calibración fue completada.
8. Luego del tono los motores se encuentran ARMADOS, si sube el joystick izquierdo un poco estos debería empezar a girar. Pruebe si los motores giran al subir y baje y repita. Si todos los motores no giran repita el proceso de calibración.
9. Baje el joystick izquierdo al mínimo y desconecte la batería para terminar el modo de calibración de los ESCs.

En teoría el UAV se encuentra listo para volar, por lo que el usuario puede proceder a seguir los pasos de armado de los motores. Si lo motores aún no arman conecte el Pixhawk a Mission Planner por USB o radios de telemetría (recuerde asignar el puerto, la tasa de baudios y presionar Connect). Encienda el control e intente armar los motores de nuevo, la primera causa por la cual los motores no se arman se mostrará con letras rojas en la estación y consulte [el portal](#) [2] la manera de solucionar el problema.

Nota hipervínculos:

[1] <https://www.youtube.com/watch?v=hiLm8-K6Z0I>

[2] http://copter.ardupilot.com/wiki/prearm_safety_check/

A.3 Código de los scripts de Python

Ruta simple de prueba (Cuadro)

```
import sys
import math
import clr
import time
import System
from System import Byte

clr.AddReference("MissionPlanner")
import MissionPlanner
clr.AddReference("MissionPlanner.Utilities") # includes the Utilities class
from MissionPlanner.Utilities import Locationwp
clr.AddReference("MAVLink") # includes the Utilities class
import MAVLink

## *****Definiciones iniciales*****##

idwp = MAVLink.MAV_CMD.WAYPOINT          ##Identificacion comando waypoint
id = int(idwp)                            ##Entero del id de waypoint
idto = MAVLink.MAV_CMD.TAKEOFF           ##Identificacion comando takeoff
idl = MAVLink.MAV_CMD.LAND               ##Identificacion comando land
lathome = MAV.getWP(0).lat;
lnghome = MAV.getWP(0).lng;

## *****Definiciones ubicaciones*****##

##Define home en ubicación actual

lathome = cs.lat;                         ##Obtiene latitud del UAV
lnghome = cs.lng;                         ##Obtiene longitud del UAV
```

```

##Punto1
lat1 = lathome - 0.00011          # Valor latitud
lng1 = lnghome                   # Valor longitud
alt = 6                          # Valor altitud
Delay = 2                        ##Delay de 2s en cada punto

##Punto2
lat2 = lat1                      # Valor latitud
lng2 = lnghome + 0.00011        # Valor longitud

##Punto3
lat3 = lat1 + 0.00011           # Valor latitud
lng3 = lng2                     # Valor longitud

wp0 = Locationwp().Set(0,0,alt,int(idto)) ##Primer comando elevar a Altitud
wp1 = Locationwp().Set(lat1,lng1,alt, id) ##Segundo comando ir al punto 1
Locationwp.p1.SetValue(wp1, Delay) ##Agregando delay
wp2 = Locationwp().Set(lat2,lng2,alt, id) ##Tercer comando ir al punto 2
Locationwp.p1.SetValue(wp2, Delay)
wp3 = Locationwp().Set(lat3,lng3,alt, id) ##Cuarto comando ir al punto 3
Locationwp.p1.SetValue(wp3, Delay)

wprtl = Locationwp().Set(lathome,lnghome,alt, id) ##Volver al punto home
Locationwp.p1.SetValue(wprtl, Delay)

wpl = Locationwp()
Locationwp.id.SetValue(wpl, int(idl)) ##Aterrizar

MAV.setWPTtotal(7) #Total de waypoints a cargar en la plataforma (Home cuenta como 1)
MAV.setWP(MAV.getWP(0),0,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #Home
MAV.setWP(wp0,1,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #Takeoff
MAV.setWP(wp1,2,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #wp1
MAV.setWP(wp2,3,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #wp2
MAV.setWP(wp3,4,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #wp3
MAV.setWP(wprtl,5,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #wprtl
MAV.setWP(wpl,6,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT); #wpl
MAV.setWPCurrent(1); #restart mission to waypoint 0
MAV.setWPACK(); ##Cargar paquete en plataforma
print 'Mision Programada!!'

```

Ruta de prueba (Comando de fotogrametría)

```
import sys
import math
import clr
import time
import System
from System import Byte

clr.AddReference("MissionPlanner")
import MissionPlanner
clr.AddReference("MissionPlanner.Utilities") # includes the Utilities class
from MissionPlanner.Utilities import Locationwp
clr.AddReference("MAVLink") # includes the Utilities class
import MAVLink

## *****Definiciones iniciales*****##

idwp = MAVLink.MAV_CMD.WAYPOINT      ##Identificacion comando waypoint
id = int(idwp)                        ##Entero del id de waypoint
idto = MAVLink.MAV_CMD.TAKEOFF       ##Identificacion comando takeoff
idl = MAVLink.MAV_CMD.LAND           ##Identificacion comando land
idc = MAVLink.MAV_CMD.DO_SET_ROI     ##Identificacion comando do set roi

lathome = MAV.getWP(0).lat;
lnghome = MAV.getWP(0).lng;

## *****Definiciones ubicaciones*****##
##Define home en ubicacion actual

lathome = cs.lat;                    ##Obtiene latitud del UAV
lnghome = cs.lng;                    ##Obtiene longitud del UAV
```

```

##Punto1
lat1 = lathome - 0.0003          # Valor latitud
lng1 = lnghome                  # Valor longitud
alt = 10                        # Valor altitud

##Apuntar1
lat2 = lat1 + 0.00015          # Valor latitud
lng2 = lnghome + 0.00009      # Valor longitud
altp = 2                       # Valor altitud

##Apuntar 2
lat3 = lat1 + 0.00015          # Valor latitud
lng3 = lnghome - 0.00009      # Valor longitud
Delay = 5

wp0 = Locationwp().Set(0,0,alt,int(idto))  ##Primer comando elevar a alt
wp1 = Locationwp().Set(lat1,lng1,alt, id)   ##Tercer comando ir al punto 1
Locationwp.p1.SetValue(wp1, Delay)

wp2 = Locationwp().Set(lat2,lng2,altp, int(idc)) ##Segundo comando apuntar frente del UAV
wp3 = Locationwp().Set(lat3,lng3,altp, int(idc)) ##Cuarto comando apuntar frente del UAV al otro lado
wprtl = Locationwp().Set(lathome,lnghome,alt, id) ##Volver al punto home
Locationwp.p1.SetValue(wprtl, Delay)

wpl = Locationwp()
Locationwp.id.SetValue(wpl, int(idl))      ##Aterrizar
MAV.setWPTotal(7) #set wp total
MAV.setWP(MAV.getWP(0),0,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wp0,1,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wp2,2,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wp1,3,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wp3,4,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wprtl,5,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWP(wpl,6,MAVLink.MAV_FRAME.GLOBAL_RELATIVE_ALT);
MAV.setWPCurrent(1);
MAV.setWPACK();

print 'Mision Programada!!'

```