

DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE RUTAS PARA EL VUELO DE UN QUADROTOR

Iván Darío Peñaloza Borrás
Iván Mauricio Jaimes Niño

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
Bucaramanga – 2013**

DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE RUTAS PARA EL VUELO DE UN QUADROTOR

Iván Darío Peñaloza Borrás
Iván Mauricio Jaimes Niño

Trabajo para optar por el título de Ingeniero Electrónico

Director:

MSc. Alfredo Rafael Acevedo Picón

Codirector:

MSc. Jorge Hernando Ramón Suárez



ESCUELA DE INGENIERÍAS
ELÉCTRICA, ELECTRÓNICA
Y DE TELECOMUNICACIONES



UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
Bucaramanga – 2013

*Me gustaría agradecer por su inmensa paciencia, a mi Abuelo y gran amigo **JUAN ANTONIO**, por estar siempre a mi lado, por ser el ejemplo, por los consejos que han sido de gran ayuda para mi vida y crecimiento personal. Es por ello Abuelo que hoy te dedico este trabajo de grado, que es el resultado de lo que me has enseñado.*

Gracias Abuelo JUAN por confiar en mí y darme la oportunidad de culminar esta etapa de mi vida.

Gracias al MSc. Jorge Hernando Ramón y grupo ERA, que me ayudaron con mi formación profesional y por dejarme ser parte de este tan selecto grupo de trabajo y personas.

Iván Darío Peñaloza Borrás

Al supremo creador por permitir mi existencia y libertad de mis acciones.

A mis padres, familia y personas cercanas por la formación continua basada en los valores humanos, la disciplina y la dedicación, pilares para tener resultados como el que se expone en este trabajo y para el desarrollo de nuestra sociedad.

A la Universidad Industrial de Santander, UIS, la formación académica y significativo apoyo del cuerpo docente, que permiten fortalecer el pensamiento científico para desarrollar las actividades.

Al grupo ERA y cada uno de sus integrantes, fundamentales en el desarrollo de la ingeniería aplicada, la solución de problemas, el trabajo en equipo y el ideal de brindar un desarrollo tecnológico en la comunidad.

A los señores directores: MSc. Jorge Hernando Ramón y MSc. Alfredo Rafael Acevedo por permitir y guiar el desarrollo de este y entre otros proyectos.

Iván Mauricio Jaimes Niño

Índice general

Lista de figuras	12
Lista de tablas	13
Glosario	14
Introducción	17
Motivación y objetivos	17
Estado del arte	18
Contribución de este trabajo	21
1. Descripción del sistema	22
1.1. Movimientos del vehículo	22
1.1.1. Movimiento de guiñada	23
1.1.2. Movimientos de alabeo y cabeceo	23
1.2. Radioenlace y modos de vuelo	24
1.3. Control de actitud y Adquisición de datos	25
1.4. Generalización del sistema	25
2. Descripción de quadrotor	27
2.1. Estructura	27
2.2. Actuadores	30
2.3. Control electrónico de velocidad	32
2.4. Batería	33
3. ArdupilotMega 2.5	36
3.1. APM 2.5	36
3.2. Sensórica del ArdupilotMega 2.5	37
3.3. Mission Planner	38
3.3.1. Firmware dentro del APM 2.5	38
3.3.2. Configuración iniciales	39
3.3.3. Calibración del Acelerómetro	39

3.3.4.	Calibración del Magnetómetro	39
3.3.5.	Comunicación con GCS	40
3.4.	Mavlink Micro Air Vehicle Communication Protocol	40
3.4.1.	Protocolo MAVLink	40
4.	Diseño del algoritmo para generar la ruta	43
4.1.	Especificaciones del problema.	43
4.2.	Análisis del problema	45
4.3.	Algoritmo determinista	45
4.3.1.	Medición de distancias. Selección del método.	46
4.3.2.	Cálculo de la orientación	48
4.4.	Estrategia para la generación de movimientos	49
4.4.1.	Comportamiento del ángulo ψ	49
4.4.2.	Sistema de control PID. Sintonización de los parámetros de control para el ángulo ψ	50
4.4.3.	Sistema de control proporcional. Sintonización empírica.	54
4.5.	Diagrama de flujo	56
5.	Sistema implementado	58
5.1.	Diseño de la PCB	59
5.1.1.	Elección de la unidad de procesamiento.	59
5.1.2.	Adquisición de los datos	63
5.1.3.	Alimentación	64
6.	Resultados experimentales	66
6.1.	Ángulo yaw. Orientación	66
6.2.	Traslación	67
7.	Conclusiones y Trabajos futuros	71
7.1.	Conclusiones	71
7.2.	Observaciones	72
7.3.	Trabajos Futuros	73
	Bibliografía	74
	Apéndices	76
A.	Códigos de programación	77
B.	Esquemático y Board	83

Índice de figuras

1.	Parrot Ar.Drone. Tomada de http://ardrone.parrot.com/ . . .	19
2.	General Atomics MQ-1 Predator. Fuente: http://www.militaryfactory.com	20
1.1.	Movimiento de guiñada. Tomada de [1]	23
1.2.	Movimiento de alabeo y cabeceo. Tomada de [3]	24
1.3.	APM:Copter V2.5. Fuente: http://diydrones.com/	25
1.4.	Diagrama de flujo del sistema. Fuente: Autores	26
2.1.	Estructura básica. Fuente: http://fpvlongrange.com	27
2.2.	Estructura comercial y modificada. Fuente: http://www.hobbyking.com y autores.	28
2.3.	Modelo 3D en SolidWorks. Fuente: Autores.	29
2.4.	Pieza central superior. Fuente: Autores.	29
2.5.	Pieza central intermedia. Fuente: Autores.	29
2.6.	Soporte para el motor. Fuente: Autores.	30
2.7.	Pieza lateral. Fuente: Autores.	30
2.8.	Motor seleccionado. Fuente: http://www.hobbyking.com	31
2.9.	Hélices. Fuente: Autores.	32
2.10.	ESC seleccionados. . Fuente: Autores.	33
2.11.	Batería. Fuente: Autores.	34
2.12.	Estructura y ensamblaje de prueba. Fuente: Autores.	35
2.13.	Estructura y ensamblaje final. Fuente: Autores.	35
3.1.	Hardware ArdupilotMega 2.5. Fuente www.copter.ardupilot.com	37
3.2.	ArdupilotMega Sensorica y Hardware interno. Fuente: www.copter.ardupilot.com	37
3.3.	Mission planner Firmware. Fuente: www.copter.ardupilot.com	38
3.4.	Posiciones de Calibración del Acelerómetro. Tomada de www.copter.ardupilot.com	39
3.5.	Posiciones de Calibración del Magnetómetro	40

3.6.	Anatomía de los paquetes de Mavlink. Fuente: https://pixhawk.ethz.ch/mavlink/	41
4.1.	Posición inicial del vehículo. Fuente: http://jasonshort.com/	44
4.2.	Secuencia de orientación y traslado. Fuente: Autores.	44
4.3.	Triángulo esférico. Fuente: Autores.	47
4.4.	Cálculo del ángulo yaw deseado. Fuente: Autores.	48
4.5.	Respuesta a la entrada escalón en lazo abierto. Fuente: autores.	50
4.6.	Segundo método de Ziegler-Nichols. Fuente: [11].	50
4.7.	Oscilación sostenida. Fuente: Autores.	51
4.8.	Respuesta al escalón con un control PI. Fuente: Autores.	52
4.9.	Respuesta al escalón con un control PID. Fuente: Autores.	52
4.10.	Acciones en la variable de control por el sistema PID. Fuente: Autores.	53
4.11.	Tendencia de la acción de control para el sistema PID. Fuente: Autores.	54
4.12.	Factor proporcional al error	55
4.13.	Diagrama de flujo del algoritmo generador de trayectorias. Fuente: Autores.	57
5.1.	Esquema del sistema de procesamiento. Fuente: Autores.	58
5.2.	Réplica de las señales del receptor.	63
5.3.	Densidad de las señales del protocolo Mavlink.	63
5.4.	Vista superior del circuito impreso.	64
5.5.	Vista posterior del circuito impreso.	64
5.6.	Tarjeta. Vista Eagle 3D. Fuente: Autores.	65
6.1.	Orientación del vehículo	67
6.2.	Comportamiento de la distancia durante el recorrido	68
6.3.	Comportamiento mejorado de la distancia durante el recorrido	68
6.4.	Prueba de vuelo realizada	69
6.5.	Recorrido de una prueba. Fuente: Google Earth.	70
B.1.	Esquemático de la tarjeta. Fuente: Autores.	84
B.2.	Superior de la tarjeta. Fuente: Autores.	85
B.3.	Posterior de la tarjeta. Fuente: Autores.	85
B.4.	Tarjeta final. Fuente: Autores.	86

Índice de cuadros

2.1. Muestra de motores disponibles en el mercado. Fuente: http://www.hobbyking.com	31
2.2. Características de la batería	34
3.1. Anatomía de los paquetes MavLink. Fuente: https://pixhawk.ethz.ch/mavlink/	41
4.1. Métodos matemáticos para el cálculo de la distancia. Referencia http://search.cpan.org/	46
5.1. Fabricantes y familias de microcontroladores	60
5.2. Características de los microcontroladores Freescale. Fuente: http://www.freescale.com/	61
5.3. Características del microcontrolador MCF51QE128. Fuente [12].	62

Glosario

<i>UAV:</i>	— <i>Unmanned Aerial Vehicle.</i>
<i>VTOL:</i>	— <i>Vertical Take-Off and Landing.</i>
<i>PWM:</i>	— <i>Pulse Width Modulation.</i>
<i>MAV:</i>	— <i>Micro Aerial Vehicle.</i>
ψ :	— <i>Ángulo yaw.</i>
θ :	— <i>Ángulo pitch.</i>
<i>PCB:</i>	— <i>Printed Circuit Board.</i>
<i>GPS:</i>	— <i>Global Position System.</i>
<i>EA:</i>	— <i>Evolutionary Algorithm.</i>
<i>UCAV:</i>	— <i>Unmanned Combat Air Vehicle.</i>
<i>ESC:</i>	— <i>Electronic Speed Control.</i>
<i>APM:</i>	— <i>ArduPilot Mega.</i>
<i>GCS:</i>	— <i>Ground Control Station.</i>
<i>UART:</i>	— <i>Universal Asynchronous Receiver Transmitter.</i>
<i>UDP:</i>	— <i>User Datagram Protocol.</i>
<i>IMU:</i>	— <i>Inertial Measurement Unit.</i>
<i>COP:</i>	— <i>Computer Operating Properly.</i>

Resumen

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE RUTAS PARA EL VUELO DE UN QUADROTOR¹.

AUTORES: IVÁN DARÍO PEÑALOZA BORRÁS, IVÁN MAURICIO JAIMES NIÑO².

PALABRAS CLAVE: UAV (Unmanned Aerial Vehicle), Quadrotor, Algoritmo de traslación, Trayectorias aéreas, GPS.

CONTENIDO:

El presente trabajo consiste en diseñar e implementar un algoritmo para generar trayectorias en un vehículo aéreo de cuatro rotores, quadrotor, en base a información de posicionamiento global y la orientación de la estructura física. El propósito es tener una trayectoria aproximadamente en línea recta desde una coordenada geográfica inicial a una final, destino o setpoint.

Inicialmente se estudia la rutina de movimientos adecuada para llevar a cabo las misiones aéreas o trayectoria, de esta forma, se decide que el desarrollo de la ruta la constituye dos tipos de movimientos: orientación y traslación.

Luego se tiene un análisis del comportamiento rotacional del vehículo para determinar la orientación (Sección 4.4), esto es con el fin de determinar el rumbo de la aeronave y dirigirla al punto geográfico deseado de forma controlada. Adicionalmente, se tiene el cálculo de la distancia (Sección 4.3.1) para determinar la velocidad de avance en el movimiento de traslación y definir la llegada del vehículo a la coordenada final. El hardware utilizado se presenta en el capítulo 2 referente a las características de la aeronave, en el capítulo 5 se describe el circuito electrónico encargado de la adquisición de las señales e información, además se encarga de la generación de señales para las acciones de movimiento y monitoreo.

Finalmente, se determinan los errores del sistema y la funcionalidad del algoritmo (Capítulo 6 y 7), se expone el comportamiento rotacional del vehículo y la precisión al llegar a la coordenada deseada.

¹Proyecto de grado

²Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: MSc. Alfredo Acevedo. Codirector: MSc. Jorge H. Ramón.

Abstract

TITLE: DESIGN AND IMPLEMENTATION OF A ROUTE ALGORITHM FOR THE FLIGHT OF A QUADROTOR ³.

AUTHORS: IVÁN DARÍO PEÑALOZA BORRÁS, IVÁN MAURICIO JAIMES NIÑO⁴.

KEY WORDS: UAV (Unmanned Aerial Vehicle), Quadrotor, Algorithm of traslation, Aerials trajectory, GPS.

CONTENT:

This document describes the design and implementation of an algorithm that generates trajectories of an unmanned air vehicle propelled by four rotors: a quadrotor. This process uses the global position provided by the GPS and the physical structure orientation. The purpose is to get a path, as linear as possible from an initial geographical coordinate to a final coordinate, destination or set point.

A study was carried out on the adequate motion routine to execute the aerial missions and trajectories. This exploration led to the decision to decompose the movement into two motions: orientation and translation.

Secondly, an analysis of the rotational behavior is made to get the orientation (Section 4.4), and determine in this way the course of the aircraft and its further direction to the desired geographical point, in a controlled way. Additionally, the distance I_s is calculated, (Section 4.3.1) para determinar la velocidad de avance en el movimiento de traslación y definir la llegada del vehículo a la coordenada final. along with the velocity of the translational motion, to define the vehicle arrival to the final coordinate. The hardware is presented in chapter2 which describes the aircraft characteristics. Chapter 5 details the signal acquisition circuit and the handling of signal generation to produce motion and monitoring actions.

Finally, the systems errors and the algorithm functionality are determined (Chapter 6 y 7), along with a description of the aircraft rotational behavior and the precision of the vehicle when arriving to the desired coordinate.

³Final Graduate Project.

⁴Physical- Mechanical Engineering Faculty. Electrical, Electronic and Telecommunications Engineering School. Director: MSc. Alfredo Acevedo. Codirector: MSc. Jorge H. Ramón.

Introducción

”Una vez hayas probado el vuelo siempre caminarás por la Tierra con la vista mirando al Cielo, porque ya has estado allí y allí siempre desearás volver”

Leonardo da Vinci (1452-1519)

Motivación y objetivos

Desde el origen de la robótica, se ha buscado obtener robots con capacidades que le permitan al ser humano disminuir el efecto presentado en situaciones adversas, situaciones fuera de la capacidad del cuerpo humano, o en su defecto situaciones que ponen en peligro su integridad o la de un grupo de personas.

La búsqueda de nuevas y desarrolladas formas de vida a llevado a la humanidad a explorar, desde sus inicios, ecosistemas diferentes a los cotidianos como la inmersión en la atmósfera y los océanos. Una de las expresiones más fascinantes de la ingeniería se presenta en los objetos voladores, es la manifestación de estudios en campos de la ciencia como los sistemas mecánicos, electrónicos, sistemas computacionales, diseño, entre otros, siendo cada uno fundamental para un óptimo funcionamiento y la evolución de los sistemas aéreos.

El trabajo presentado a continuación se expone como iniciativa nacida en 2012 con el propósito de plantear una propuesta de investigación y desarrollo, ejecutada como forma de apoyo a una Colombia en vía de desarrollo, teniendo como punto de partida y en el marco de los trabajos previos en el área se tiene la tesis[1] planteada en 2010, buscando continuidad y manteniendo la iniciativa de desarrollo en la línea de investigación de *UAV* en el Grupo Electrónica y Robótica Aplicada⁵(E.R.A.).

⁵Sitio web: <http://grupoera.com.co/>

La propuesta consiste en adquirir la coordenada geográfica por medio de una tarjeta de circuito impreso (PCB), formular una estrategia algorítmica (Capítulo 4) para orientar la estructura hacia un punto deseado del espacio (latitud y longitud) y la traslación hacia el frente para llegar a la coordenada mencionada; de esta forma, se consolida la generación de la trayectoria o ruta de un determinado vehículo aéreo ejecutando un algoritmo en base a las lecturas del GPS y sensores de movimiento.

Inicialmente, se necesita satisfacer una serie de requerimientos para desarrollar lo propuesto. Es necesario adecuar la estructura, el sistema de potencia y los actuadores del vehículo (Capítulo 2), esto se realiza con el propósito de tener aceptables capacidades de carga (aprox. 5 kg) para cubrir con necesidades de este índole, como llevar a bordo cámaras especiales (aproximadamente 1,5 [kg]) para desarrollar *SIG (Sistemas de Información Geográfica)* u organizar paquetes en bodegas, por ejemplo. Como guía se tiene a disposición el estudio realizado en 2010 [1] en donde se hace una selección de los dispositivos que constituyen las etapas mencionadas, además se requiere realizar una calibración del módulo de estabilización (Capítulo 3) para mantener un adecuado vuelo.

Estado del arte

Los *vehículos aéreos no tripulados, UAV*, también son conocidos como *drones*, son tema principal de gran número de investigaciones debido principalmente a la buena relación costos-capacidades, pues resulta más económico invertir en estos vehículos que en las aeronaves tradicionales, además, si se presentan fallas se garantiza una mayor integridad de la vida humana de los operarios y/o pilotos.

Los estudios que se realizan giran en torno a mejoras a nivel estructural, por ejemplo, en la búsqueda de nuevos materiales de construcción que resulten más livianos y resistentes para soportar grandes cargas; en optimizar las técnicas de control y modelado utilizadas para llevar a cabo misiones en condiciones extremas; y en desarrollar *pilotos automáticos* que cumplan con la ejecución de específicos objetivos.

Los avances en este campo se desarrolla paralelamente con la evolución de la electrónica y el aeromodelismo, se considera que en la actualidad se

tiene sensores más fiables, sistemas de control más precisos, materiales y diseños mejorados con respecto a la aerodinámica, entre otros; llegando a tener productos como el *Parrot Ar.Drone* (Figura 1) herramienta que ayuda al desarrollo y entretenimiento de la comunidad interesada, "the flying saucer" ⁶ que ofrecen servicios de seguridad y espionaje en determinadas zonas urbanas por medio de accesorios como cámaras de vídeo, también se pueden encontrar aeronaves de tipo quadrotor que ofrecen especiales servicios, por ejemplo, en el ámbito medio ambiental detectando zonas de riesgo por desastres naturales (incendios, deslizamientos, contaminación, búsqueda y rescate, etc), en el campo industrial colaborando con los procesos y la organización de materiales ⁷, entre otros.



Figura 1: Parrot Ar.Drone. Tomada de <http://ardrone.parrot.com/>

En las investigaciones de algoritmos que rigen el funcionamiento del sistema, se encuentran los *EA, Algoritmos Evolutivos* [9], basados en los procesos de la evolución biológica, con el objetivo de optimizar los procesos de sensado y mejorar las técnicas de reducción de secciones de cruce por radares, por ejemplo; en su mayoría, las investigaciones se desarrollan con fines de crear eficientes *UCAVs*, vehículos de combate, sin embargo el uso militar no limita las aplicaciones de los vehículos aéreos.

La implementación de algoritmos, sistemas autónomos de aprendizaje o aplicación de *Inteligencia artificial* en las aeronaves, contribuyen con la precisión en los vuelos y la creación de pilotos automáticos de más alta calidad. Un sistema robusto puede estar constituido por *leyes de control* [10], inicialmente se hace necesario un modelo de la planta de alta precisión con el cual se puedan relacionar las variables de salida y las variables de control; además, debido a la necesidad de tener múltiples formas de vuelo corresponde tener

⁶tomado de: Drone makes first UK 'arrest' as police catch car thief hiding under bushes, leer más: <http://www.dailymail.co.uk/news/article-1250177/Police-make-arrest-using-unmanned-drone.html>

⁷Watch Flying Robots Build a 6-Meter Tower, IEEE Spectrum: Inside Technology: www.spectrum.ieee.org

acciones de control y algoritmos de navegación.

Los *drones* han desarrollado misiones en procesos militares. El *General Atomics MQ-1 Predator* (Figura 2) es uno de los más desarrollados, de acuerdo a las especificaciones de la Fuerza Aérea de los Estados Unidos, es considerado como un vehículo de altitud media y largo alcance; cuenta con piloto automático para mantener la altura y ejecutar misiones de reconocimiento.



Figura 2: General Atomics MQ-1 Predator. Fuente: <http://www.militaryfactory.com>

En la generación de movimientos, que constituyen un piloto automático, es fundamental tener un sistema de posicionamiento con alta confiabilidad y adecuada frecuencia en la entrega de datos.

El Departamento de defensa de Estados Unidos⁸ (DoD) desarrolla, para propósitos militares, el primer sistema de posicionamiento en 1978 con el objetivo de monitorear además de la posición, la velocidad y el tiempo del movimiento de objetos, como por ejemplo misiles balísticos; en un principio se llamaba TRANSIT y pronto NAVSTAR (*Navigation System with Timing and Ranging*) que se considera como la antigua versión del actual sistema.

Para tener acceso a las señales satelitales en todo el globo terráqueo se necesita de una cantidad mínima de satélites alrededor de nuestro planeta, entre 1978 y 1985 se lanzaron 11 satélites, pero no ha sido suficiente, hasta 1992 llegando a los 24 satélites; a partir del año 1993 esta tecnología empezó a tener el mayor crecimiento en aplicaciones civiles superando las limitaciones y restricciones de años anteriores.

⁸Actual encargado del sistema por medio del Consolidated Space Operations Center, CSOC

Contribución de este trabajo

El presente trabajo tiene como finalidad principal exponer el diseño e implementación de un algoritmo para generar trayectorias.

Es un desarrollo que plantea algunas de las bases para tener un vehículo aéreo totalmente autónomo, las contribuciones se dan en la caracterización de algunos de sus ejes, en este caso, del ψ u orientación del UAV.

Lograr la autonomía en un vehículo de este tipo implica hacer uso dispositivos como el GPS para determinar la coordenada espacial en la que se encuentre el UAV, una vez se tenga en funcionamiento de manera óptima el algoritmo para generar rutas, será posible hacer adecuaciones como por ejemplo ensamblar una cámara ya sea para fotografía o video aéreo, lo cual se convierte en una herramienta importante para los estudios en geomática, pues permitirán tener acceso a lugares en donde debido a las condiciones naturales no ha sido posible llegar.

Otro aspecto importante es el tiempo de vuelo, determinado principalmente por la carga de las baterías, debe encontrarse una relación adecuada entre peso y capacidad y su vez respecto al precio, debido a que mientras mayor sea la calidad de las baterías, mayor será su costo, así mismo con la capacidad y vida útil.

Capítulo 1

Descripción del sistema

En el desarrollo de aeronaves se tienen avances significativos, como son los pilotos automáticos dispuestos a maniobrar el vehículo con varias formas de vuelo, de esta manera, se tienen comportamientos más precisos, con menor probabilidad de errores en comparación con las acciones humanas y con la capacidad de hacer misiones de alto riesgo sin pérdidas significativas como se tiene tradicionalmente.

El presente trabajo se enfoca en dar solución a la mencionada necesidad por medio del estudio de los comportamientos y la generación de movimientos en un vehículo, quadrotor. Consiste en formular un algoritmo que le permita al *UAV* desarrollar una trayectoria y llegar a un punto espacial deseado; por lo tanto, es importante tener la orientación y traslación de la aeronave periódicamente regulada y supervisada a través de sensores.

En mecanismos autónomos, como el expuesto en el presente documento, es fundamental conocer y hacer un análisis en varios aspectos que afectan la funcionalidad del vehículo y constituyen un sistema estructurado. A continuación se presentan los más relevantes.

1.1. Movimientos del vehículo

La plataforma en donde se implementa el algoritmo es considerado como un vehículo con capacidad de despegue y aterrizaje vertical (*Vertical Take-off and Landing, VTOL*)[2]. Los movimientos y el control en los *6 grados de libertad* se realiza mediante la manipulación de las velocidades en los rotores, es decir, con variaciones del empuje que produce cada hélice.

1.1.1. Movimiento de guiñada

Es una rotación en torno al eje Z (ver Figura 1.1). Manipulando este movimiento se puede orientar el vehículo hacia un punto deseado del espacio. Esto es posible si se aumenta la velocidad de dos rotores y se disminuye la de los otros dos; las mencionadas parejas de rotores son izquierda-derecha y frente-atrás. La cantidad de incremento y decremento en cada pareja de rotores es igual, de esta forma la fuerza de empuje resultante se mantiene pero cambia de sentido.

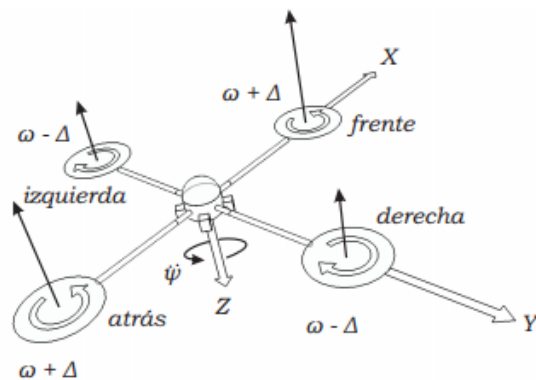


Figura 1.1: Movimiento de guiñada. Tomada de [1]

1.1.2. Movimientos de alabeo y cabeceo

El alabeo y cabeceo son movimientos rotacionales que generan una posición angular alrededor del eje X y Y, respectivamente (ver Figura 1.2).

Para modificar el ángulo ϕ (alabeo) se establece un diferencial de velocidades en los rotores izquierda-derecha (2 y 4), proporcionalmente los otros dos motores aumentarán la fuerza de empuje para mantener constante la altura; de esta manera se obtiene una aceleración lateral.

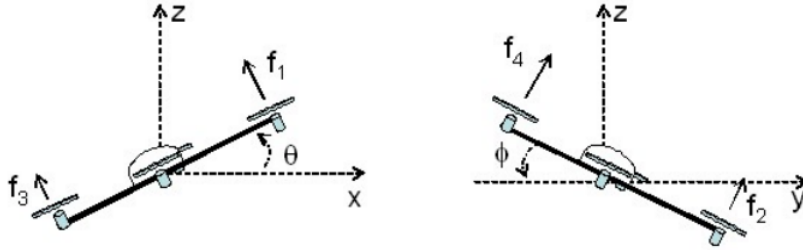


Figura 1.2: Movimiento de alabeo y cabeceo. Tomada de [3]

Para obtener la aceleración longitudinal se modifica el ángulo θ (cabeceo), es similar al proceso anteriormente descrito, en este caso se aplica un diferencial de velocidad a los rotores de frente-atrás (1-3), esta señal diferencial se genera con el módulo de estabilización que se describe en la Sección 1.3. Entonces, estos dos movimientos y los efectos que ocasionan en la dinámica del UAV son conceptos fundamentales para ubicar la estructura en un determinado punto espacial.

1.2. Radioenlace y modos de vuelo

El control remoto o radiocontrol es un dispositivo de comunicación inalámbrica que permite al piloto maniobrar el vehículo; envía la información de actitud o trayectoria a un receptor; este último recibe los datos y se encarga de transmitir una señal PWM hacia el *Control Electrónico de Velocidad (ESC, siglās en inglés)* para generar movimiento en cada rotor. Cada canal del receptor altera un tipo de movimiento (alabeo, cabeceo, traslación, etc), como se menciona en la sección anterior, se necesita manipular por cada tipo de movimiento un determinado par de rotores, esto no sucede por ejemplo en los aeroplanos (disponen de servomotores y un rotor, generalmente).

Se presentan tres modos de vuelo: manual (se controla mediante las ordenes del piloto desde tierra), *loiter mode*¹ (hace referencia a un modo automático que se encarga de mantener la posición actual: latitud, longitud, orientación y altura) y *modo piloto automático* (cuando se activa este modo, la rutina o algoritmo que se propone empieza a ejecutarse y desarrollar una trayectoria aproximadamente lineal).

¹Es un modo de vuelo que ofrece APM-Copter 3, mantiene la posición y altura del UAV.

1.3. Control de actitud y Adquisición de datos

El término *actitud* hace referencia a la postura de un elemento. En la aeronautica civil se conoce como el comportamiento y estabilización de la aeronave, (*attitude*, en inglés). En la sección anterior(1.1) se exponen los movimientos básicos que presenta un vehículo aéreo, en particular el quadrotor, estos determinan la inclinación que presenta la estructura con respecto a sistemas de referencia, estas referencias se conocen como *actitud* del helicóptero[4].

Para estabilizar el vuelo, con respecto a la actitud, se tiene el *APM:Copter* (Ver Figura 1.3). Este módulo está encargado de la suspensión del vehículo, de tal forma que se pueda implementar el *algoritmo de trayectorias* sin entrar en inestabilidad en el comportamiento de los ángulos θ , ϕ y guiñada (ψ). Además, permite extraer información de los *sensores inerciales*, del *magnetómetro* (orientación) y del *GPS*, por medio del protocolo de comunicación *MavLink* (ver Capítulo 3).



Figura 1.3: APM:Copter V2.5. Fuente: <http://diydrones.com/>

1.4. Generalización del sistema

En las secciones 1.1 y 1.3 se describen las características principales para desarrollar el algoritmo que genera las trayectorias. El sistema requiere de componentes para establecer comunicación entre el vehículo y un operario (Figura 1.4), de modo tal, que se pueda maniobrar el helicóptero y activar el modo *piloto automático*; se necesita flujo de datos entre el UAV y un orde-

nador, esta información contiene la configuración de la coordenada deseada (*setpoint*) y la telemetría permanente.

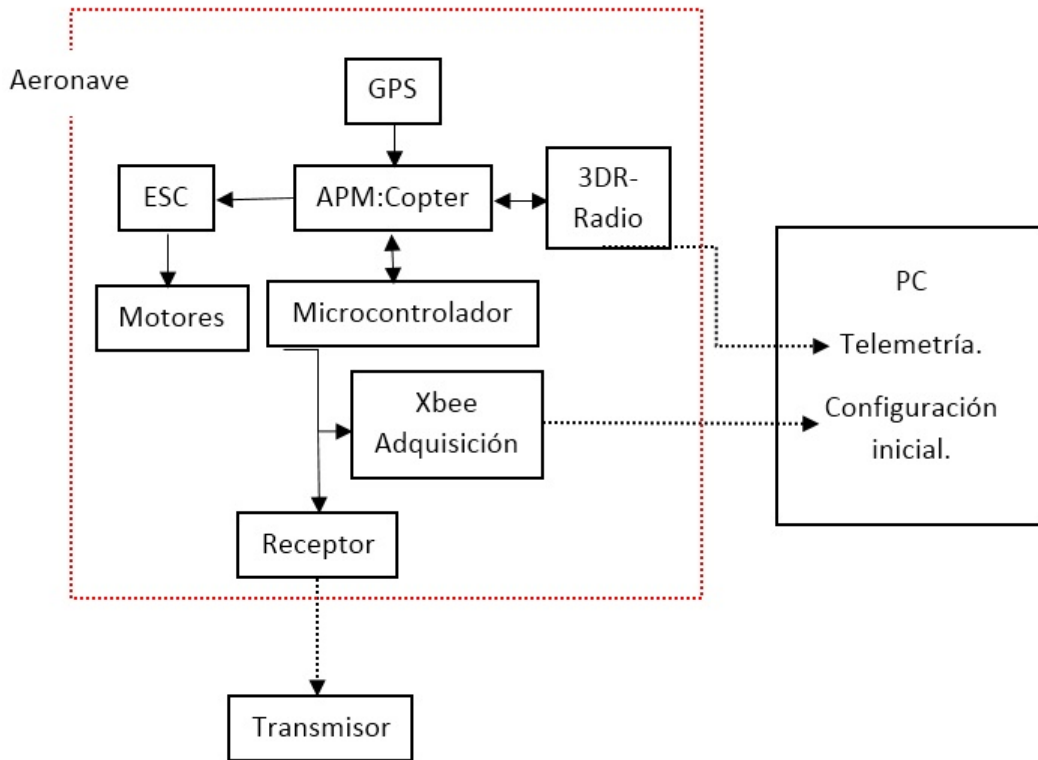


Figura 1.4: Diagrama de flujo del sistema. Fuente: Autores

En el microcontrolador se implementa el algoritmo, este dispositivo programable recibe las señales del radiocontrol por medio del receptor y de acuerdo al modo de vuelo: las replica si está en modo manual o las modifica de acuerdo a la rutina del modo automático. Además gestiona la comunicación inalámbrica serial por medio de un *Xbee* que va a un computador, mencionado anteriormente.

Capítulo 2

Descripción de quadrotor

Teniendo en cuenta las condiciones y el estado del vehículo desarrollado en el año 2010[1], se decide realizar una adecuación, consiste en mejorar la estructura física utilizando un material más resistente, flexible y de poco peso, emplear motores de mayor potencia con sus respectivas hélices y baterías de alta capacidad de carga.

2.1. Estructura

Básicamente la estructura de un quadcopter está dada en forma de cruz (Figura 2.1), con el propósito de mantener una simetría y un centro de masa en el punto donde se interceptan los brazos del vehículo.



Figura 2.1: Estructura básica. Fuente: <http://fpvlongrange.com>

Inicialmente se desarrolla una estructura similar al marco comercial X525(Figura

2.2) pero teniendo las siguientes consideraciones:

- Los soportes verticales no están en los extremos de los brazos del vehículo, sino que se ubican a una pequeña distancia detrás de los motores. Esto se acoplan alrededor de los "brazos" de aluminio y permiten un cableado interno entre los ESC y los motores, es decir, estos conductores están dentro de los tubos.
- Mayor tamaño y agarre al brazo de la pieza base o soporte para los motores, debido a que en esta investigación se van a utilizar *actuadores* de mayor peso y tamaño (Ver sección 2.2).
- La sección central se elabora en forma circular, aquí se ubican los componentes electrónicos y las etapas de potencia. El objetivo es tener una simetría aceptable y un centro de masa aproximadamente en esta región, factor importante en el control y la maniobrabilidad de vuelo; además, al igual que en la primera consideración, la sección se acopla alrededor de los tubos para permitir el cableado.



Figura 2.2: Estructura comercial y modificada. Fuente: <http://www.hobbyking.com> y autores.

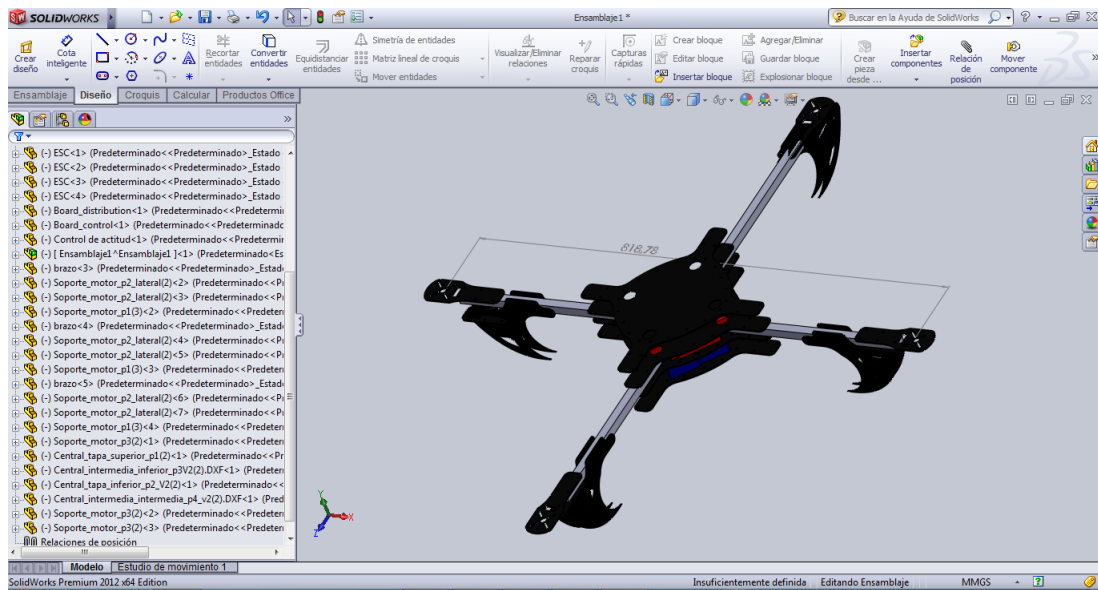


Figura 2.3: Modelo 3D en SolidWorks. Fuente: Autores.

Las secciones o piezas que conforman la estructura en la parte central y que soportan los componentes electrónicos se presentan en las Figuras 2.4 y 2.5.

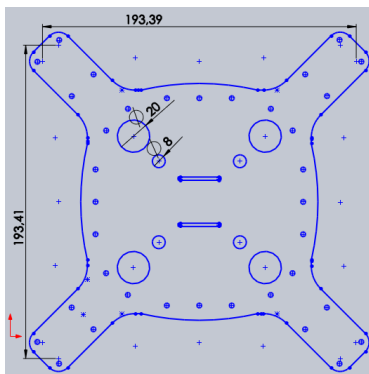


Figura 2.4: Pieza central superior.
Fuente: Autores.

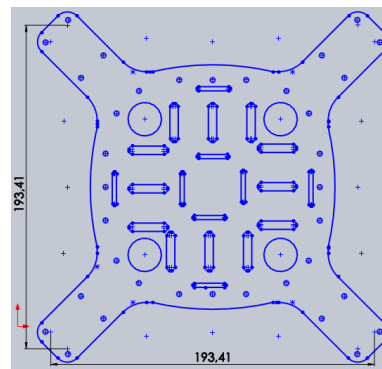


Figura 2.5: Pieza central intermedia.
Fuente: Autores.

Los soportes para los motores y para apoyar la estructura en la superficie donde se encuentre, se presentan en las Figuras 2.6 y 2.7.

Motor	Potencia [W]	Kv[rpm/V]	Corriente máxima [A]	Precio
Turnigy G15	500	950	34	27.47
D4023-850	380	850	35	13.13
Series 42-38	785	750	55	25.99
Turnigy Multistar 2834	660	800	45	42.38
NTM Prop Drive 28-26	315	1000	21	15.99
Brushless Outrunner 3536	450	850	45	13.08
Turnigy G25	600	610	44	21.45
NTM Prop Drive 35-48	815	900	55	21.30
Turnigy Multistar 3508	550	640	30	29.78

Tabla 2.1: Muestra de motores disponibles en el mercado. Fuente: <http://www.hobbyking.com>

Se selecciona el *Series 42-38* (ver Figura 2.8) ya que es el segundo mayor en potencia y tiene una relación de velocidad-tensión baja (mayor empuje), es decir, el torque que puede ejercer es considerable, además el costo es aceptable.



Figura 2.8: Motor seleccionado. Fuente: <http://www.hobbyking.com>

Las especificaciones completas del motor seleccionado son:

Modelo	NTM Prop Drive Series 42-38
Kv	750rpm/V
Corriente máxima	55 A
Potencia máxima	403W a 12V(3S) / 785W a 15V(4S)
Eje	5mm
Peso	169g
ESC	60A
Batería	3s-4s Lipo
Conexión	Conector bala de 3.5mm

Pruebas entregadas por el fabricante:

Hélice	Tensión [V]	Potencia [W]	Corriente [A]	Empuje [kg]
13x10	11.1	344	31	1.79
13x10	14.8	740	50	2.8
14x6	11.1	403	36	2.23
14x6	14.8	785	53	3.05

Las hélices seleccionadas son de tamaño 14x4.7 pulgadas de *APC Propellers*¹. (Ver Figura 2.9).



Figura 2.9: Hélices. Fuente: Autores.

2.3. Control electrónico de velocidad

Este circuito hace parte de la etapa de potencia y control de los motores, convierte la tensión continua de entrada en trifásica. Los ESC (Electronic Speed Control) son utilizados generalmente en aplicaciones de radio control

¹Encargada a la fabricación de hélices para aeromodelos

y en motores sin escobillas. Para el control de la velocidad, generan pulsos PWM en función de las señales que provienen del radio control situado en tierra.



Figura 2.10: ESC seleccionados. . Fuente: Autores.

Para la selección de este dispositivo, se consideran las especificaciones de corriente máxima que puede entregar y la tensión de alimentación, los *HobbyKing Red Brick* (Figura 2.10) pueden entregar hasta 70A y se energiza con una batería entre 2 a 7 celdas, por lo tanto, es aceptable para controlar los motores seleccionados.

2.4. Batería

Los parámetros de selección para una batería son:

- La tensión de salida.
- La capacidad eléctrica. Con este parámetro se determina el tiempo que tardará en descargarse teniendo un consumo específico.
- El tipo de batería: Ni-Cd(Níquel-cadmio), Ni-MH(Níquel-metal-hidruro), Li-ion (Iones de litio), Li-Po(Polímero de litio).

Se utilizan baterías Li-Po principalmente por su gran densidad de carga, en poco espacio y peso. La batería seleccionada se observa en la Figura 2.11.

Capacidad	6000mAh
Tensión	4S1P / 4 Celdas / 14.8V
Descarga	Constante de 25C / 50C Burst
Peso	623g (incluye cables y conector)
Dimensiones	175x49x38mm

Tabla 2.2: Características de la batería



Figura 2.11: Batería. Fuente: Autores.

Teniendo las anteriores consideraciones, se procede a realizar la estructura, inicialmente se busca realizar pruebas de estabilidad, tamaño, forma de ensamble, consistencia y simetría; para ello se utiliza un material de prueba y que el momento estaba al alcance del grupo de trabajo. Con láminas de acrílico de color negro se elaboran las tres piezas que conforman la parte central, 4 soportes para los motores y 8 soportes verticales que van a estar alrededor de los tubos como se ha mencionado anteriormente, se realiza la compra de 2.5 metros de tubo cuadrado de aluminio de media pulgada de espesor y se efectúa el ensamble con los motores, los ESC y las baterías (ver Figura 2.12).

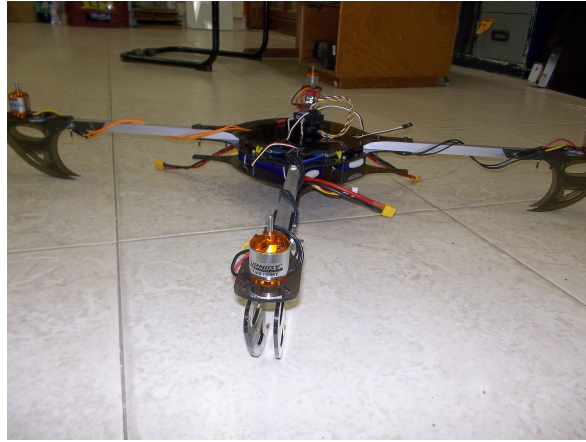


Figura 2.12: Estructura y ensamble de prueba. Fuente: Autores.

Los resultados de esta construcción de prueba son:

- Comprobación del modelo con respecto a la ubicación de los componentes: ESC, baterías, motores, cables, etc.
- Confirmación de los puntos de tensión debido a la presencia de fuerzas o impactos. De acuerdo a este resultado se decide ampliar el área de agarre de las secciones centrales y de los soportes de los motores, además, estos últimos van a tener doble capa de material.

La estructura final se puede observar en la Figura 2.13.



Figura 2.13: Estructura y ensamble final. Fuente: Autores.

Capítulo 3

ArdupilotMega 2.5

3.1. APM 2.5

El ArduPilotMega es la versión más reciente de piloto automático diseñado por *Chris Anderson* y *Jordi Muñoz*, de DIY Drones.

Este módulo electrónico con tiene principalmente un dispositivo programable, sensores inerciales y de posicionamiento angular y un Sistema de Posicionamiento Global, GPS. La agrupación y organización de estos componentes permite establecer un adecuado control de vuelo en un vehículo aéreo no tripulado (UAV), por lo tanto, se encarga de la estabilización del vehículo. También se puede utilizar en modo *"fly-by-wire"*, esto quiere decir, que puede estabilizar un avión al volar manualmente bajo el control de RC, por lo que es más fácil y seguro.

El microcontrolador usado, ATmega2560 - 16MHz, es compatible con el entorno de programación Arduino. De esta forma, se puede modificar el código proporcionado por el desarrollador que al igual que el hardware es de código abierto (open source).

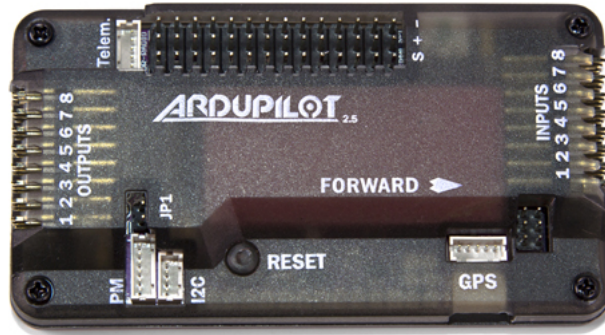


Figura 3.1: Hardware ArdupilotMega 2.5. Fuente www.copter.ardupilot.com

3.2. Sensorica del ArdupilotMega 2.5

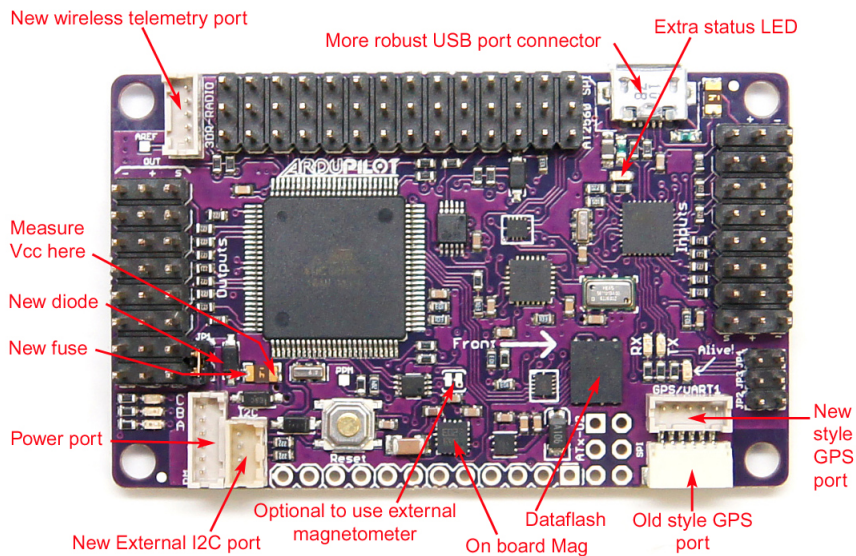


Figura 3.2: ArdupilotMega Sensorica y Hardware interno. Fuente: www.copter.ardupilot.com

En cuanto al hardware que lo constituye y que permite establecer una caracterización de los movimientos de la aeronave se encuentra un acelerómetro

de 3 ejes, giroscopio de 3 ejes, magnetómetro de 3 ejes, sensor de presión absoluta y de un GPS(externo) para ubicar el quadcopter.

El APM 2.6 es una versión que tiene un magnetómetro (compass) externo, es decir, no posee un compass integrado; es una mejora u optimización para vehículos donde el compass debe ubicarse lo más alejado de las fuentes de ruido magnético tales como los motores evitando distorsiones en la lectura.

3.3. Mission Planner

Mission Planner es software libre, es decir, de código abierto disponible para Windows. Debe ser instalado para poder programar el ArdupilotMega 2.5. Permite configurar el hardware, las características de vuelo y visualizar parámetros (telemetría) antes, durante y después de las misiones.

Las instrucciones de descarga e instalación se pueden encontrar en la página APM¹.

3.3.1. Firmware dentro del APM 2.5

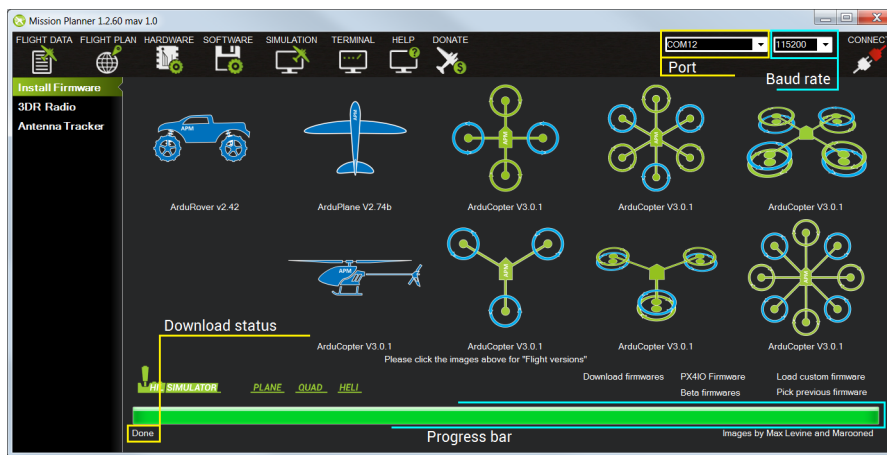


Figura 3.3: Mission planner Firmware. Fuente: www.copter.ardupilot.com

El controlador esta desarrollado para trabajar en diferentes plataformas o vehículos, se debe cargar el Firmware correspondiente a la estructura del UAV por medio de Mission Planner. A través de una serie de iconos (Figura 3.3) es posible seleccionar la última versión correspondiente al vehículo de

¹<http://copter.ardupilot.com/wiki/initial-setup/common-install-mission-planner/>

interés (Quad, Hexa, Y6, plane, rover). En este caso se ha realizado la carga del firmware correspondiente a un ArduCopter(quadrotor).

3.3.2. Configuración iniciales

El sistema tiene configuraciones iniciales que son de carácter obligatorio para el posterior uso del vehículo. El sistema Ardupilot soporta también un gran número de hardware adicional el cual también deberá configurarse posteriormente. Para obtener información más detallada sobre la calibración del hardware adicional puede visitar el sitio web <http://copter.ardupilot.com/wiki/initial-setup/configuring-hardware/>.

3.3.3. Calibración del Acelerómetro

Configurar el acelerómetro es fundamental, permite tener una adecuada referencia en cuanto a la inclinación y/o nivel de la estructura (estado de actitud) que es usada durante el vuelo.

Este procedimiento requiere que el vehículo sea llevado a las posiciones mostradas en la Figura 3.4.

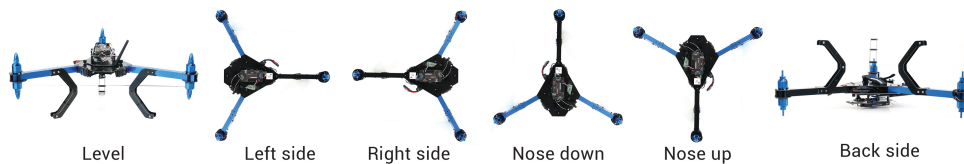


Figura 3.4: Posiciones de Calibración del Acelerómetro. Tomada de: www.copter.ardupilot.com

3.3.4. Calibración del Magnetómetro

Durante 60 segundos se debe girar el quadrotor hacia cada lado (frente, espalda, izquierda, derecha, boca arriba y boca abajo), esto permite realizar una recopilación de magnitudes del campo magnético en cada eje, genera una serie de valores proporcionales (offset) a las interferencias en el lugar realizada la calibración y corrigen la lectura del Magnetómetro. La secuencia de calibración se muestra en la figura 3.5.

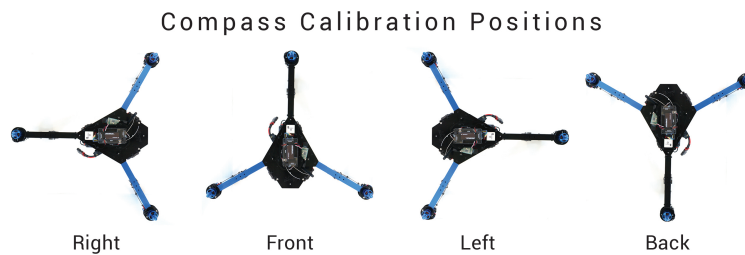


Figura 3.5: Posiciones de Calibración del Magnetómetro

3.3.5. Comunicación con GCS

La estación de tierra tiene como tarea principal tener un vínculo de enlace con el Ardupilot. El objetivo es monitorizar los parámetros de vuelo en todo momento, incluido antes del despegar. La comunicación está constituida por dos radios con una comunicación serial de 57600 bps con el protocolo UART. La transmisión de datos inalámbrica permite al usuario disponer de los datos enviados por telemetría hacia la interfaz de Mission Planner y mostrar dicha información de manera que sea productiva para mejorar las acciones de control y desarrollar misiones deseadas.

3.4. Mavlink Micro Air Vehicle Communication Protocol

3.4.1. Protocolo MAVLink

MAVLink es un protocolo de mensajes, se caracteriza por tener un *cálculo ligero de referencias* optimizado para vehículos aéreos no tripulados. La codificación eficiente también permite ejecutar el protocolo en microcontroladores.

MAVLink ya ha sido adoptado por un número de otros sistemas entre ellos se encuentran: pxIMU piloto automático, ArduPilotMega piloto automático, SLUGS piloto automático, piloto automático UDB. Las sentencias MAVLink se generan en base a un archivo de especificación del formato XML en el protocolo MAVLink. El generador de código asegura que los mensajes estén bien constituidos y ofrece código en C - compatibles - C89 para empaquetar y desempaquetar el mensaje.

Obtención de los paquetes

Esta es la anatomía de un paquete de MAVLink.

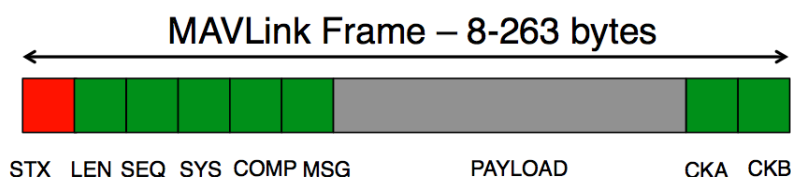


Figura 3.6: Anatomía de los paquetes de Mavlink. Fuente: <https://pixhawk.ethz.ch/mavlink/>

Nombre de Campo	(Byte)	propósito
Inicio del mensaje	0	Este byte denota el principio de la transmisión (v1.0: 0xFE)
Longitud del mensaje	1	longitud del mensaje payload
Secuencia de los paquetes	2	Each component counts up his send sequence. Allows to detect packet loss
Identificador del sistema	3	Identificación del sistema que esta enviando la transmisión. Permite diferenciar diferentes sistema en la misma red.
Component ID	4	Identification of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
Identificador del mensaje	5	Identificación del mensaje - el identificador define que tipo de información viene en el payload y como puede ser correctamente decodificado.
Información (PAYLOAD)	6 to (n+6)	Los datos dentro del mensaje, dependerá del tipo del mensaje ID del mensaje.
CRC	(n+7) to (n+8)	Comprobador de suma del paquete entero, excluyendo el primer byte (byte 0) o start byte. (LSB to MSB)

Tabla 3.1: Anatomía de los paquetes MavLink. Fuente: <https://pixhawk.ethz.ch/mavlink/>

De acuerdo a los requerimientos (Sección 4.1), para obtener la información de los sensores inerciales y de los valores de la longitud y latitud proporcionados por el GPS, como se explicó en la sección anterior 3.3.5 el Arducopter se comunica con MissionPlanner a la Estación de tierra (GCS) mediante una comunicación serial inalámbrica, la información o paquetes proveniente son todas las variables de control y monitoreo del ArduPilot en el formato de MavLink.

Para capturar los datos mencionados, se cuenta con un hardware 5 capaz de discriminar los mensajes y seleccionar los que poseen dicha información. Como se tiene en la sección 3.3.5 hay 255 tipos de mensajes y solo 2 de los mensajes llevan los datos requeridos, uno de ellos es: mensaje ATTITUDE con identificador de mensaje (#30) contiene la información de yaw, el segundo tipo de mensaje es GLOBAL_POSITION_INT (#33) donde se encuentra

la información referente a la posición dada por coordenadas geográficas *Latitud* y *longitud*; esto conforma la información suficiente para desarrollar un algoritmo de trayectorias.

ATTITUDE (#30)

The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right).

Nombre de Campo	(Tipo)	Descripción
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
roll	float	ángulo Roll (rad, -pi..+pi)
pitch	float	ángulo pitch (rad, -pi..+pi)
yaw	float	ángulo yaw (rad, -pi..+pi)
rollspeed	float	Velocidad angular de roll (rad/s)
pitchspeed	float	Velocidad angular de pitch (rad/s)
yawspeed	float	Velocidad angular de yaw (rad/s)

GLOBAL_POSITION_INT (#33)

Nombre de Campo	(Tipo)	Descripción
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
lat	int32_t	Latitud, expresada en *1E7
lon	int32_t	Longitud, expresada en *1E7
alt	int32_t	Altitud en metros, expresada en *1000 (milímetros), above MSL.
relative_alt	int32_t	Altitude above ground in meters, expressed as * 1000 (milímetros)
vx	int32_t	Velocidad en X (Latitude), expresada en m/s * 100
vy	int32_t	Velocidad en Y (Longitud), expresada en m/s * 100
vz	int32_t	Velocidad en Z (Altitud), expresada en m/s * 100
hdg	int32_t	medida del cabeceo en grados * 100, 0.0..359.99 grados.

Para obtener mas información acerca de el set completo de mensajes se puede visitar la página <https://pixhawk.ethz.ch/mavlink/>.

Capítulo 4

Diseño del algoritmo para generar la ruta

El término algoritmo está definido como un "*conjunto ordenado y finito de operaciones que permite hallar la solución de un problema*" [17], esto consiste en llevar a cabo tareas específicas teniendo en consideración aspectos característicos del problema y las acciones que satisfacen la presente necesidad [5]. Primero que todo, se necesita aclarar la situación en la que se encuentra el sistema, esta etapa se conoce como *especificaciones del problema* y permite organizar mejor la información, hacer un *análisis del problema* y plantear una solución.

4.1. Especificaciones del problema.

Teniendo en cuenta las características del sistema (Capítulo 1) y los recursos que ofrece el módulo controlador de vuelo (Capítulo 3), las condiciones en las que se ha configurado el vehículo están dadas, de tal forma, que se tiene un vuelo controlado mediante un radiotransmisor y se posiciona en un punto determinado del espacio (*coordenadas iniciales*) como se ilustra en la Figura 4.1.

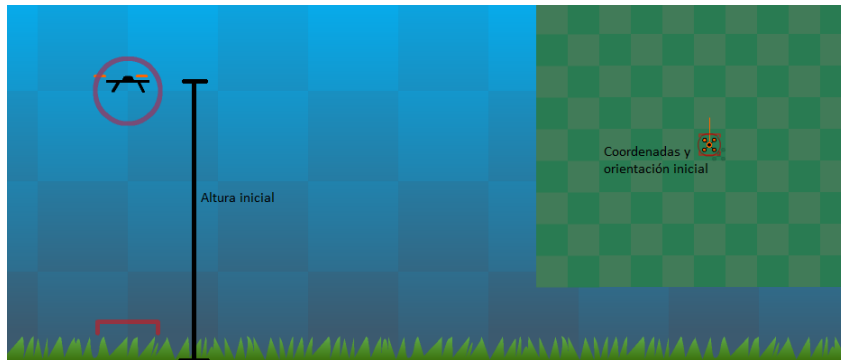


Figura 4.1: Posición inicial del vehículo. Fuente: <http://jasonshort.com/>

Entonces como información inicial se tiene: latitud, longitud geográficas, orientación geográfica (ángulo) con respecto al norte terrestre y la altura; esto constituye el estado inicial, antes de que se accione el *modo automático* para generar la ruta o trayectoria.

A partir de estas condiciones iniciales, se necesita determinar la secuencia de acciones que se deben ejecutar para que el UAV se traslade y posicione en una segunda coordenada la cual llamaremos *coordenada deseada*. Primero, se debe orientar hacia el punto deseado (Figura 4.2), este ángulo se denomina *yaw deseado* debido a que el movimiento es de tipo guiñanda (yaw). Finalizando el movimiento anterior, se procede a generar el desplazamiento por medio de variaciones en el *ángulo pitch* o cabeceo (ver Sección 1.1).

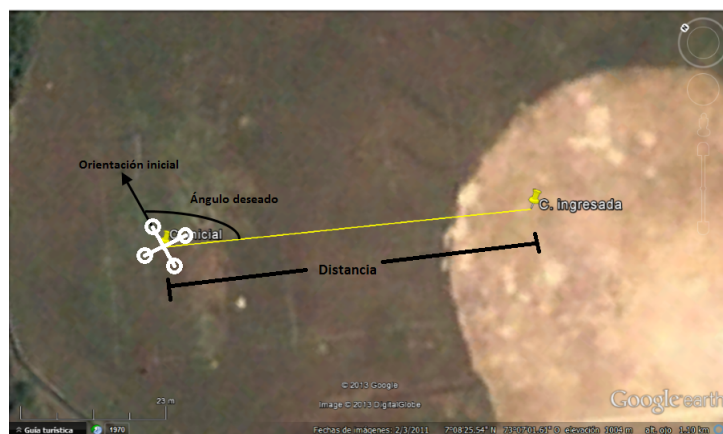


Figura 4.2: Secuencia de orientación y traslado. Fuente: Autores.

4.2. Análisis del problema

Como se presentó en la sección anterior, se necesita generar en el vehículo dos tipos de movimientos: uno rotacional y otro traslacional; estas acciones se realizan respecto al clásico sistema de coordenadas geográficas que utiliza la longitud y la latitud para el posicionamiento, entonces, de acuerdo al problema planteado se debe tomar en consideración 3 coordenadas: las finales o destino, las coordenadas iniciales y las coordenadas instantáneas; las dos primeras son constantes en cada prueba de vuelo, mientras que la última varía de acuerdo a las instrucciones que constituyen el algoritmo.

El movimiento de guiñada tiene como objetivo obtener una orientación hacia la coordenada final, principalmente se basa en establecer un ángulo respecto al *norte terrestre*. Antes de realizar acciones sobre los actuadores, es necesario calcular la magnitud angular mencionada, se puede determinar a partir de las coordenadas finales y las instantáneas (Sección 4.3.2). De esta manera, se pueden accionar los rotores haciendo que la estructura gire una cantidad equivalente al ángulo calculado, por lo tanto, se debe realizar periódicamente una comparación entre la orientación instantánea y la deseada en todo el recorrido del vuelo.

El movimiento traslacional se debe efectuar luego de que la estructura tenga una adecuada orientación, con el objetivo de mantener el marco de referencia interno (frente-atrás-izquierda-derecha) y realizar una ruta aproximadamente en forma de línea recta (Figura 4.2). Es suficiente realizar este movimiento por medio de la inclinación del *cabeceo* (ángulo pitch) de la aeronave, dependiendo de la cantidad de los grados de esta inclinación el UAV se va a trasladar a una determinada velocidad. Es fundamental conocer las aproximaciones del vehículo hacia la coordenada final, se puede realizar una comparación entre las longitudes y latitudes finales e instantáneas. Considerando la coordenada deseada como el punto central y la coordenada instantánea del UAV como el polo de una esfera, entonces se puede calcular el radio existente que corresponde a la distancia variable entre los puntos mencionados.

4.3. Algoritmo determinista

Un algoritmo determinista hace referencia a una secuencia de reglas o procesos que producen siempre la misma salida para una determinada entrada, son el tipo de algoritmos más estudiados y que más se utilizan en la práctica, son altamente eficientes [6].

4.3.1. Medición de distancias. Selección del método.

En la actualidad, existen diversas formas de conocer la distancia entre dos coordenadas geográficas y la precisión es aceptable, aunque este último factor depende de la aplicación; se puede optar por utilizar bases de datos que contengan la información necesaria, para esto se necesita sobresalientes recursos de almacenamientos y acceso (conexión) a internet. Teniendo en cuenta las disposiciones que ofrece el hardware, como la precisión del GPS (Capítulo 5), los recursos del sistema de procesamiento y lo que corresponde a la adquisición de datos, como el tipo de variables y la información que representan, se evidencia que es viable un sistema de cálculo menos robusto y con los datos adquiridos es posible implementar un algoritmo de cálculo para obtener la distancia con una precisión mínima similar a la del GPS.

Método	Características
Vincenty	Son dos procesos iterativos y relacionados. Utiliza un preciso modelo elipsoidal de la Tierra. Precisión de $0,5mm$.
Haversine	El proceso es más sencillo que el Vincenty. Se deriva de la <i>ley de haversine</i> . Relaciona los lados y ángulos de un triángulo esférico.
Polar	Eficiente cálculo en los polos.
Cosenos	Es matemáticamente exacta. Inconsistente para pequeñas distancias debido al uso del coseno inverso ($d = R \arccos(x)$).

Tabla 4.1: Métodos matemáticos para el cálculo de la distancia. Referencia <http://search.cpan.org/>

En la Tabla 4.1 se presentan los métodos representativos por su eficiencia y poca complejidad para realizar el cálculo de la distancia entre dos coordenadas geográficas. Las fórmulas de Vincenty y Haversine son las más precisas, la primera emplea un proceso iterativo para acotar el error bajo un valor, entonces la velocidad del cálculo es mayor que la segunda fórmula y en el sistema de vuelo que se presenta es de suma importancia mantener al mínimo en retardos en los procesos.

Se decide seleccionar, por su equilibrio en velocidad y precisión, la *fórmula de Haversine* como método para calcular la distancia entre las dos coordenadas.

Fórmula del Haversine

Es una ecuación matemática aplicada a la navegación, por medio de este cálculo se determina la distancia del mayor círculo entre dos puntos (coordenadas) que pertenecen a una esfera[7]. Esta fórmula se deriva de la *trigonometría esférica*, por medio de la relación de lados y ángulos en triángulos esféricos (ley de haversine).

La función de haversine está dada por:

$$\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos \theta}{2}$$

Las coordenadas geográficas A y B están compuestas por una latitud τ_1 y una longitud λ_1 para el punto A y τ_2 , λ_2 para el punto B (ver Figura 4.3). Por medio de esta información, se puede calcular el ángulo central entre los dos puntos, Ecuación 4.1.

$$\text{haversin}\left(\frac{d}{r}\right) = \text{haversin}(\tau_1 - \tau_2) + \cos(\tau_1) \cos(\tau_2) \text{haversin}(\lambda_2 - \lambda_1) \quad (4.1)$$

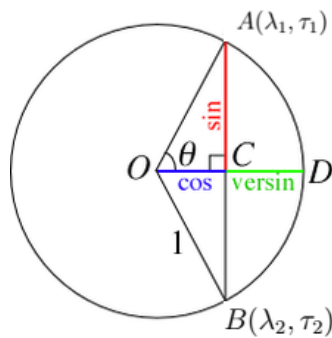


Figura 4.3: Triángulo esférico. Fuente: Autores.

De la Ecuación 4.1, se puede extraer la distancia, d , entre las dos coordenadas, que es equivalente al arco de la esfera con radio r . Estos cálculos asumen al planeta Tierra como un elemento esférico, se ignora los efectos elípticos; a pesar de esta aproximación, la precisión es suficiente para los propósitos de la aplicación; el error típico es de 0,3 % [8].

La distancia de forma explícita se expresa en la ecuación 4.2.

$$h = \sin^2 \left(\frac{\Delta\tau}{2} \right) + \cos \tau_1 \cos \tau_2 \sin^2 \left(\frac{\Delta\lambda}{2} \right)$$

$$distancia = r \arcsin \sqrt[3]{h} \quad (4.2)$$

Donde $\Delta\tau = \tau_1 - \tau_2$ y $\Delta\lambda = \lambda_1 - \lambda_2$, r varía entre 6356,752 km (en los polos) a 6378,137 km en el Ecuador.

4.3.2. Cálculo de la orientación

La orientación del vehículo se representa mediante el ángulo, ψ , es la posición angular respecto al norte terrestre; a través de la rotación de la estructura en el eje z se orienta el frente hacia el punto geográfico deseado.

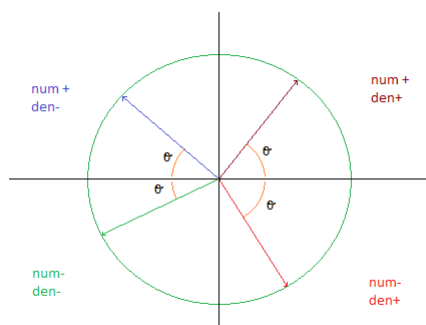


Figura 4.4: Cálculo del ángulo yaw deseado. Fuente: Autores.

El cálculo de la mencionada posición angular se realiza a partir de las coordenadas deseadas y las instantáneas, se obtiene un ángulo que denominamos *yaw_deseado*. Si se supone que la aeronave se encuentra en un punto geográfico que corresponde al origen de un plano xy o cartesiano, como se ilustra en la Figura 4.2, se puede conocer el ángulo respecto al eje horizontal, θ , de acuerdo a la definición de la función trigonométrica *tangente* (Ecuación 4.3).

$$\theta = \arctan \left(\frac{\Delta latitudes}{\Delta longitudes} \right) \quad (4.3)$$

Donde $\Delta latitudes$ y $\Delta longitudes$ es la diferencia entre las coordenadas deseadas y las coordenadas instantáneas del UAV. La parte superior del eje

vertical es equivalente al norte terrestre, el ángulo respectivo, ψ , se calcula por medio una ecuación que puede variar de acuerdo a dos condiciones: cuando la coordenada deseada se encuentra en el primero y cuarto cuadrante (Ecuaciones 4.4) o cuando se encuentra en el segundo y tercer cuadrante (Ecuaciones 4.5).

$$\psi = 90 - \theta \quad (4.4)$$

$$\psi = 270 - \theta \quad (4.5)$$

4.4. Estrategia para la generación de movimientos

Teniendo en cuenta las consideraciones y necesidades mencionadas en las anteriores secciones, se hace necesario adecuar la velocidad de los motores para modificar el ángulo denominado ψ y orientar el vehículo.

La estrategia que se utiliza para lograr este movimiento se desarrolla a partir de la alteración de los parámetros de entrada del *APM-Copter*. Como se presenta en el Capítulo 3, en el módulo de estabilización se tienen señales de entrada para maniobrar el UAV; el movimiento de guiñada (*yaw*) está en función de la señal correspondiente al cuarto canal y es la que se va a modificar para posicionar la estructura en el *yaw deseado*.

Las señales se caracterizan por tener un periodo constante de $22,6[ms]$ y un *ciclo de trabajo* variable, esto corresponde a señales PWM (*Pulse Width Modulation*). El rango del ciclo de trabajo, DC , es de 1 a $2ms$; el estado neutral, MID , para el cuarto canal es de $1,5ms$. Por consiguiente, si $DC < MID$ la rotación se desarrolla hacia la izquierda a una determinada velocidad dependiendo de la diferencia $MID - DC$, similarmente se genera el movimiento hacia la derecha ($DC > MID$).

4.4.1. Comportamiento del ángulo ψ

Para establecer las características y la forma como reacciona el vehículo ante una entrada se genera la respuesta ante una entrada tipo *escalón* (Figura 4.5). La entrada escalón que se observa de color rojo hace referencia a la señal del cuarto canal, como se ha mencionado anteriormente, es la que genera el movimiento en el ángulo *yaw*; el ciclo de trabajo máximo permitido por el

sistema es de $2ms$, se asume como el 100 % y el mínimo, $1ms$ como el 0 %, en la Figura se presenta una entrada escalón de 65 %.

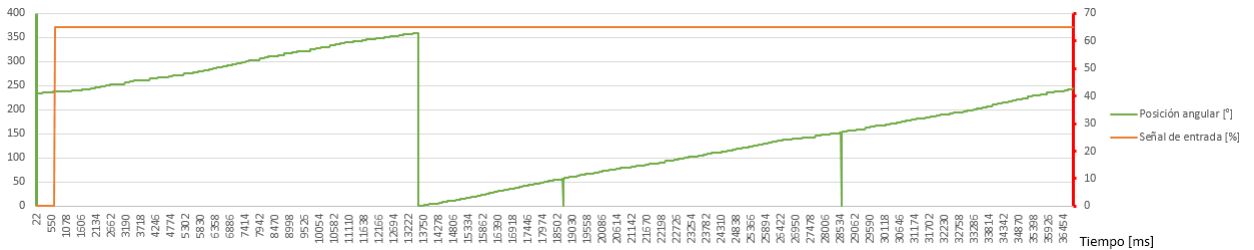


Figura 4.5: Respuesta a la entrada escalón en lazo abierto.

Fuente: autores.

El comportamiento del ángulo yaw, ψ , en lazo abierto ante una entrada tipo escalón es inestable. Esta respuesta es predecible, la señal de entrada está controlando la velocidad angular que equivale a la pendiente de la señal de color verde (posición) observada en la Figura 4.5, por lo tanto, la posición angular cambia a la mencionada *razón*.

Teniendo en cuenta lo observado, se necesita hacer un lazo de realimentación para determinar el error instantáneo de la orientación y realizar una respectiva corrección.

4.4.2. Sistema de control PID. Sintonización de los parámetros de control para el ángulo ψ

Teniendo en cuenta que no se tiene un modelo matemático de la planta, la sintonización se realiza mediante métodos experimentales. Se utiliza la segunda estrategia de *Ziegler-Nichols*[11] debido a que en la primera la planta debe tener una respuesta estable en lazo abierto. El sistema se constituye como se muestra en la Figura 4.6, el parámetro de entrada corresponde al ángulo deseado y la salida es la orientación ψ .

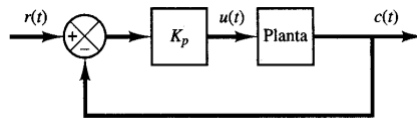


Figura 4.6: Segundo método de Ziegler-Nichols.

Fuente: [11].

El método consiste en configurar una constante proporcional K_p de tal forma que la señal de salida, ψ , oscile periódicamente y con una amplitud constante; el valor de esta constante en el control proporcional se conoce como constante crítica, K_{cr} y el periodo como periodo crítico, T_{cr} . De esta forma, se inicia el proceso de sintonización de las constantes y formular un *control PID* para regular el posicionamiento angular del vehículo. Como se observa en la Figura 4.7 se genera las oscilaciones aceptables para utilizar el método, inicialmente se propone un sistema de control PI:

$$K_p = 0,45K_{cr} = 13,5; \quad T_i = \frac{P_{cr}}{1,2} = 1,906$$

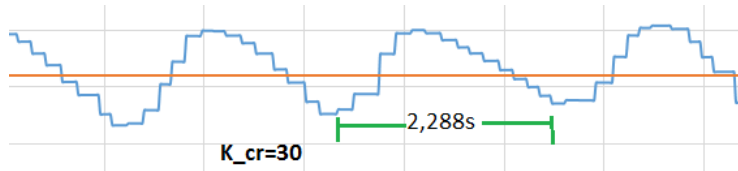


Figura 4.7: Oscilación sostenida.
Fuente: Autores.

Asumiendo el valor de los parámetros anteriores, se implementa como primera aproximación el sistema de control PI, como lo indica la ecuación 4.6, donde U_{PI} es la señal PWM del cuarto canal del APM-Copter, esta señal a sido acotada en su intervalo de correcto funcionamiento [1-2ms] para evitar desborde de la variable y excesivas acciones de control, $e[kT]$ es el error angular en la muestra k : $e[kT] = yaw[k] - yaw_{deseado}$, T es el periodo de muestreo (22 ms), las constantes de control P: $K_p = 13,5$ y en el control I: $K_i = \frac{K_p}{T_i} = 7,083$.

$$U_{PI}[kT] = K_p e[kT] + K_i \sum_{h=1}^k e[hT] - e[(h-1)T] \quad (4.6)$$

El comportamiento de la señal de salida, ψ , utilizando el control PI se observa en la Figura 4.8. Para alcanzar la referencia (línea de color azul) es necesario aumentar el peso del control integral, cuando se realiza esta acción en el vehículo se presentan perpetuas oscilaciones en estado estable, es decir, está a punto de entrar en inestabilidad; para corregir esto, se disminuye lentamente las acciones de control proporcional y se logra estabilizar el sistema, pero con un tiempo de estabilización demasiado alto (22s).

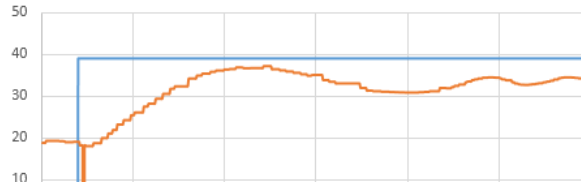


Figura 4.8: Respuesta al escalón con un control PI.
Fuente: Autores.

Para disminuir el tiempo de estabilización se adiciona al sistema un control tipo derivativo. El valor de los parámetros resultantes son $K_p = 1,003$, $K_i = 0,4$ y $K_d = 0,25$, el comportamiento se puede observar en la Figura 4.9, presenta un error de estado estable de $\pm 3^\circ$ es aceptable para generar las trayectorias, un sobreimpulso de 10° y un tiempo de estabilización de $9s$. Aclaración: las líneas verticales que aparecen en aproximadamente la muestra 600, 700 y 1100 son errores de transmisión de datos, debido a las interferencias presentes entre el receptor y transmisor de datos.

La relación error-senal de control definitiva (*Sistema de control PID*) es:

$$U_{PID}[kT] = K_p e[kT] + K_i T \sum_{h=1}^k e[hT] - e[(h-1)T] + K_d \left(\frac{e[kT] - e[(k-1)T]}{T} \right) \quad (4.7)$$

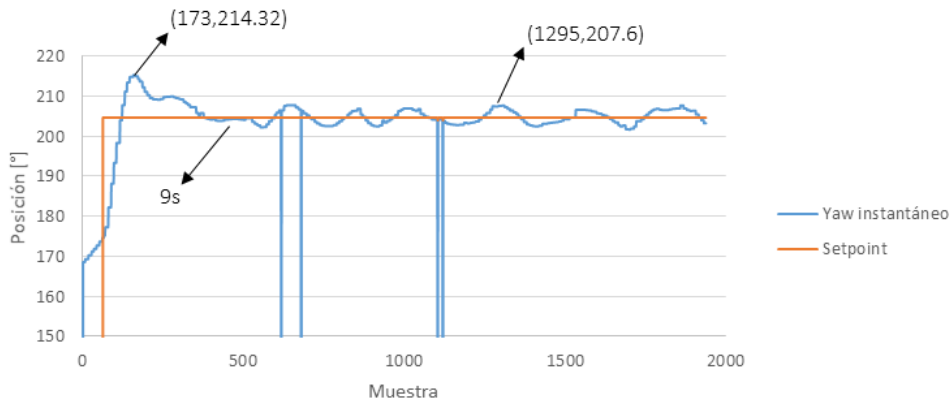


Figura 4.9: Respuesta al escalón con un control PID.
Fuente: Autores.

Las acciones de control del sistema descrito se pueden observar en la Figura 4.10, la línea de color azul es la información de la variable de control

(señal PWM) capturada con los módulos de transmisión inalámbrica *Xbee* (ver Capítulo 5), los impulsos que se observan son interferencias debido a señales de radiofrecuencias de $2,4[GHz]$ del control remoto y $433[MHz]$ del radio *3DR*. La línea de color amarillo corresponde a la tendencia de la anterior señal, esto último se realiza para visualizar mejor el comportamiento de las acciones de control y sus efectos sobre los actuadores de la planta y la fuente de energía.

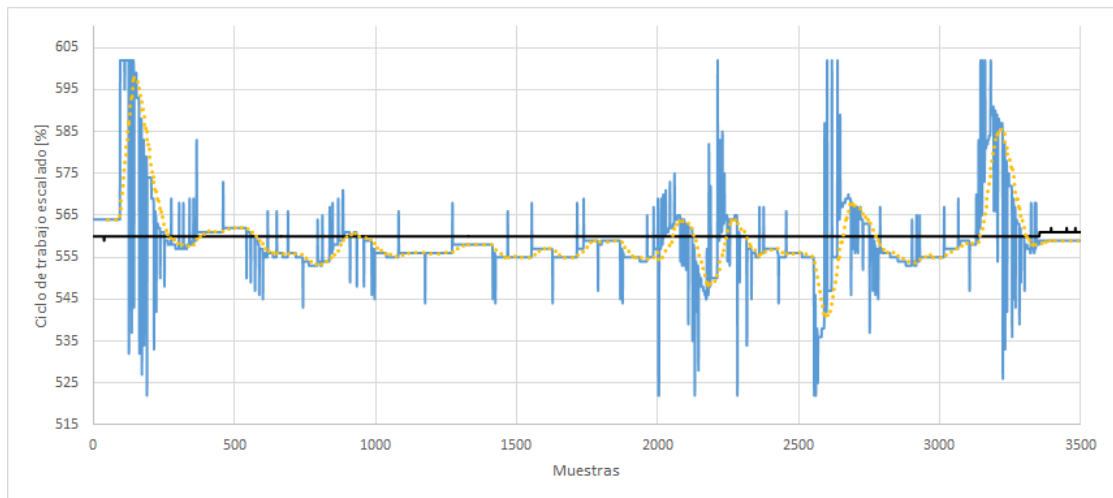


Figura 4.10: Acciones en la variable de control por el sistema PID.
Fuente: Autores.

La tendencia de la acción de control se observa explícitamente en la Figura 4.11), inicialmente se presenta un valor máximo de $598[\%]$ donde las acciones del control derivativo e integral pueden ocasionar esfuerzos excesivos sobre la planta, para evitar estas acciones que reducen la vida de los actuadores y la batería se limita las acciones de control a valores máximos y que resultan efectivos para hacer las respectivas correcciones.

Nota: El ciclo de trabajo se escala por un valor aleatorio para mejor visualización en las gráficas: $DC_{escalado} = 373300 * DC$.

Luego de orientar adecuadamente el frente de la estructura, se acciona el segundo canal para generar el movimiento hacia adelante, de igual forma, este canal recibe señales PWM.

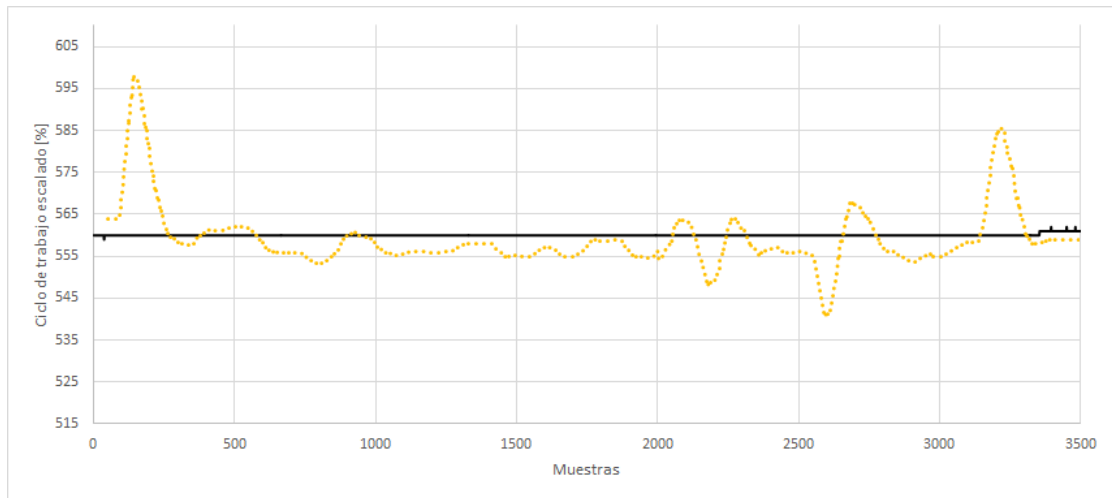


Figura 4.11: Tendencia de la acción de control para el sistema PID.
Fuente: Autores.

Se aumenta el ciclo de trabajo de la señal que se dirige al segundo canal y cuando la distancia (Sección 4.3.1) sea mínima el ciclo de trabajo volverá al estado anterior (neutral), es decir, que se tienen dos estados que dependen de la distancia entre la coordenada deseada y la instantánea para efectuar la correspondiente acción de control sobre el ángulo pitch θ . En este eje de movimiento no es necesario formular un sistema de control como el anteriormente expuesto ya que no se presenta inestabilidad, debido a que, las acciones sobre los motores están acotadas y permiten un ángulo máximo de inclinación de 45° en el eje de cabeceo y alabeo (pitch and roll), esto se configura en el módulo de estabilización *APM:Copter* como medida de seguridad y protección.

4.4.3. Sistema de control proporcional. Sintonización empírica.

En la sección anterior, se presenta la formulación de un sistema de control tipo PID, a continuación se expone otro sistema de control completamente proporcional con un tratamiento del error, esto se realiza de forma empírica y se presenta como segunda opción para el control del ángulo yaw, ψ .

El comportamiento de la señal a modificar es como lo indica la ecuación 4.8.

$$DC = MID + DC_{max} * \vartheta_{factor} \quad (4.8)$$

Donde DC representa el ciclo de trabajo, MID el estado neutral, DC_{max} el cambio máximo del ciclo del trabajo y ϑ_{factor} es un factor de proporcionalidad.

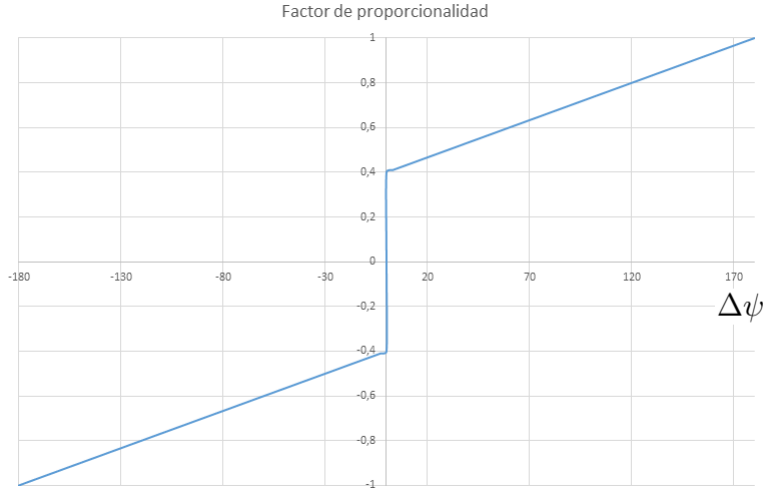


Figura 4.12: Factor proporcional al error

El parámetro ϑ_{factor} se puede caracterizar por sus máximos en magnitud cuando la diferencia ($\Delta\psi$) entre el ángulo deseado y el instantáneo es $\pm 180^\circ$ (Figura 4.12), esto quiere decir, el ciclo de trabajo va a cambiar en un factor de $\pm DC_{max}$, cuando esa diferencia disminuye, el factor hace reducir los efectos del cambio máximo en el DC . El punto de corte con el eje vertical representa un fenómeno físico, es una *zona muerta*, depende del estado y exactitud de los mecanismos del radiotransmisor, debido a que, se presentan acciones mínimas de maniobrabilidad en el mencionado transmisor y el sistema no responde, es una característica propia del módulo de transmisión; este punto de corte, η es variable y se actualiza cuando se inicia el sistema de acuerdo a las señales que recibe del control remoto, aunque, también se puede aproximar a $\frac{10}{DC_{max}}$ de acuerdo a lo experimentado. Por lo tanto, el comportamiento del factor ϑ_{factor} se presenta en la ecuación 4.9.

$$\vartheta_{factor} = \left(\frac{1 - \eta}{180} \right) * \Delta\psi + \left(\frac{\Delta\psi}{abs(\Delta\psi)} \right) * \eta \quad (4.9)$$

El término $\Delta\psi$ es el error angular del sistema y el factor ϑ está definido en función del error, esto permite que las acciones del control proporcional sean mayores cuando el error tienda, de forma proporcional al comportamiento que se observa en la Figura 4.12.

4.5. Diagrama de flujo

El algoritmo completo del sistema se presenta en la Figura 4.13, incluye el sistema de control del ángulo ψ y respectivas acciones de acuerdo a resultados de condicionales. El término *Error*, hace referencia al error angular: $\Delta\psi = yaw_{actual} - yaw_{deseado}$.

La *interrupción KBI*, *Keyboard Interrupt* son funciones que se ejecutan cuando se presentan cambios en señales de entrada, es decir, en las señales provenientes del control remoto.

La *interrupción SCI*, *Serial Communications Interface* hace referencia a funciones en ejecución cuando se presentan cambios en las señales de la comunicación serial, en este caso, estas señales contienen la información de posicionamiento y postura; se hace la adquisición y transmisión a tierra.

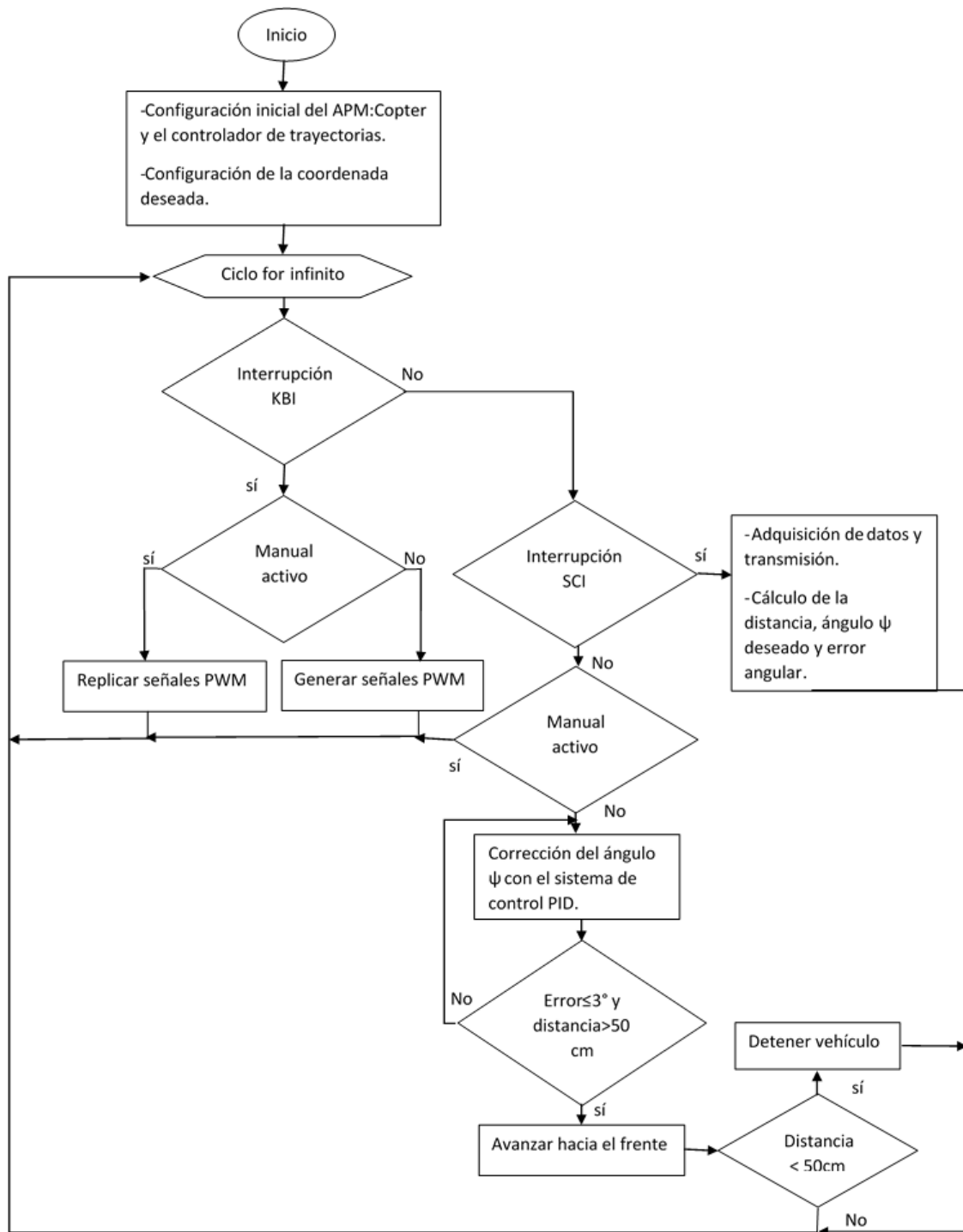


Figura 4.13: Diagrama de flujo del algoritmo generador de trayectorias.
Fuente: Autores.

Capítulo 5

Sistema implementado

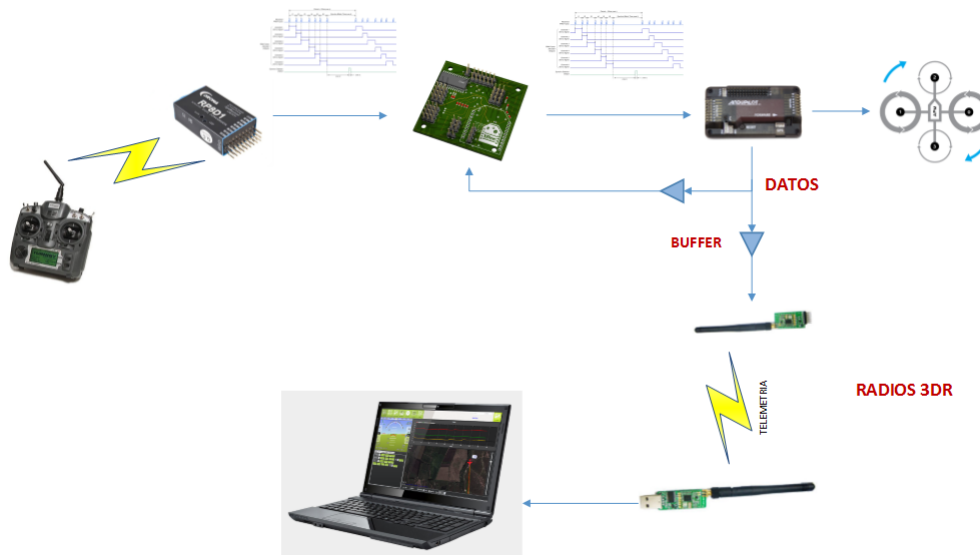


Figura 5.1: Esquema del sistema de procesamiento. Fuente: Autores.

El esquema general del sistema se presenta en la Figura 5.1, de izquierda a derecha se tiene un transmisor con su respectivo receptor. La *PCB* (*Printed Circuit Board*) recibe las señales provenientes del receptor, las transmite al módulo de estabilización *APM:Copter* y hace la adquisición de datos; simultáneamente se tiene comunicación con la estación en tierra por medio de *radios 3DR*.

5.1. Diseño de la PCB

En este capítulo se describe el diseño e implementación de la tarjeta desarrollada para la integración del hardware, adquisición y generación de señales. En la sección 3.4.1 se expone el formato y modo de transmisión de los datos, protocolo UART. Teniendo en consideración estas características y requerimientos del sistema se hace la selección de las unidades de procesamiento y los módulos que debe gestionar.

La unidad de procesamiento y los módulos que componen la tarjeta deben satisfacer los siguientes requerimientos:

- Tener la capacidad de adquirir datos utilizando un protocolo de comunicación serial.
- Enviar información por medio de un módulo de transmisión inalámbrica hacia un ordenador (Estación de Control en Tierra, GCS).
- Poder capturar cambios en las señales de entrada, conocer sus estados y tener la capacidad de duplicarlas, es decir, clonar las señales. Es necesario tener este proceso con alta prioridad respecto a los demás, por que hace parte fundamental de la maniobrabilidad y mando del UAV.
- Tener los suficientes recursos lógico-aritméticos para realizar operaciones con variables tipo flotante.

5.1.1. Elección de la unidad de procesamiento.

Con base a las características requeridas del sistema y condiciones de ensamblaje en la estructura que debe tener el dispositivo programable, se decide utilizar la tecnología que ofrece el *microcontrolador*.

Los microcontroladores son una de las herramientas más adaptables, en cuanto a software, para dar soluciones, ofrece módulos de comunicación, adquisición de señales, gestores de tiempo, etc; "existen más de 15000 millones de soluciones basadas en este dispositivo"¹ y es suficiente la tecnología que ofrecen para implementar el algoritmo propuesto. En la Tabla 5.1 se presentan los principales fabricantes y las familias de dispositivos programables que ofrecen.

¹Tomado de *Microcontroladores de 32 bits, Coldfire V1/Familia JM*, Revisión técnica:0.1, por MÚNERA HOYOS, DIEGO ALEJANDRO

Fabricante	8 bits	16 bits	32 bits
Atmel	AVR, 89Sxxxx familia similar 8051		SAM7, SAM3, SAM9, AVR32
Freescale Semiconductor	68HC05, 68HC08, 68HC11, HCS08	68HC12, 68HCS12, 68HCSX12, 68HC16	683xx, PowerPC, ColdFire
Intel	MCS-48, MCS51, 8xC251	MCS96, MXS296	
Microchip	10F2xx, 12Cxx, 12Fxx, 16Cxx, 16Fxx, 18Cxx, 18Fxx	PIC24F, PIC24H, ds- PIC30FXX, dsPIC33F	PIC32
Texas Instruments	TMS370	MSP430	C2000, Cortex-M3, TMS570
STMicroelectronics	ST 62, ST 7		STM32

Tabla 5.1: Fabricantes y familias de microcontroladores

En el mercado se encuentran compañías que ofrecen dispositivos programables y unidades de procesamiento, entre las más importantes se tiene *FREESCALE*, *RENESAS*, *ATMEL*, *RABBIT*, *DALLAS*, *INTEL*, *INFINEON*, *TOSHIBA*, *NEC*, *MICROCHIP*, *SILICON LABORATORIES*, etc; su competencia rige en torno a la capacidad de bits, velocidad, migración, los periféricos o módulos que ofrecen, memoria, la tendencia y necesidad del bajo consumo que ofrecen sus productos.

Teniendo como bases los estudios realizados en el plan de la *E3T*², el dominio de uso sobre la tecnología y sus eficientes recursos; se decide adquirir y emplear microcontroladores de *Freescale Semiconductor*. Es necesario realizar una comparación, identificar las características y determinar el dispositivo que cumple los requerimientos establecidos, ver Tabla

²Asignatura: diseño de circuitos analógicos (código 23416)

5.2.

Tamaño de la palabra	Familias	Características
8bits	S08	No reduce el rendimiento para disminuir el consumo, tiene varios modos de administrar la energía. A 2,1 V: 40-20MHz. A 1,8 V: 16-8MHz. Cuenta con puertos seriales, 8 timers/PWM, 8 canales ADC (10-bit). Se pueden encontrar de 3,6V o 5,5V. Máxima memoria interna Flash de 128kB.
	RS08	Es la versión reducida del S08. Máxima frecuencia de 20MHz. 254kB máximo de memoria flash. Se pueden encontrar módulos I2C, SPI y SCI.
	HC08	Memoria flash desde 1 a 60 KB, 8 o 10 bits de resolución en ADC, múltiples canales timer/PWM.
16bits	S12	Diseñados con capacidades de detección de eventos por medio de interrupciones.
	HCS12 , S12X	
	HC16	
32bits	ColdFire+/ColdFire	Son de los microcontroladores de 32bits que tienen más bajo consumo de potencia, ofrecen alta flexibilidad. La precisión y eficiencia se garantiza en diferentes aplicaciones. Se pueden encontrar frecuencias de operación de hasta 50MHz, 256kB de memoria flash.

Tabla 5.2: Características de los microcontroladores Freescale. Fuente: <http://www.freescale.com/>

Luego de considerar los requerimientos en los desarrollos de los algoritmos y las operaciones matemáticas (Capítulo 4), se decide seleccionar el microcontrolador *MCF51QE128* de 32 bits, con núcleo *ColdFire V1*.

CPU	ColdFire V1, 32-Bit
Frecuencia de operación	50,33-MHz (3,6V-2,1V) y 20-MHz (2,1V-1,8V). 2,1 MIPS por MHz
Fuente del oscilador	Cristal o resonador cerámico, 31,25-38,4kHz o 1-16MHz. Interno 2-50,33MHz.
Sistema de protección	<i>Watchdog</i> COP, detección de baja tensión con reset o interrupción
ADC	24 canales, 12 bits de resolución, tiempo de conversión 2,5 μ .
ACMP	Dos comparadores analógicos con interrupciones y cambios seleccionables.
SCI, SPI, IIC	Dos módulos.
TPM	Uno de 6 canales y dos de 3 canales. Captura de entrada, salida comparativa o salida PWM.
RTC	Módulo contador de 8 bits con base binaria o decimal.
Entradas/salidas	70 GPIO, 16 KBI.

Tabla 5.3: Características del microcontrolador MCF51QE128. Fuente [12].

El dispositivo se adapta para adquirir las señales necesarias (Figura 5.1). Las señales provenientes del receptor del radioenlace se conectan a entradas del microcontrolador con la capacidad de utilizar el módulo *KBI*, (*Keyboard Interrupt*), de tal modo, que no se pierdan los cambios en la señal mencionada y se mantenga el mando del UAV, aspecto principal en el sistema. Se hace necesario replicar o clonar (Modo Manual) las señales del receptor para ingresarlas al módulo de estabilización *APM:Copter*, de esta forma, se puede manipular las acciones sobre los rotores y el movimiento del vehículo; en la Figura 5.2 se observa la señal clonada (amarilla), con el osciloscopio no es posible medir la diferencia entre las dos señales debido a su alta semejanza.

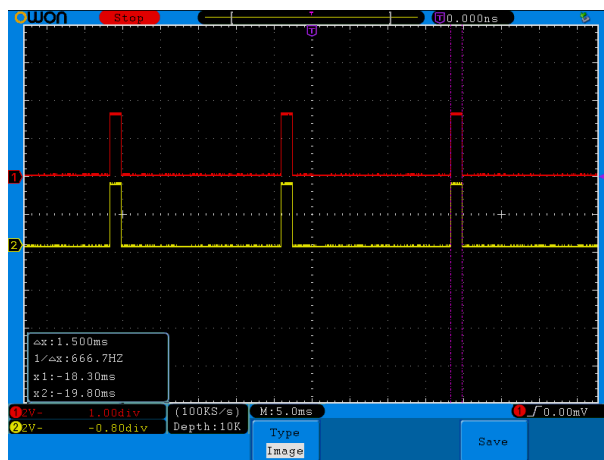


Figura 5.2: Réplica de las señales del receptor.

5.1.2. Adquisición de los datos

El protocolo de comunicación Mavlink presenta una densidad de información significativa, la señal a detectar y de donde se seleccionan los datos es la que se representa de color rojo en la Figura 5.3 comparando con la señal PWM del receptor de color amarillo.

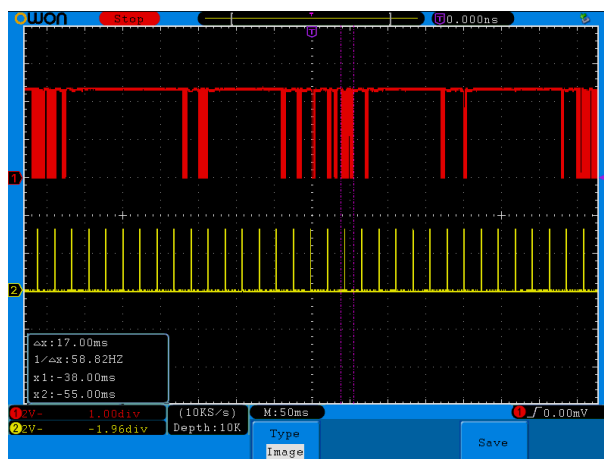


Figura 5.3: Densidad de las señales del protocolo Mavlink.

Como se observa, la densidad de las señales que contienen la información indispensable para desarrollar el algoritmo es demasiado alta y teniendo en

cuenta que también se necesitan detectar las señales PWM, entonces se presentan conflictos en el tiempo entre estas dos señales, esto quiere decir que, pueden ocurrir dos casos dependiendo del grado de prioridad que se configura en las detecciones de cada señal: se puede perder información por mantener la adquisición de las señales PWM o se puede perder el control de los rotores por adquirir la información en su totalidad; en ambos casos los resultados de un vuelo en esas condiciones sería negativo y probablemente dañino.

Para dar solución a lo anterior mencionado, se utiliza un microcontrolador adicional, el *MC9S08QE32*, dedicado a la adquisición de los datos, mientras que el *MCF51QE128* se dedica al control y mando directo de la aeronave, (ver Figura 5.4 y 5.5).

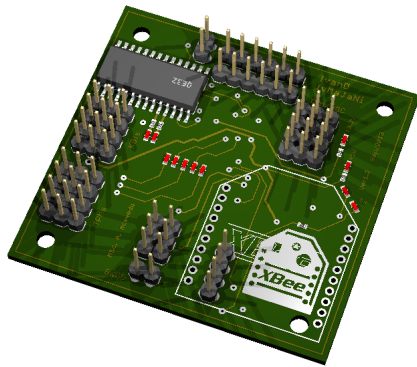


Figura 5.4: Vista superior del circuito impreso.

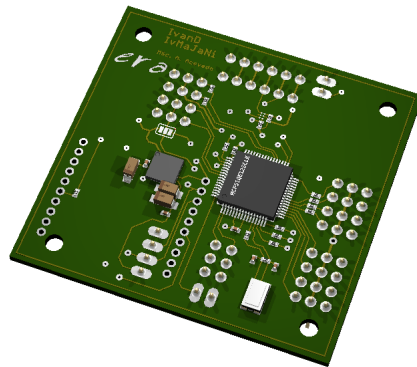


Figura 5.5: Vista posterior del circuito impreso.

Este Hardware cuenta con un módulo de comunicación inalámbrica conocido como radio 3DR por el cual se envían todas las variables que se monitorean desde la estación de tierra, dado que en esta transmisión de datos están las variables de interés, se debe capturar algunos datos de la comunicación (véase sección 3.4.1). Se duplica la señal de los datos con un buffer 74LV34 de *TEXAS INSTRUMENT*³ para conservar la integridad de los datos y con la capacidad de enviar señales de comunicación serial al radio 3DR y a la tarjeta propuesta en este proyecto.

5.1.3. Alimentación

Todos los módulos y hardware que componen el sistema tales como: APM, receptor del radio control, radio 3DR, microcontrolador, y módulo de comuni-

³ Empresa productora de circuitos integrados www.ti.com

cación XBEE; serán alimentados por medio de los reguladores internos *BEC* de los *ESC* los cuales entregan una tensión de 5 Voltios con una corriente máxima de 2 Amperios. El microcontrolador y el módulo XBEE tendrán un etapa de regulación adicional de 3.3 Voltios por medio del *TLV1117-33* de *TEXAS INSTRUMENT* ya que son la tensión nominal de estos componentes.

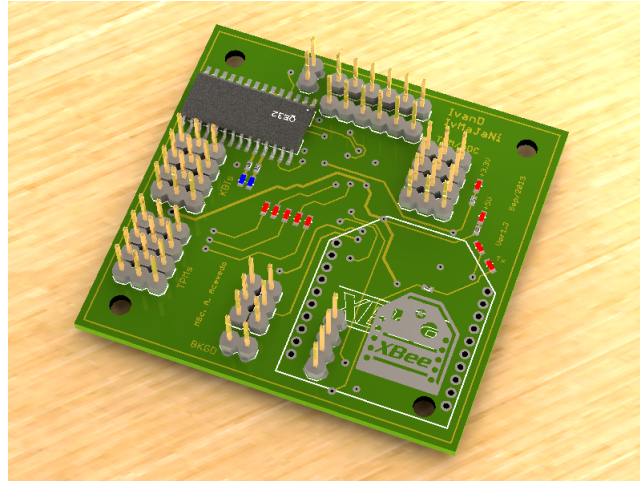


Figura 5.6: Tarjeta. Vista Eagle 3D. Fuente: Autores.

Capítulo 6

Resultados experimentales

Con el objetivo de validar el hardware diseñado y las estrategias de procesamiento de datos se desarrollan una serie de pruebas. Los experimentos se realizan en zonas deportivas de la *Universidad Industrial de Santander, UIS* en adecuadas condiciones de seguridad y con un apropiado entrenamiento de vuelo en forma manual, esto último se tiene como acción de emergencia para recuperar el UAV si el modo automático presenta fallas.

Los datos son enviados, cada 22[ms] (periodo de muestreo), por medio de una comunicación asíncrona serial a 57600 baudios, de esta forma se observa el comportamiento de parámetros que caracterizan el funcionamiento del algoritmo propuesto.

6.1. Ángulo yaw. Orientación

Inicialmente el vehículo tiene un ángulo (ψ) de $78,13^\circ$ con respecto al norte terrestre (Figura 6.1), la orientación correspondiente a la coordenada deseada es $132,9^\circ$. Se presentan oscilaciones de 10° pico-pico máximos ($\Delta\psi = 5^\circ$). Este aumento del error angular con respecto a las pruebas desarrolladas en el laboratorio es debido a la mayores interferencias, por ejemplo, las corrientes de aire presentes a $32[m]$ de altura.

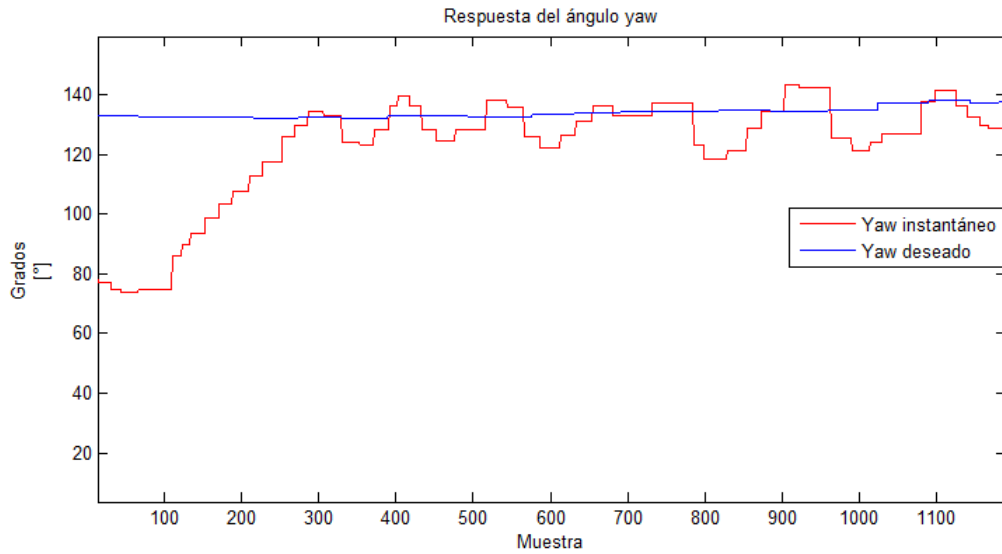


Figura 6.1: Orientación del vehículo

6.2. Traslación

La ejecución del algoritmo inicia a una distancia inicial máxima, en los datos que se presentan corresponde a $46,38[m]$, se mantiene aproximadamente constante mientras orienta la estructura, cuando alcanza el ángulo (punto **A**) aumenta la velocidad hacia el frente y se mantiene constante (segmento **A-B**). Idealmente, el punto **B** representa la coordenada final, pero puede ocurrir interferencias en el ángulo ψ debido a las corrientes de viento, por ejemplo, como se tiene en el segmento **B-C**, disminuye la velocidad hasta que retome la orientación. Finalmente, llega al destino con un error de posicionamiento final, $8,4[m]$, en la prueba se tiene una velocidad máxima de $1[m/s]$.

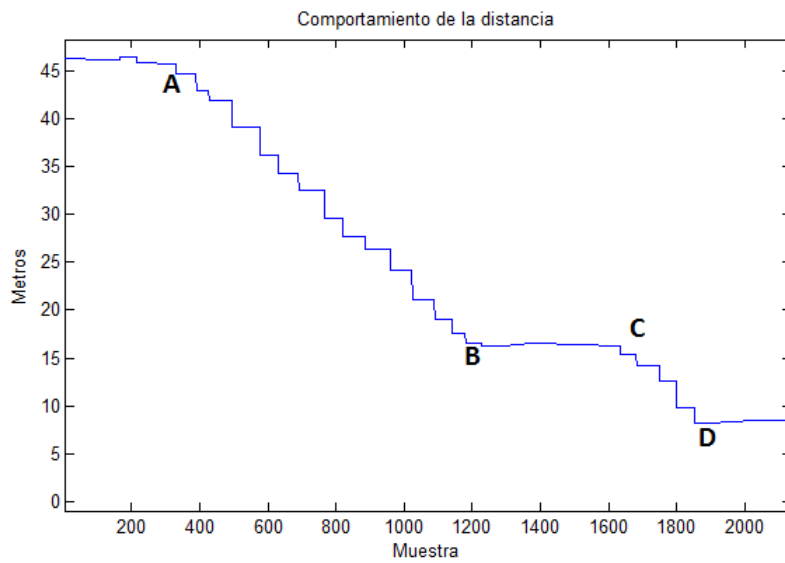


Figura 6.2: Comportamiento de la distancia durante el recorrido

El error final de posicionamiento se puede reducir con un aumento de la velocidad de traslación. El error mínimo es de 3,14[m] como se muestra en la Figura 6.3 con una velocidad máxima de 1,54[m/s].

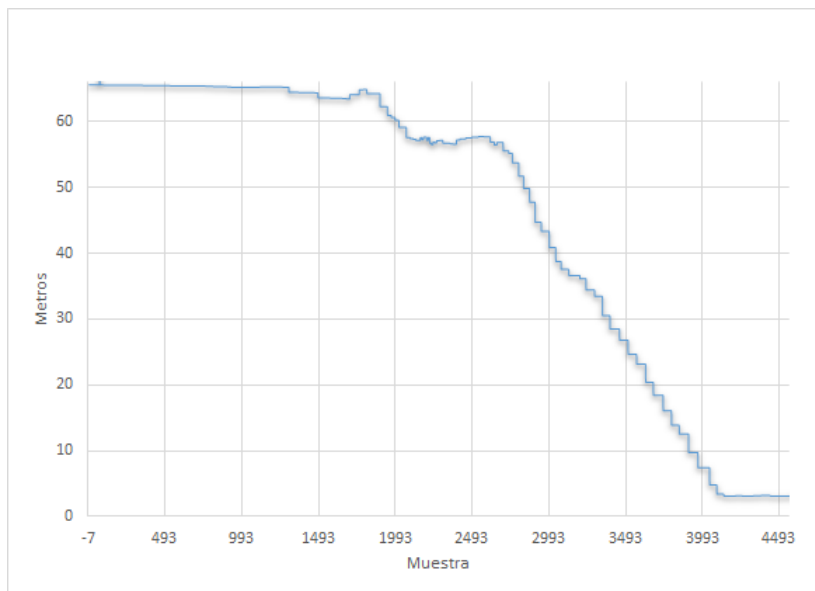


Figura 6.3: Comportamiento mejorado de la distancia durante el recorrido

En la Figura 6.4 se observa el comportamiento aproximadamente lineal de la trayectoria generada (segmento **A-B**). Las trayectorias **home-A** y **B-C** corresponden al despegue y aterrizaje en modo manual.

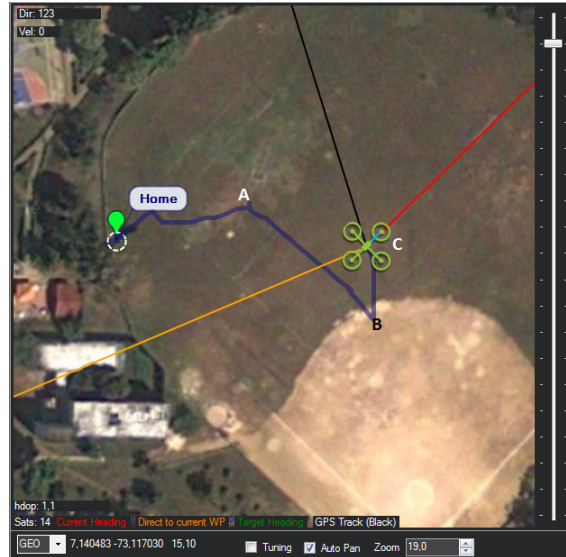


Figura 6.4: Prueba de vuelo realizada

En la Figura 6.5 se observa el recorrido desarrollado por el vehículo en vista 3D, realizado con Google Earth. Se tiene la coordenada deseada (setpoint) y la estación de control en tierra.

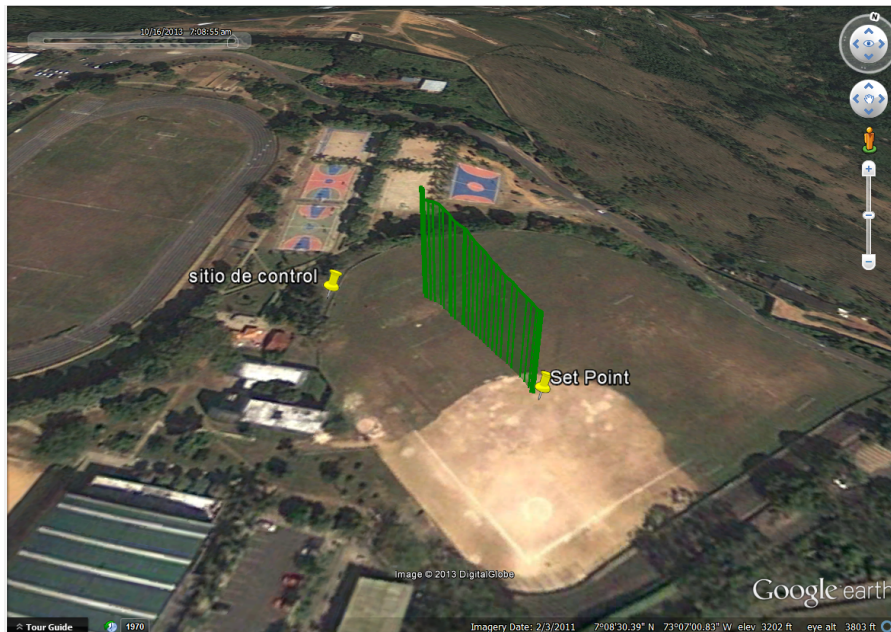


Figura 6.5: Recorrido de una prueba. Fuente: Google Earth.

Capítulo 7

Conclusiones y Trabajos futuros

En este trabajo de tesis se diseñó e implementó un algoritmo de rutas para la generación de trayectorias en un Vehículo Aéreo No Tripulado (UAV). En general los resultados obtenidos fueron satisfactorios, cumpliendo con los objetivos propuestos. A continuación se presentan las conclusiones generadas en diferentes partes del proyecto, así como las observaciones y los posible trabajos a futuro a partir de este proyecto.

7.1. Conclusiones

- Es importante determinar las acciones que tienen la mayor prioridad en el sistema y que son necesarias para un correcto funcionamiento. El control sobre los rotores y la adquisición de los datos, son dos condiciones fundamentales para el desarrollo del UAV y poder realizar la propuesta, una opción en el diseño del hardware para no perder ninguna acción mencionada es implementar dispositivos dedicados a cada tarea y luego reunir la información (Capítulo 5), de esta forma no se pierden los datos y el dominio sobre los actuadores.
- El error de posicionamiento final o *error absoluto*, es decir, cuando se considera que ha llegado a la coordenada deseada depende de la precisión del GPS, del algoritmo o fórmula *haversine* y de la velocidad de traslación proporcionalmente; este último se debe al efecto que tiene la masa inercial, cuando se traslada a una velocidad menor la inercia al detenerse tendrá menos efectos y la diferencia entre la coordenada deseada y la final obtenida será menor. El error angular, también depende de los cálculos realizados con las lecturas y la eficiencia del GPS.

- El error angular máximo es de $\pm 10^\circ$, no afecta significativamente la generación de la trayectoria y es aceptable para desarrollar una ruta aproximadamente lineal.
- Es fundamental realizar un estudio aerodinámico del vehículo, esto hace parte del diseño de la estructura para su posterior elaboración; de esta forma, el consumo energético y las acciones sobre el vehículo son menores para cumplir con los propósitos.
- En esta clase de proyectos, que se caracterizan por su alto nivel de implementación y donde se presentan actuadores mecánicos expuestos al contacto humano, es elemental tener siempre condiciones de seguridad para proteger tanto la integridad humana como los recintos cercanos y un protocolo de emergencia, debido a que, son estudios de investigación y una "pérdida de control" o mando tiene grandes posibilidades, en este proyecto fue un hecho donde los daños solamente fueron en la estructura de la aeronave.

7.2. Observaciones

A través del desarrollo de este proyecto se presentaron algunos acontecimientos que deben ser mencionados.

Se realizaron pruebas con varios materiales para la construcción de la estructura del quadrotor, usando en primer instancia acrílico, donde después de realizar las primeras pruebas experimentales de vuelo y tener impactos leves como el aterrizaje se comprobó que no era adecuado el uso de ese material. Actualmente la plataforma cuenta con fibra de vidrio como base de su estructura.

El uso de las baterías implica un proceso de recarga para reutilizarlas, este proceso condiciona la vida útil de la batería, traduciéndose en una disminución de autonomía en el vuelo a medida que se desgasta la batería.

El sistema *Ardupilot* permite la actualización de su *Firmware* pero todas sus versiones no ofrecen la estabilidad requerida para la ejecución de este proyecto, dado que una vez actualizado, alteraba los parámetros

de configuración, generando mal desempeño en las pruebas experimentales.

7.3. Trabajos Futuros

Como trabajos futuros se pueden tener muchas alternativas debido a la versatilidad del sistema, dado que pueden tratarse diversos temas de ingeniería como control moderno, visión artificial, robótica cooperativa entre otros. A continuación se hará mención de algunos posible trabajos que se deriven de este proyecto.

- Planificación de Rutas: Implementar una herramienta de *software* que permita la configuración de las rutas a realizar, para que pueda visitar múltiples puntos espaciales predefinidos durante un solo recorrido, interactuando con el usuario.
- Visión Artificial: Una vez adecuada una cámara, se pueden aplicar algoritmos de visión y procesamiento de imágenes que pueden ayudar a mejorar el desempeño del vehículo, así como también puede ser útil para el sistema de instrumentación.
- Diseñar e implementar una estructura que permita tener un mayor nivel de seguridad para las personas al momento de maniobrar el UAV y que de forma simultánea brinde una mayor robustez al sistema.
- Implementar algoritmos que permitan la interacción de varios UAVs de manera coordinada para cumplir una labor específica, bajo los lineamientos de la robótica cooperativa.

Bibliografía

- [1] JÁUREGUI URIBE, ALEJANDRA LUCÍA; VIANCHADA ESTÉVEZ CARLOS ALBERTO.(2010), *Diseño de un sistema de control de cabeceo y alabeo de un vehículo aéreo no tripulado (UAV)*, Trabajo de grado, Colombia: Universidad Industrial de Santander.
- [2] BOUABDALLAH, SAMIR.(2007), *Design and control of quadrotors with applicatioin to autonomous flying*, Thèse N ° 3727, École Polytechnique Fédérale de Lausanne, Switzerland.
- [3] ARGENTIM, LUCAS M.; REZENDE, WILLIAN C.; SANTOS, PAULO E.; AGUIAR, RENATO A. (2012). *PID, LQR and LQR-PID on a Quadcopter Platform*, Centro Universitario da FEI.
- [4] ZIYANG, ZHEN; ZHISHENG, WANG; YONG, HU. (2009), *Multi-Sensor Information Fusion for Aircraft Attitude Determination System*, Nanjing University of Aeronautics and Astronautics, Nanjing; Zhejiang University, Hangzhou; China.
- [5] SKIENA, STEVEN S., *The algorithm design manual*, second edition, New York, Springer editorial, 2008.
- [6] SÁNCHEZ, ÁNGELES MARTÍNEZ; VIEJO, GREGORIA BLANCO. (2001), *Algoritmos aleatorizados sobre grafos*, III Seminario de Matemática Discreta, E.U. Informática, U.P.M. España.
- [7] WILLIS, SHELDON G. (2008). U.S. Patente No. 20090210149 A1. United States.
- [8] MOVABLE TYPE LTD, *Calculate distance, bearing and more between Latitude/Longitude points*, En línea. Disponible en: <http://www.movable-type.co.uk/scripts/latlong.html>.
- [9] LEE, D.S.; GONZALEZ, L.F.; SRINIVAS, K.; PERIAUX, J. (2007). *Robust evolutionary algorithms for UAV/UCAV aerodynamic and RCS design optimisation*, Elsevier Ltd., Australia.

- [10] MING, ZUO; YING, LIU; YI, QIAN; XIONGWEN, HU; XIAOCHUAN, ZHAO; JINHUA, WANG. (2012), *Model-Based Design of UAV Autopilot Software*, International Conference on Computer and Information Application (ICCIA), Institute of Computer Application, Beijing, China.
- [11] OGATA, KATSUHIKO. (1998). *Ingeniería de control moderna*, tercera edición, Universidad de Minnesota. Pearson educación.
- [12] MCF51QE128. *Reference manual*. Disponible en <http://www.freescale.com/>
- [13] FERRI, ANDREU BERNA.(2010)., *Desarrollo de una plataforma de tiempo real para la implementación de algoritmos de control multivariables: Ampliación al control de orientación de vehículos aéreos*, Tesis de Maestría, España: Universidad de Politécnica de Valencia.
- [14] ALL THE WORLD'S ROTORCRAFT, *De Bothezat 1922*, citado el 16-07-2012, disponible en: <http://www.aviastar.org/>
- [15] STEPANIAK, MICHAEL J.(2008)., *A Quadrotor Sensor Platforms*, Tesis doctoral, Ohio University.
- [16] BERMÚDEZ RIVERA, FREDDY ALEXANDER; BARRERA TORRES, YESID JHAIR.(2011), *Diseño e implementación de una plataforma para la adquisición y manejo de datos de vehículos de transporte utilizando tecnología GPS y GPRS*, Trabajo de grado, Colombia: Universidad Industrial de Santander.
- [17] REAL ACADÉMIA ESPAÑOLA, *Diccionario de la lengua española*, Vigésima segunda edición, rae.es

Apéndices

Apéndice A

Códigos de programación

main.c

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include <string.h>
#include <stdlib.h>
#include <extras/extras_stdlib.h>
#include <stdio.h>
#include <math.h>
#include <inicializaciones y declaraciones.h>
#include <funciones.h>

#define LED0 PTED_PTED6
#define LED1 PTAD_PTAD1
#define LED2 PTHD_PTHD0
#define LED3 PTAD_PTAD0
#define LED4 PTHD_PTHD1

int estable;
char str_yaw[7], str_yaw_dec[3];
int yaw_send, yaw_send_dec;

#ifdef __cplusplus
extern "C"
#endif
void MCU_init(void); /* Device initialization function declaration */

void main(void) {
MCU_init(); /* call Device Initialization */
/* include your code here */
//*****LEDS *****//
PTADD_PTADD0=1;
PTADD_PTADD1=1;
PTHDD_PTHDD0=1;
PTHDD_PTHDD1=1;
PTEDD_PTEDD6=1;
//*****//

while(man==1)
{
umbraizar(); /// Almacena los valores centrales.
d_i=1;
}

while(man==0)
{LED4=~LED4;
Delay(100);}
RTSC_RTIE=0; // RTC Disable

for(;;) {
/* __RESET_WATCHDOG(); By default COP is disabled with device init. When enabling, also reset the watchdog. */

if(RX_fLag){
LED0=1;
lon_lat_yaw(); // Calcula los datos de lng, lat y yaw, y determina el yaw_dese

if(man==0 && d_i==1)
{
distancia();

d_inicial=d; // Distancia inicial: d_inicial
d_i=0;
}

if(abs((int)(yaw-yaw_ant))<8.0)
{
error_ant=error;
error=(int)(yaw_dese-yaw);
if(error>180)
error=error-360;
else if(error<-180)
error=360+error;

//kalman_update2(error);

if(error>180)
error=error_ant;
}

distancia();

cont_ciclos++;
if(cont_ciclos==7)
{
velocidad=(d-d_anterior)/(TPM2CNT*5.333e-6);
d_anterior=d;
cont_ciclos=0;
TPM2CNT=0;
}

RX_fLag=0;
LED0=0;
}
if(man==0)
{
if(nm==0)
{
yaw_dese = yaw + 50;
nm=1;
}
auto_pilot(); // Control del yaw
}

} /* loop forever */
/* please make sure that you never leave main */
}
```

Funciones.h

```
#include <math.h>

#define LED0 PTED_PTED6
#define LED1 PTAD_PTAD1
#define LED2 PTHD_PTHD0
#define LED3 PTAD_PTAD0
#define LED4 PTHD_PTHD1

void serial_TX (char dat){
  SCI1D = dat;
  while (!SCI1S1_TC){}
}
void serial2_TX (char dat){
  SCI2D=dat;
  while (!SCI2S1_TC){}
}
void cadena_tx_terminal (const char *str,byte
Enter)
{
  byte TX;
  for (TX=0;TX<strlen(str);TX++)
  {
    while(SCI2S1_TC==0){}
    SCI2D=(unsigned char)str[TX];
    while(SCI2S1_TC==0){}
  }
  if (Enter == 1)
  {
    SCI2D=0x0A;
    while(SCI2S1_TC==0){}
    SCI2D=0x0D;
    while(SCI2S1_TC==0){}
  }
}
void Delay (unsigned int t){
  unsigned int l,m;
  for(l=0;l<300;l++){
    for(m=0;m<t;m++){}
  }
}
float btf(unsigned char b3, unsigned char b2,
unsigned char b1, unsigned char b0)
{
  float salida;
  *((unsigned char*)&salida+3)=b3;
  *((unsigned char*)&salida+2)=b2;
  *((unsigned char*)&salida+1)=b1;
  *((unsigned char*)&salida+0)=b0;
  return salida;
}

float deg_to_rad(float x)
{
  float y;
  y=x*3.1416/180;
  return y;
}

void lon_lat_yaw(void)
{
  yaw_ant=yaw;

  //===== Latitud, longitud
  //=====//
  Lat=(data[3]<<24 | data[2]<<16 | data[1]<<8 |
data[0])/(1e7);

  Lng=(data[7]<<24 | data[6]<<16 | data[5]<<8 |
data[4])/(1e7);

  //====Cálculo del roll, pitch y yaw====//
  yaw_cal=(btf(data[8], data[9], data[10],
data[11]))*180/(3.14159);

  if(yaw_cal>180)
  yaw=yaw_cal-360;
  else if(yaw_cal>-180 && yaw_cal<0)
  yaw=yaw_cal+360;
  else
  yaw=yaw_cal;
  //=====//

  if(yaw<0 || yaw>361)
  {
    yaw=yaw_ant;
  }

  //===== Coordenadas deseadas
  (setpoint, lng_sp,lat_sp) y cálculo del
  Yaw_deseado =====//
  /*
  den=Lng_sp-Lng;
  num=Lat_sp-Lat;
  if(den>0)
  yaw_dese=90-(atan(num/den)*180/(3.14159));
  else
  yaw_dese=270-(atan(num/den)*180/(3.14159));

  //=====//
  //=====//

  void umbralizar(void)
  {
    mid_ch1=cont_ch1;
    mid_ch2=cont_ch2;
    mid_ch4=cont_ch4;
  }

  void distancia(void)
  {
    Lat1=deg_to_rad(Lat); //0.124605407;
    // Coordenadas instantáneas, En
    radianes
    Lon1=deg_to_rad(Lng); //-1.276167129;

    Lat2=deg_to_rad(Lat_sp); //0.124612437;
    // Setpoint o coordenadas deseadas
    Lon2=deg_to_rad(Lng_sp);//-1.276154378;

    dLat=Lat2-Lat1; //7.02982e-06; // Lat2-
    Lat1
  }
}
```

```

dLon=Lon2-Lon1; //1.27505e-05;

h_haversine=(sin(dLat/2)*sin(dLat/2))+cos(Lat1)*
cos(Lat2)*sin(dLon/2)*sin(dLon/2));

d=2000*r*asin(sqrt(h_haversine)); // En
metros
}

unsigned int velocidad_adelante(void)
{
unsigned int v;
if(d>1)
v=450; //mid_ch2-(98(-abs(error)/180)); //470;
//367: Máxima aceleración hacia adelante;
else
v=mid_ch2;
return v;
}

void correccion_yaw(void)
{
if(abs(error_int) > 20)
error_int=(error_int/abs(error_int))*20;

queremos_ch4_ant=queremos_ch4;
if(error==0)
queremos_ch4=queremos_ch4_ant;
else
{
temp_queremos_ch4=(int)((180.5*error/180)+(0.40*error_int)+(0.25*error_der));
if(abs(temp_queremos_ch4)>40)
temp_queremos_ch4=(temp_queremos_ch4/abs
(temp_queremos_ch4))*40;

queremos_ch4=mid_ch4+temp_queremos_ch4;

void transmitir(float x, float y, float z, float
w)
{
/**Transmisión Xbee***/
int x_send, x_send_dec, y_send, y_send_dec,
z_send, z_send_dec, w_send, w_send_dec, d_send,
d_send_dec;
char str_x[7], str_x_dec[3], str_y[7],
str_y_dec[3], str_z[7], str_z_dec[3], str_w[7],
str_w_dec[3], str_e[5], str_d[7], str_d_dec[3];
itoa(x, str_x,10);
x_send=x*100;
x_send_dec=x_send-((int)(x)*100);
itoa(x_send_dec, str_x_dec, 10);

itoa(y, str_y,10);
y_send=y*100;
y_send_dec=y_send-((int)(y)*100);
itoa(y_send_dec, str_y_dec, 10);

itoa(z, str_z,10);
z_send=z*100;
z_send_dec=z_send-((int)(z)*100);
itoa(z_send_dec, str_z_dec, 10);

itoa(w, str_w,10);
itoa(error, str_e,10);

itoa(d, str_d,10);
d_send = d*100;
d_send_dec=d_send-((int)(d)*100);

itoa(d_send_dec, str_d_dec, 10);

cadena_tx_terminal(str_x, 0);
serial2_TX('.');
cadena_tx_terminal(str_x_dec, 0);
serial2_TX(',');
cadena_tx_terminal(str_y, 0);
serial2_TX('.');
cadena_tx_terminal(str_y_dec, 0);

serial2_TX(',');
cadena_tx_terminal(str_z, 0);
serial2_TX('.');
cadena_tx_terminal(str_z_dec, 0);

serial2_TX(',');
cadena_tx_terminal(str_w, 0);

serial2_TX(',');
cadena_tx_terminal(str_d, 0);
serial2_TX('.');
cadena_tx_terminal(str_d_dec, 0);

serial2_TX(',');
cadena_tx_terminal(str_e, 1);
//*****
*****//
}

void auto_pilot(void)
{
correccion_yaw();
queremos_ch2=mid_ch2;
queremos_ch1=mid_ch1;
LED2=0;
}

```

Inicializaciones.h

```
//*****MavLink*****//
byte leng, id_msj, cont, id_comp, sys_id,
Longitud, Longitud_dos;
byte data[14], data2[14], data3[14];
float roll, pitch, yaw, yaw_cal, yaw_dese,
yaw_ant;
float alt, lat, lng, num, den;
unsigned int cont_ch4, cont_ch3, cont_ch5,
cont_ch2, cont_ch1, queremos_ch1, queremos_ch2,
queremos_ch3, queremos_ch4=564, queremos_ch4_ant,
mid_ch4, mid_ch2, mid_ch1, cont_ciclos;
unsigned int numero_dos;
char datost[5], lat_aux[10], lng_aux[10],
yaw_aux[10];
int n, nn, nnn, m, TPM1CNT_AUX,
temp_queremos_ch4;
//*****
*****//
int man=1;
extern long int cont_man;
int h, f;
int time_t, RX_flag;
int segundos=0, umbral=0;
////////////////////////////////////
////
int d_i=1, nm=0;

float lat1, lat2, lon1, lon2, dlat, dlon,
h_haversine, r=6378.137, d, d_inicial,
d_anterior, velocidad, error, error_int=0,
error_ant, error_der=0;

//Coordenadas deseadas, setpoint
=====//
float lng_sp=-73.1184983; // Cancha de
futbol: -73.1184983 Diamante: -
73.11696389;
float lat_sp=7.1404722; // Cancha de
futbol: 7.1404722 Diamante: 7.140275;
///=====
=====//
```


Apéndice B

Esquemático y Board

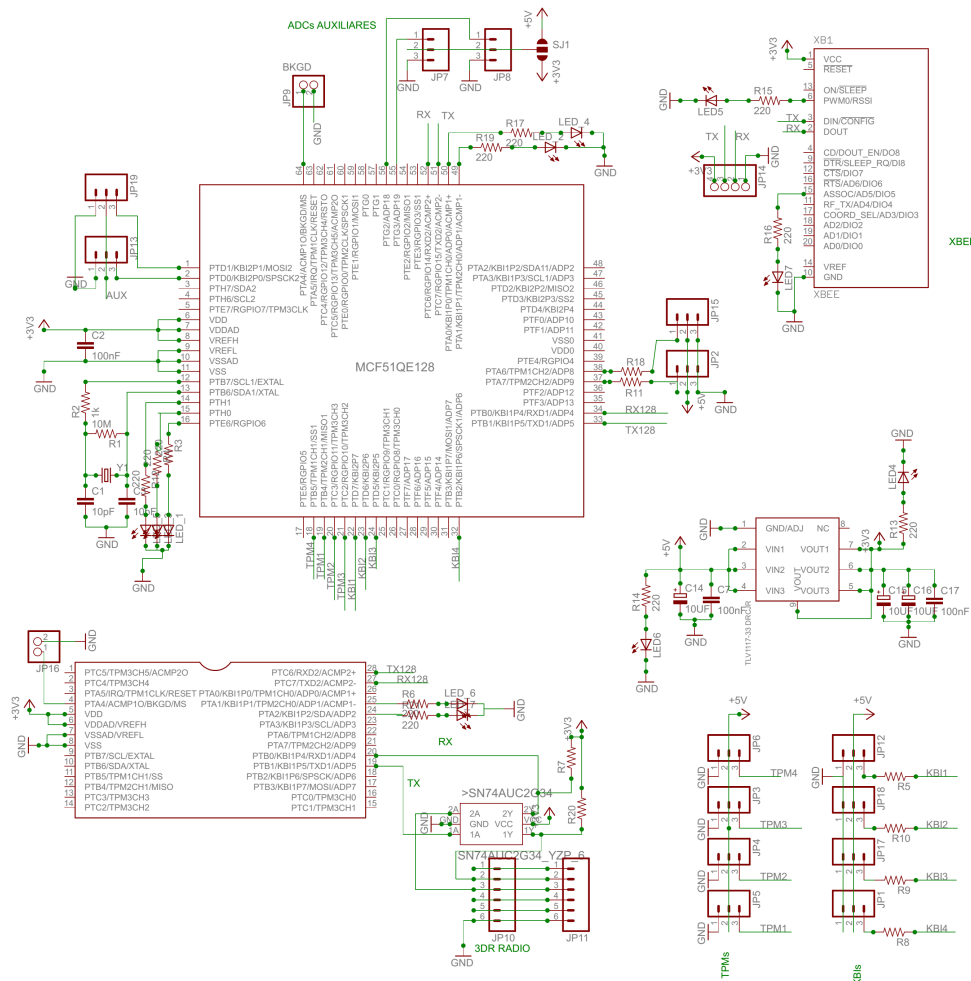


Figura B.1: Esquemático de la tarjeta. Fuente: Autores.

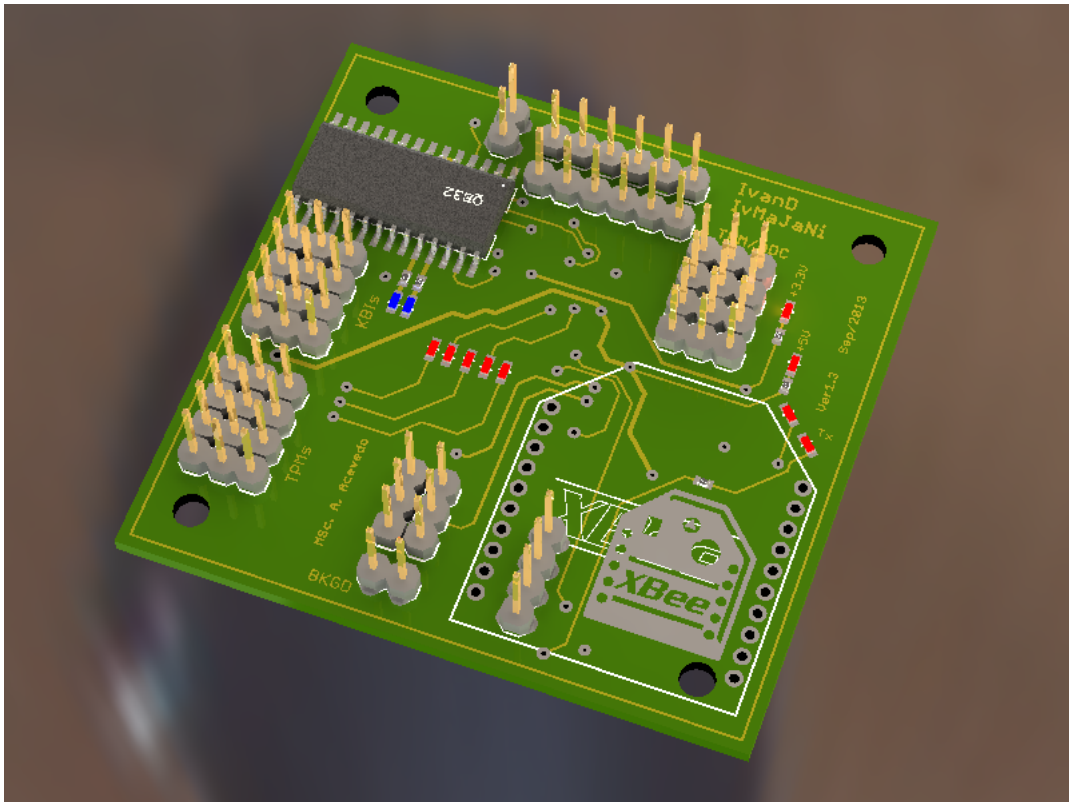


Figura B.4: Tarjeta final. Fuente: Autores.